Expression data analysis with

Chipster

Eija Korpelainen, Massimiliano Gentile
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?

- **How to operate the Chipster software**
- **How to analyze microarray data**
  - Central concepts
  - Analysis workflow
  - What happens in the different analysis steps
- **How to analyze RNA-seq data**
  - Short introduction to analysis workflow and central concepts

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Introduction to Chipster

Chipster

- **Provides an easy access to over 450 analysis tools**
  - No programming or command line experience required
- **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

Analysis tools

➢ **260 NGS tools for**

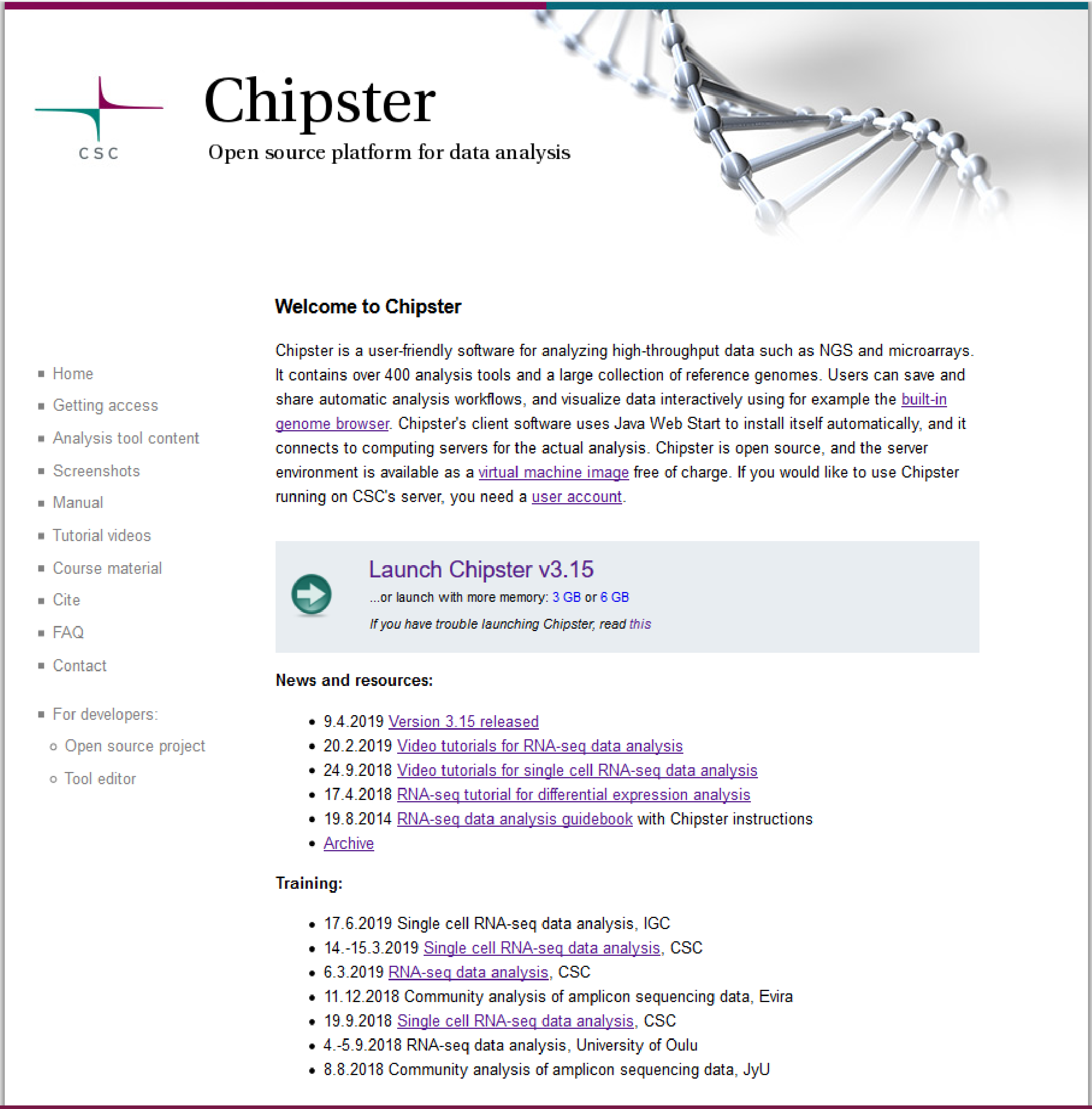
- RNA-seq
- single cell RNA-seq
- miRNA-seq
- exome/genome-seq
- ChIP-seq
- FAIRE/DNase-seq
- CNA-seq
- 16S rRNA sequencing

➢ **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



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 v3.15

...or launch with more memory: 3 GB or 6 GB

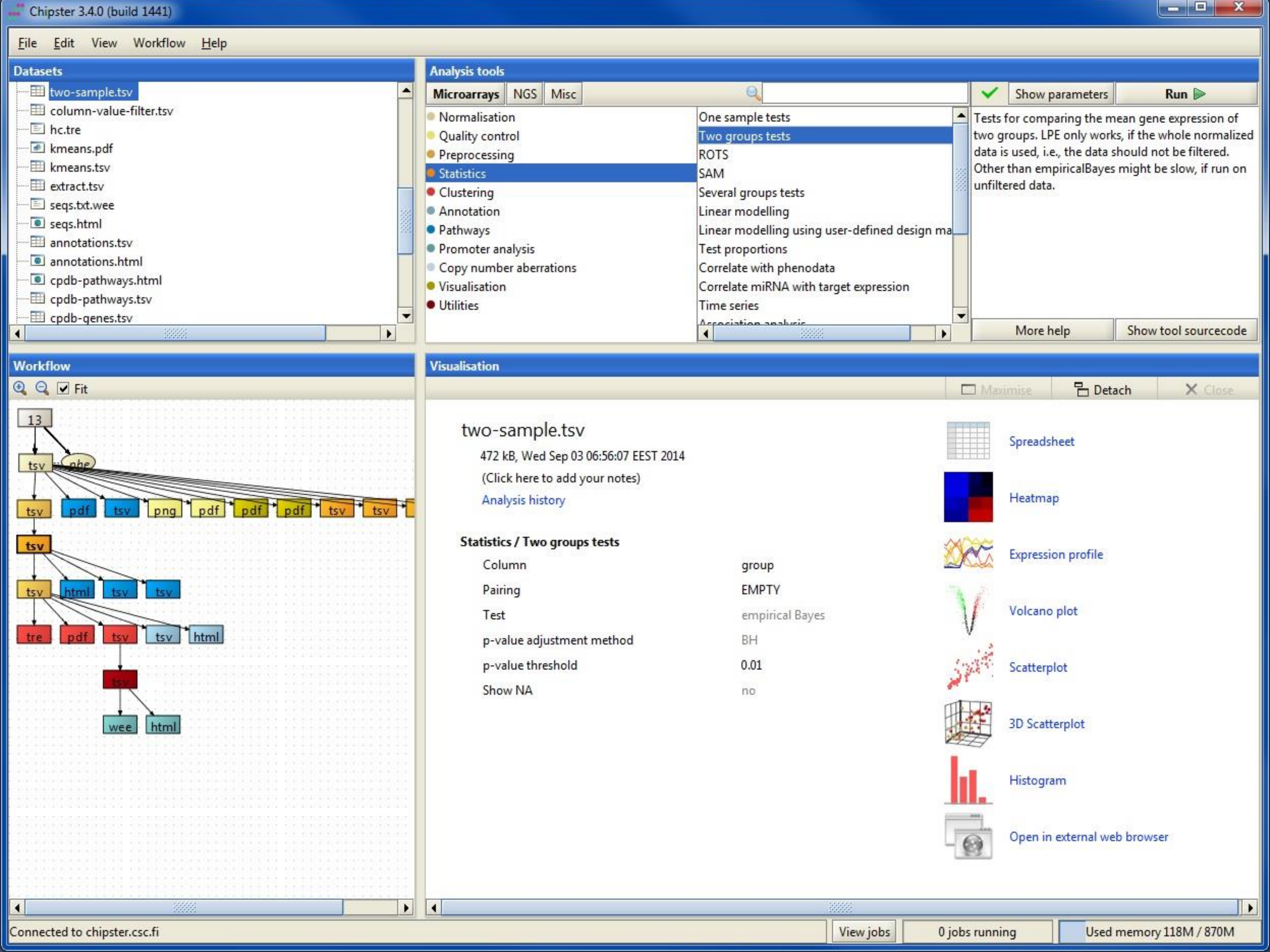
If you have trouble launching Chipster, read this

News and resources:

- 9.4.2019 Version 3.15 released
- 20.2.2019 Video tutorials for RNA-seq data analysis
- 24.9.2018 Video tutorials for single cell RNA-seq data analysis
- 17.4.2018 RNA-seq tutorial for differential expression analysis
- 19.8.2014 RNA-seq data analysis guidebook with Chipster instructions
- Archive

Training:

- 17.6.2019 Single cell RNA-seq data analysis, IGC
- 14.-15.3.2019 Single cell RNA-seq data analysis, CSC
- 6.3.2019 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



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.

two-sample.tsv

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

(Click here to add your notes)

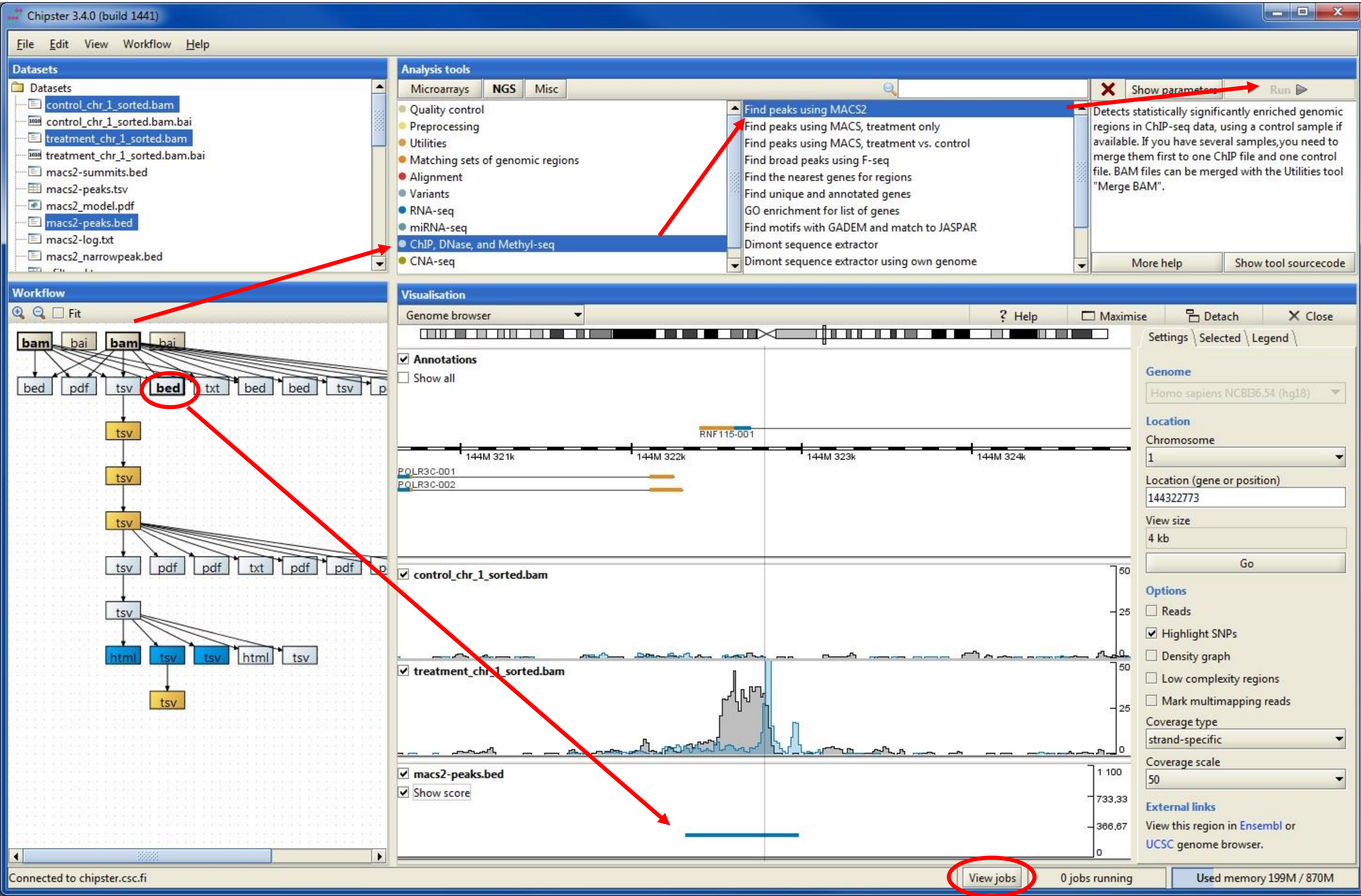
Analysis history

Statistics / Two groups tests

Column	group
Pairing	EMPTY
Test	empirical Bayes
p-value adjustment method	BH
p-value threshold	0.01
Show NA	no

Mode of operation

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



Chipster 3.4.0 (build 1441)

File Edit View Workflow Help

Datasets

- Datasets
  - 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
  - macs2_narrowpeak.bed

Analysis tools

Microarrays | NGS | Misc

- Quality control
- Preprocessing
- Utilities
- Matching sets of genomic regions
- Alignment
- Variants
- RNA-seq
- miRNA-seq
- ChIP, DNase, and Methyl-seq
- CNA-seq

- 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
- Dimont sequence extractor using own genome

Detects statistically significantly enriched genomic regions in ChIP-seq data, using a control sample if available. If you have several samples,you need to merge them first to one ChIP file and one control file. BAM files can be merged with the Utilities tool "Merge BAM".

Visualisation

Genome browser

☑ Annotations
□ Show all

☑ control_chr_1_sorted.bam

☑ treatment_chr_1_sorted.bam

☑ macs2-peaks.bed
☑ Show score

Settings | Selected | Legend

Genome
Homo sapiens NCBI36.54 (hg18)

Location
Chromosome: 1
Location (gene or position): 144322773
View size: 4 kb

Options
□ Reads
☑ Highlight SNPs
□ Density graph
□ Low complexity regions
□ Mark multimapping reads

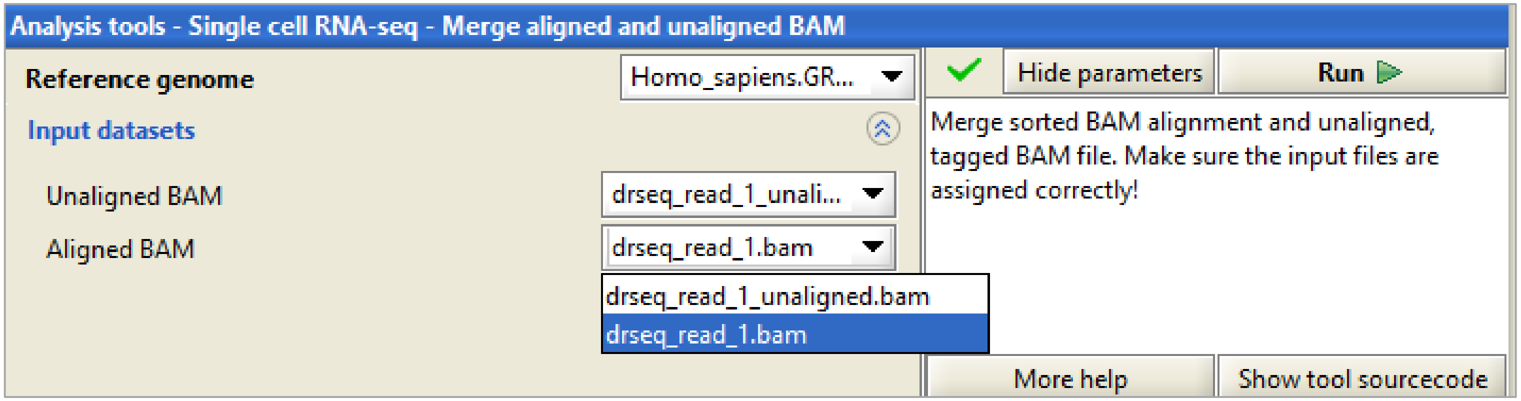
Coverage type: strand-specific
Coverage scale: 50

External links
View this region in Ensembl or UCSC genome browser.

Connected to chipster.csc.fi | View jobs | 0 jobs running | 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...

Input datasets

Unaligned BAM: drseq_read_1_unali...

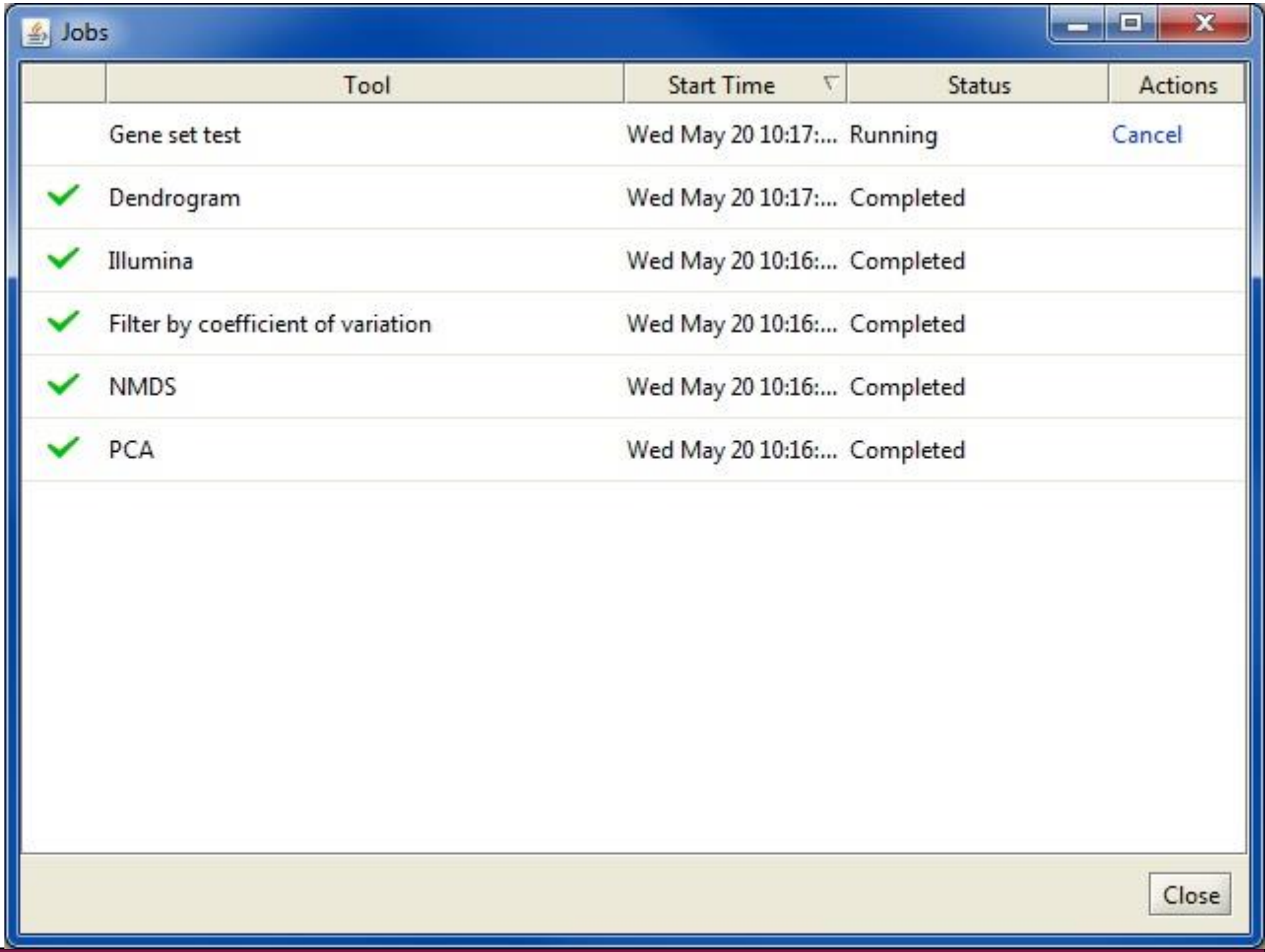
Aligned BAM: drseq_read_1.bam

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

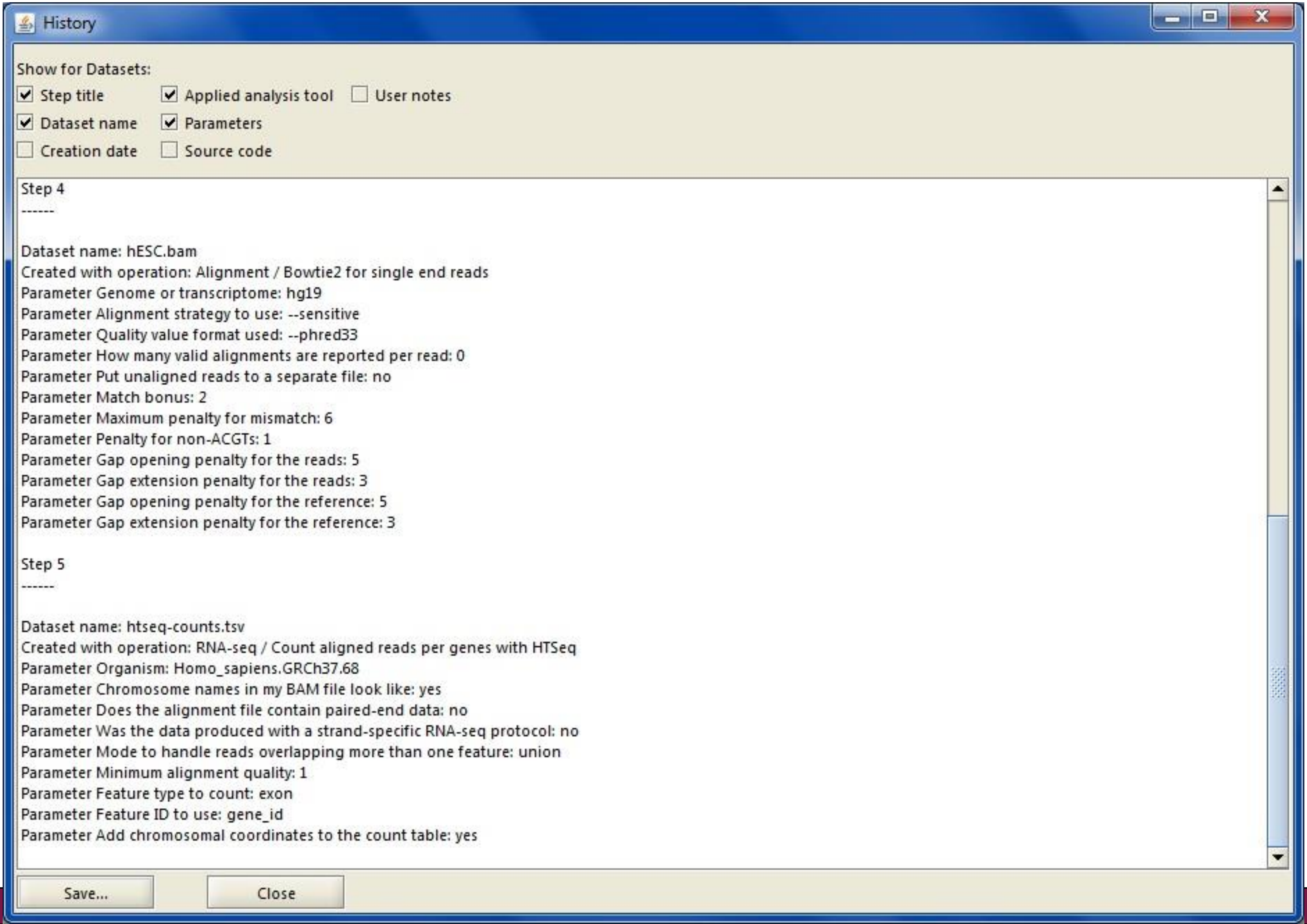
- **You can run many analysis jobs at the same time**
- **Use Job manager to**
  - view status
  - cancel jobs
  - view time
  - view parameters



	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	

Analysis history is saved automatically

-you can add tool source code to reports if needed



Show for Datasets:

☑ Step title ☑ Applied analysis tool ☐ User notes

☑ Dataset name ☑ Parameters

☐ Creation date ☐ Source code

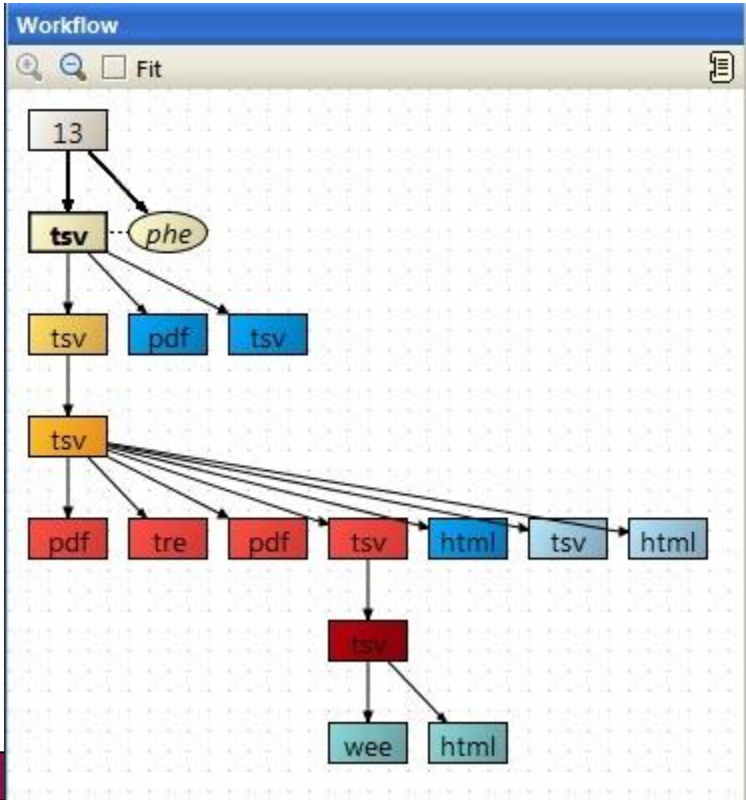
```
Step 4
------

Dataset name: hESC.bam
Created with operation: Alignment / Bowtie2 for single end reads
Parameter Genome or transcriptome: hg19
Parameter Alignment strategy to use: --sensitive
Parameter Quality value format used: --phred33
Parameter How many valid alignments are reported per read: 0
Parameter Put unaligned reads to a separate file: no
Parameter Match bonus: 2
Parameter Maximum penalty for mismatch: 6
Parameter Penalty for non-ACGTs: 1
Parameter Gap opening penalty for the reads: 5
Parameter Gap extension penalty for the reads: 3
Parameter Gap opening penalty for the reference: 5
Parameter Gap extension penalty for the reference: 3

Step 5
------

Dataset name: htseq-counts.tsv
Created with operation: RNA-seq / Count aligned reads per genes with HTSeq
Parameter Organism: Homo_sapiens.GRCh37.68
Parameter Chromosome names in my BAM file look like: yes
Parameter Does the alignment file contain paired-end data: no
Parameter Was the data produced with a strand-specific RNA-seq protocol: no
Parameter Mode to handle reads overlapping more than one feature: union
Parameter Minimum alignment quality: 1
Parameter Feature type to count: exon
Parameter Feature ID to use: gene_id
Parameter Add chromosomal coordinates to the count table: yes
```

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

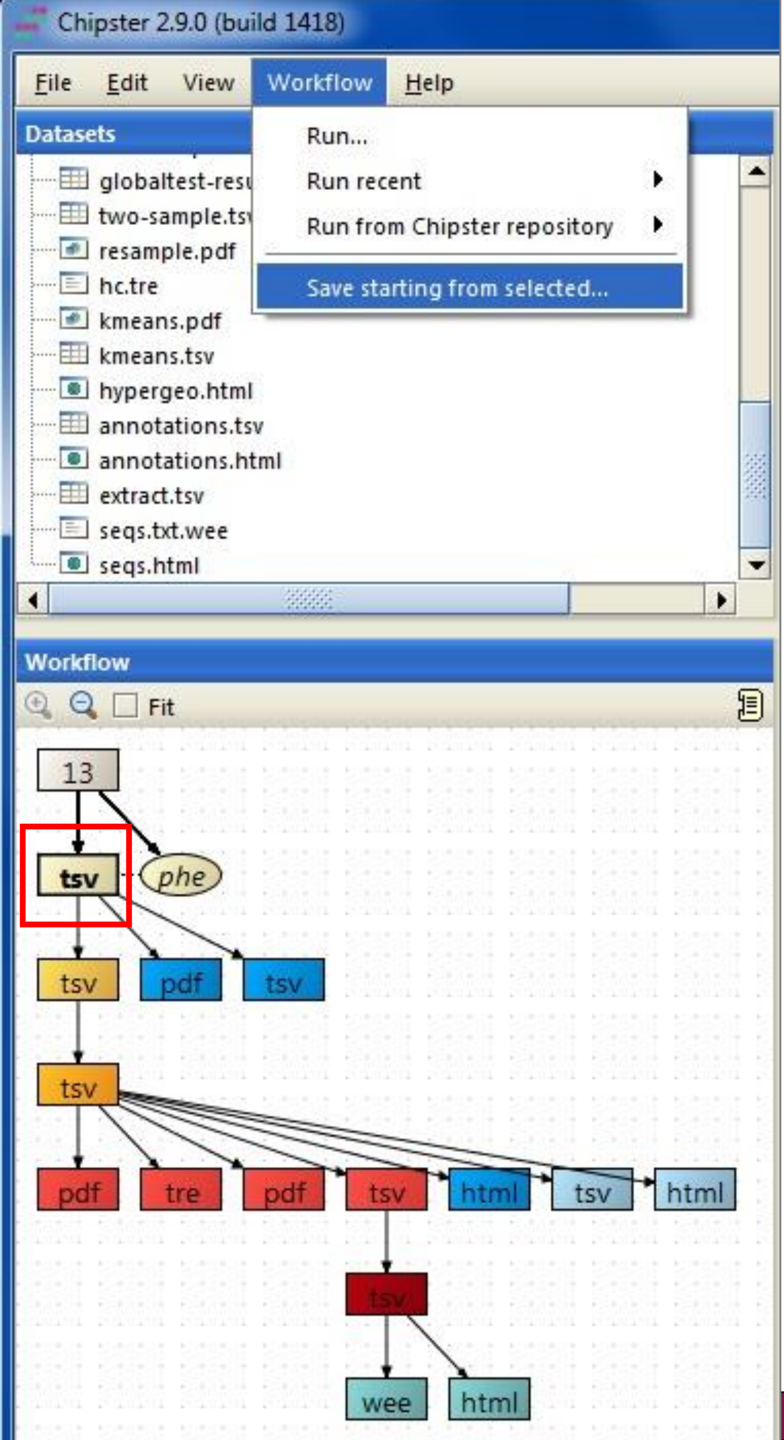
- **Shows the relationships of the files**
- **You can move the boxes around, and zoom in and out.**
- **Several files can be selected by keeping the Ctrl key down**
- **Right clicking on the data file allows you to**
  - Save an individual result file (”Export”)
  - Delete
  - Link to another data file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

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

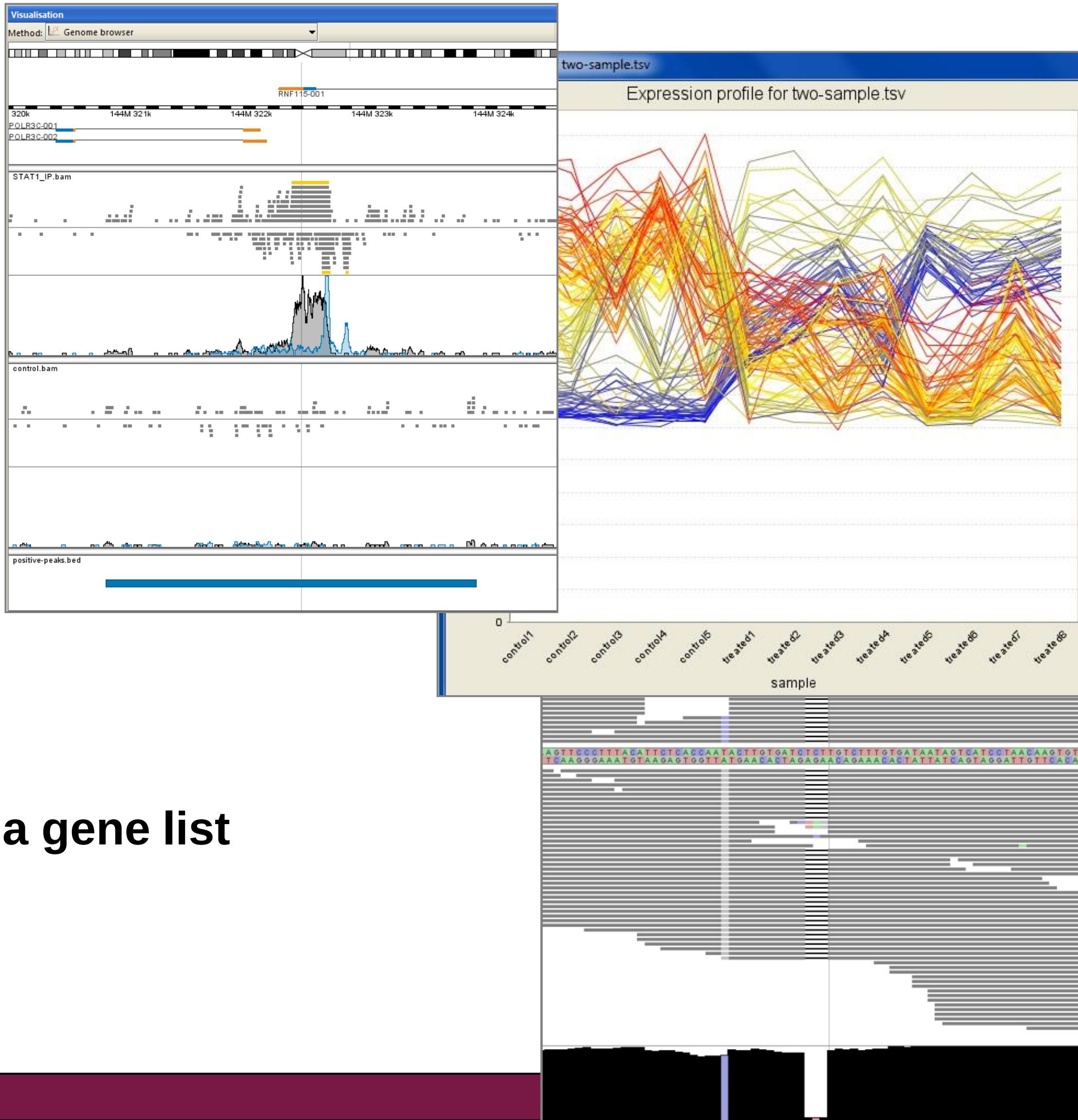
- **Two types of visualizations**

  1. **<u>Interactive visualizations</u> produced by the client program**
     - Select the visualization method from the pulldown menu
     - Save by right clicking on the image

  2. **<u>Static images</u> 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

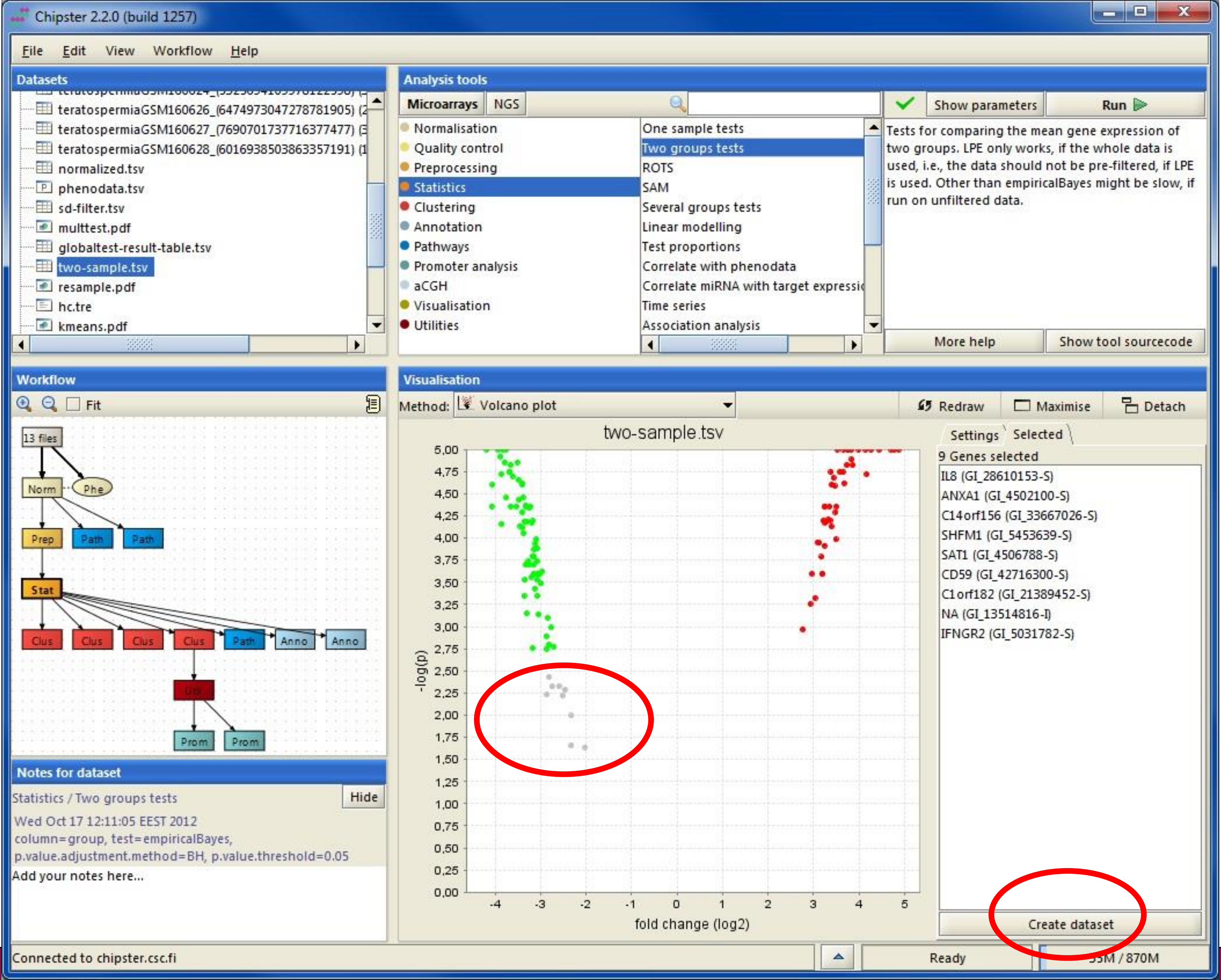
- ➢ **Genome browser**
- ➢ **Spreadsheet**
- ➢ **Histogram**
- ➢ **Venn diagram**
- ➢ **Scatterplot**
- ➢ **3D scatterplot**
- ➢ **Volcano plot**
- ➢ **Expression profiles**
- ➢ **Clustered profiles**
- ➢ **Hierarchical clustering**
- ➢ **SOM clustering**



Method: Genome browser

Available actions:

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



teratospermiaGSM160624_(3323034103378122338) (
teratospermiaGSM160626_(6474973047278781905) (2
teratospermiaGSM160627_(7690701737716377477) (3
teratospermiaGSM160628_(6016938503863357191) (1
normalized.tsv
phenodata.tsv
sd-filter.tsv
multtest.pdf
globaltest-result-table.tsv
two-sample.tsv
resample.pdf
hc.tre
kmeans.pdf

Normalisation
Quality control
Preprocessing
Statistics
Clustering
Annotation
Pathways
Promoter analysis
aCGH
Visualisation
Utilities

One sample tests
Two groups tests
ROTS
SAM
Several groups tests
Linear modelling
Test proportions
Correlate with phenodata
Correlate miRNA with target expressi
Time series
Association analysis

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

Statistics / Two groups tests

Wed Oct 17 12:11:05 EEST 2012
column=group, test=empiricalBayes, p.value.adjustment.method=BH, p.value.threshold=0.05

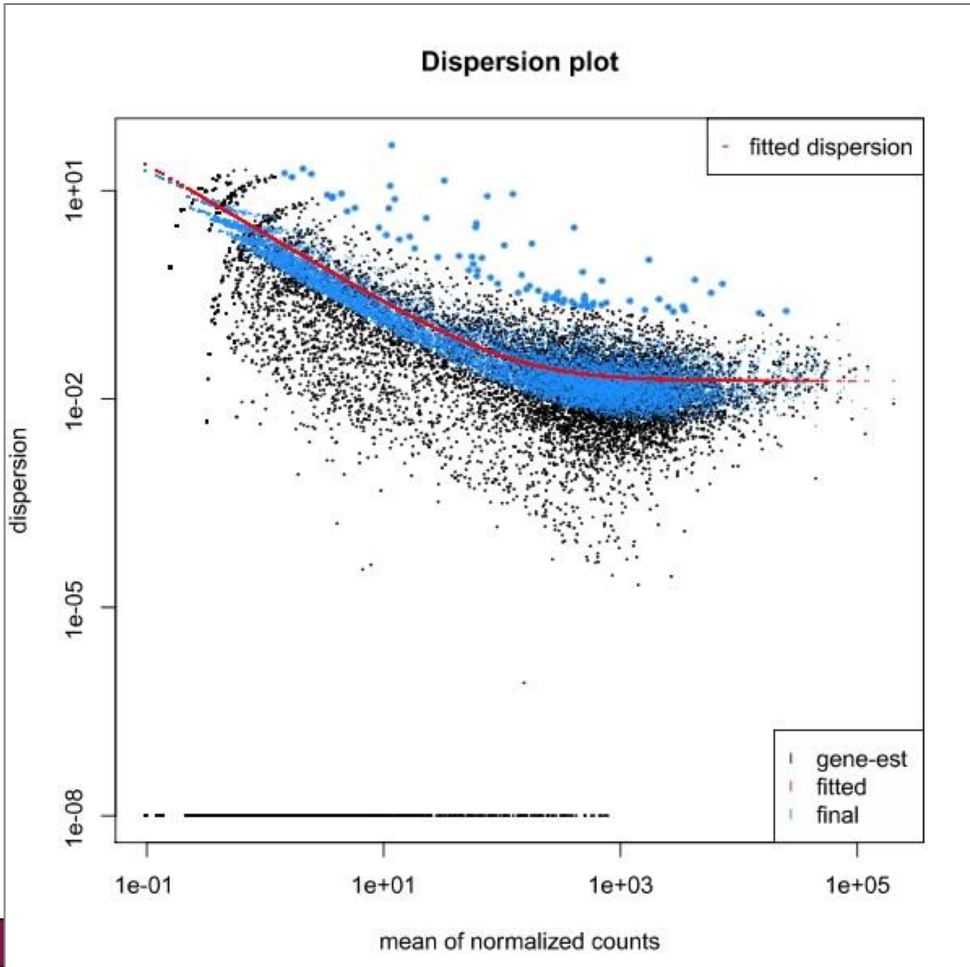
Add your notes here...

9 Genes selected

IL8 (GI_28610153-S)
ANXA1 (GI_4502100-S)
C14orf156 (GI_33667026-S)
SHFM1 (GI_5453639-S)
SAT1 (GI_4506788-S)
CD59 (GI_42716300-S)
C1orf182 (GI_21389452-S)
NA (GI_13514816-I)
IFNGR2 (GI_5031782-S)

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
- **Import from BaseSpace**
- **Import from ENA**
- **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

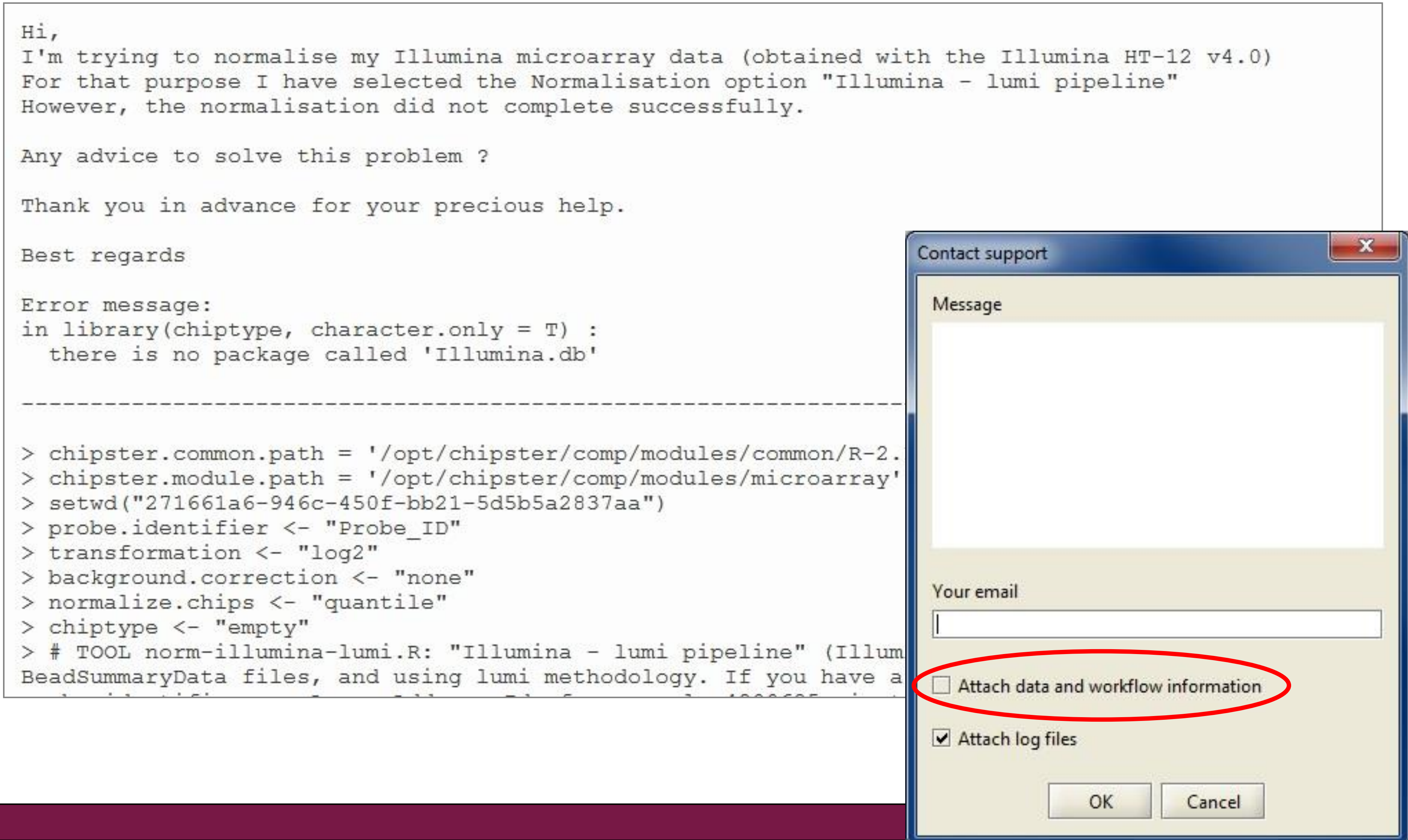
- **What kind of data files can I use in Chipster?**
    - Compressed files (.gz) are ok
    - FASTQ, BAM, read count files (.tsv), GTF

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

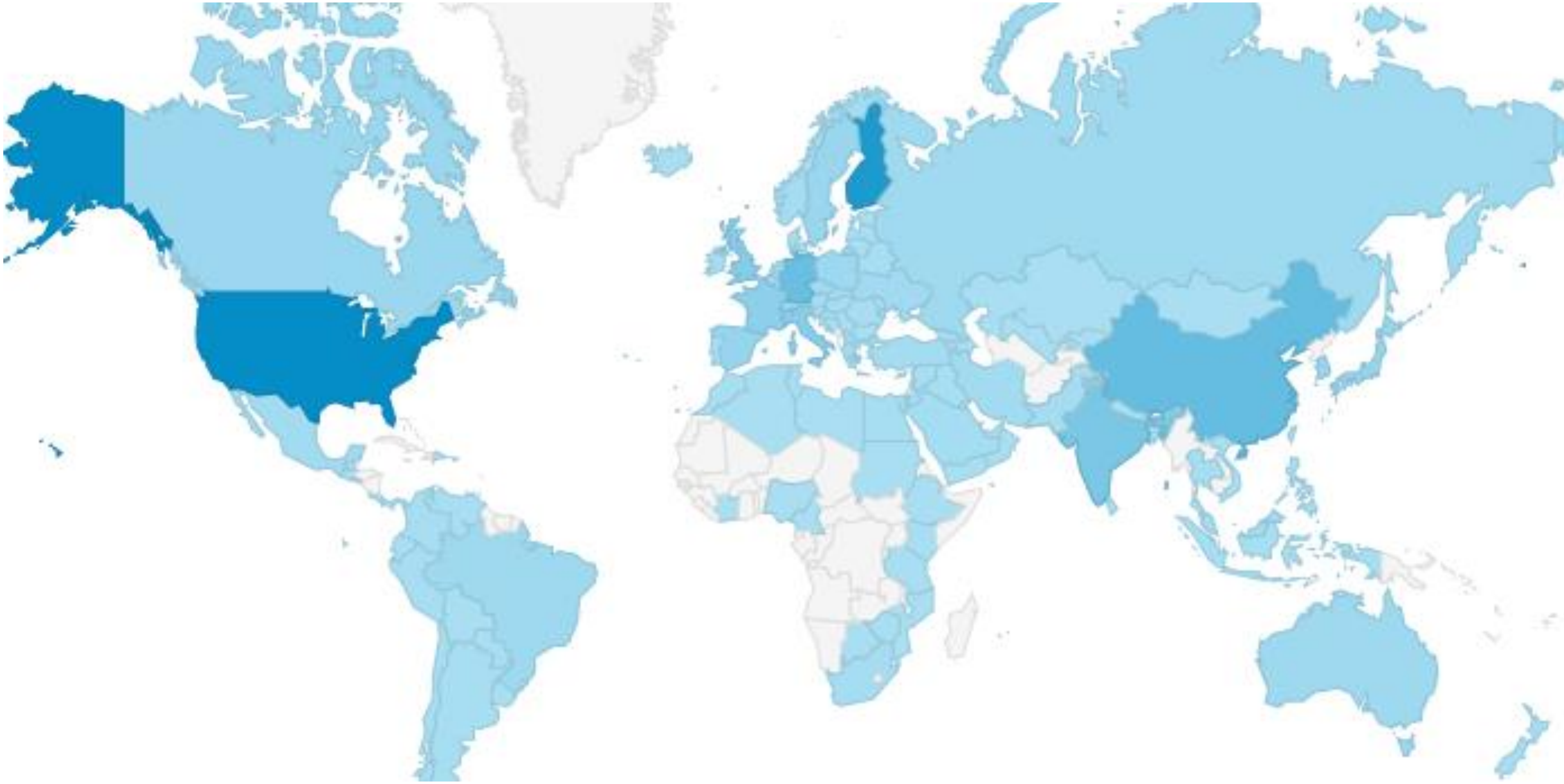
-request includes the error message and link to analysis session (optional)



```
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
```

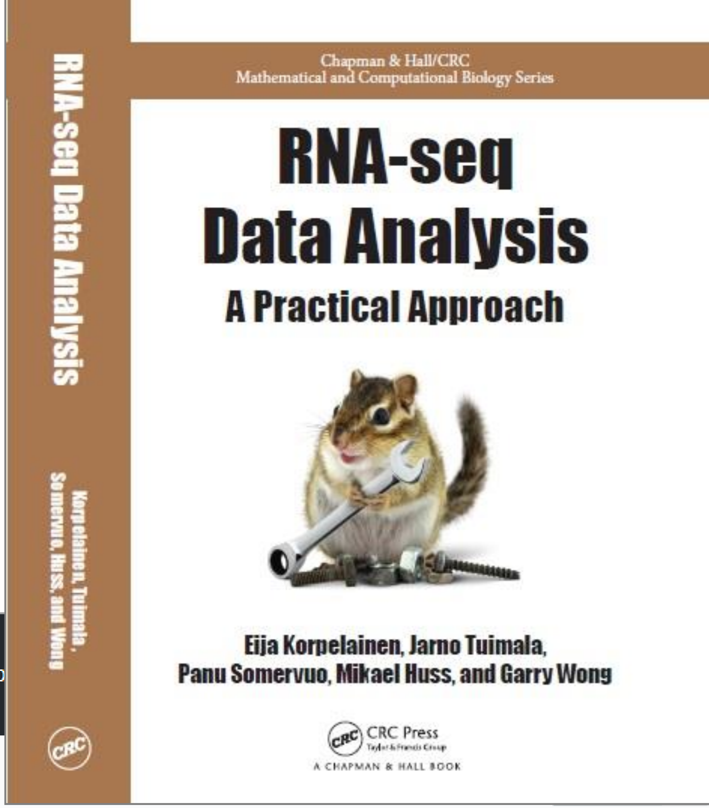
Acknowledgements to Chipster users and contibutors

- Users’ feedback and ideas have really helped us to shape the software over the years. Let us know what needs to be improved!



More info

- chipster@csc.fi
- http://chipster.csc.fi
- YouTube channel Chipster tutorials
- https://chipster.csc.fi/manual/courses.html



Chapman & Hall/CRC Mathematical and Computational Biology Series

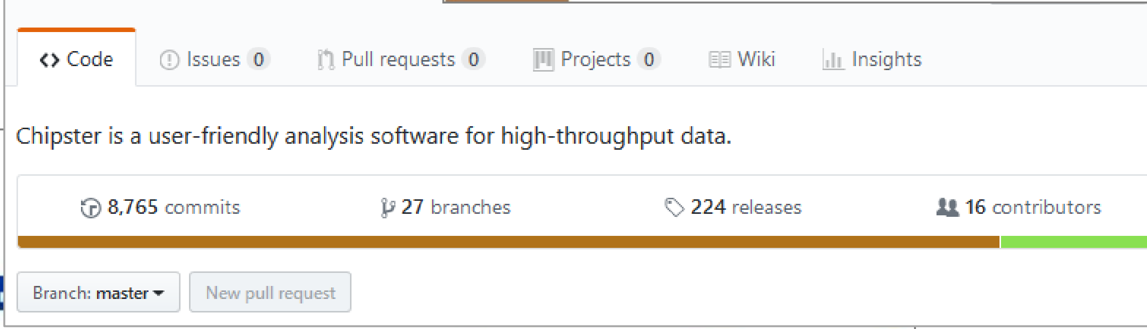
RNA-seq Data Analysis

A Practical Approach

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

CRC Press Taylor & Francis Group

A CHAPMAN & HALL BOOK

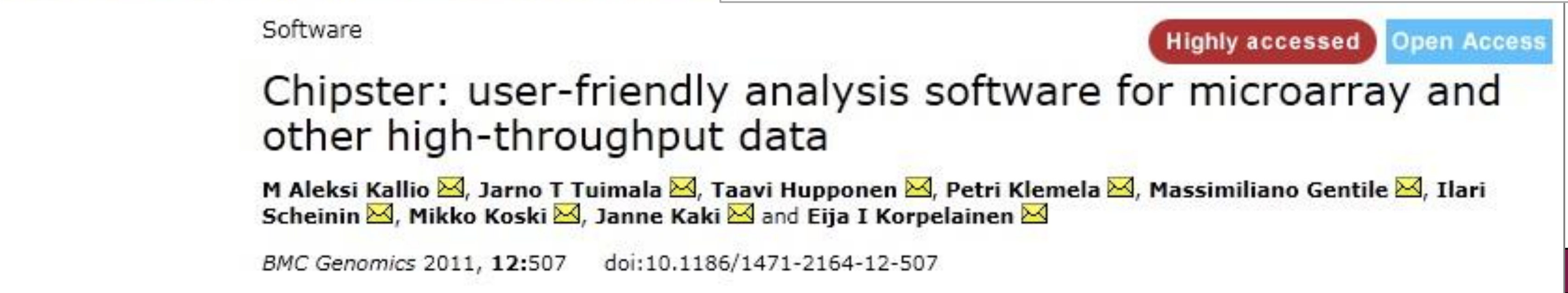


Why GitHub? Enterprise Exp

chipster / chipster

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

8,765 commits · 27 branches · 224 releases · 16 contributors



BMC Genomics

IMPACT FACTOR 4.21

home | journals A-Z | subject areas | advanced search | authors |

Software

Highly accessed · Open Access

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

M Aleksi 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

Microarray data analysis

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Importing data

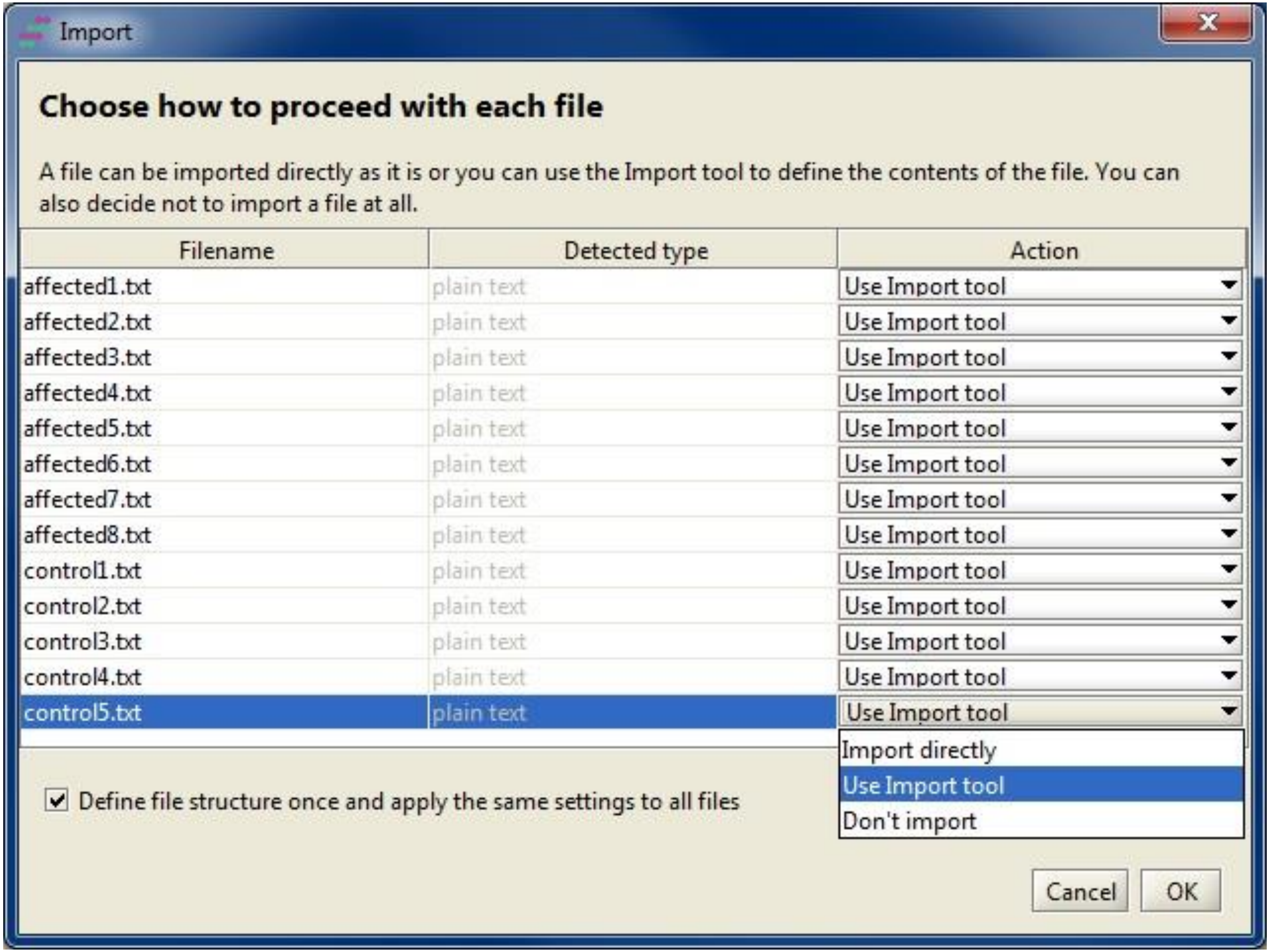
- **Affymetrix**
  - CEL-files are recognized by Chipster automatically

- **Illumina: two importing options**
  1. Import the GenomeStudio file as it is
     - All the samples need to be in one file.
     - Need columns AVG, BEAD_STDERR, Avg_NBEADS and DetectionPval
     - When imported this way, the data has to be normalized in Chipster using the lumi method
  2. Use Import tool to define the sample columns in the file(s)
     - Use the tool ”Normalization / Illumina” to normalize the data

  → **The import option influences your normalization options later**

- **Agilent (and any other tab delimited files)**
  - Use Import tool to define the sample columns

1. Import tool: Select what to do



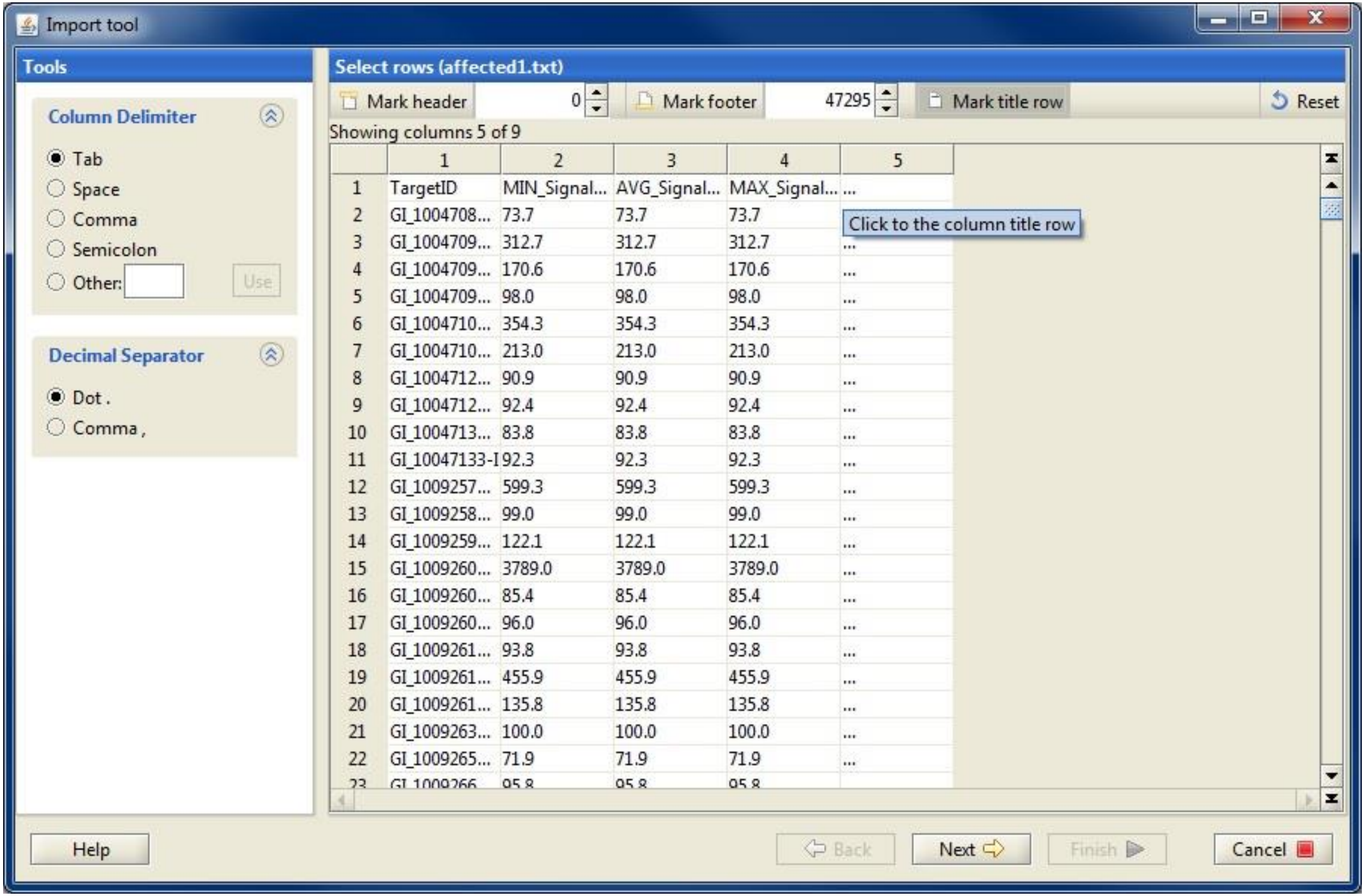
Choose how to proceed with each file

A file can be imported directly as it is or you can use the Import tool to define the contents of the file. You can also decide not to import a file at all.

Filename	Detected type	Action
affected1.txt	plain text	Use Import tool
affected2.txt	plain text	Use Import tool
affected3.txt	plain text	Use Import tool
affected4.txt	plain text	Use Import tool
affected5.txt	plain text	Use Import tool
affected6.txt	plain text	Use Import tool
affected7.txt	plain text	Use Import tool
affected8.txt	plain text	Use Import tool
control1.txt	plain text	Use Import tool
control2.txt	plain text	Use Import tool
control3.txt	plain text	Use Import tool
control4.txt	plain text	Use Import tool
control5.txt	plain text	Use Import tool

☑ Define file structure once and apply the same settings to all files

2. Import tool: Define rows (header, title, etc)



Column Delimiter

● Tab
○ Space
○ Comma
○ Semicolon
○ Other:

Decimal Separator

● Dot .
○ Comma ,

Mark header 0

Mark footer 47295

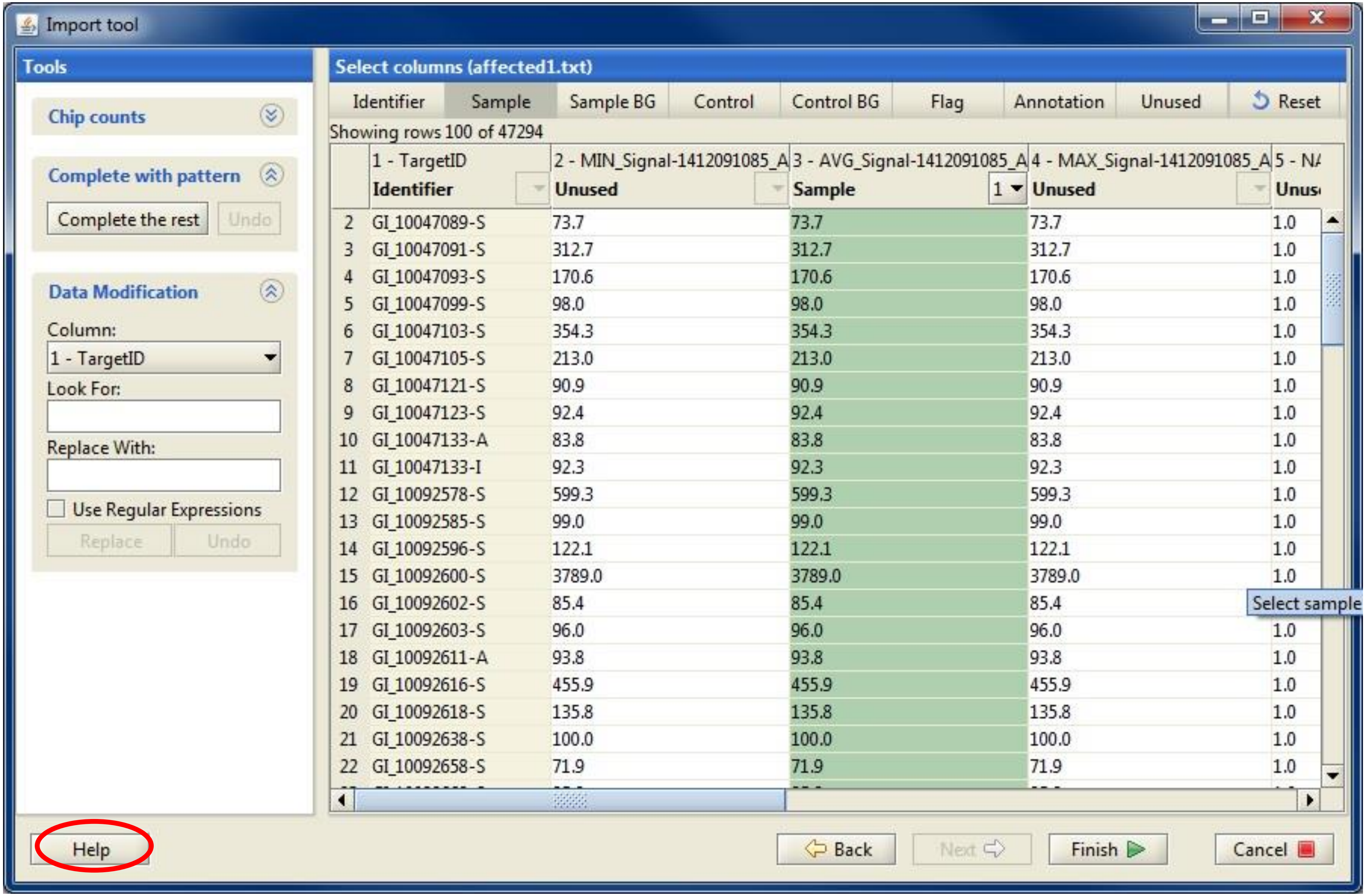
Mark title row

Showing columns 5 of 9

	1	2	3	4	5
1	TargetID	MIN_Signal...	AVG_Signal...	MAX_Signal...	...
2	GI_1004708...	73.7	73.7	73.7	
3	GI_1004709...	312.7	312.7	312.7	...
4	GI_1004709...	170.6	170.6	170.6	...
5	GI_1004709...	98.0	98.0	98.0	...
6	GI_1004710...	354.3	354.3	354.3	...
7	GI_1004710...	213.0	213.0	213.0	...
8	GI_1004712...	90.9	90.9	90.9	...
9	GI_1004712...	92.4	92.4	92.4	...
10	GI_1004713...	83.8	83.8	83.8	...
11	GI_10047133-I	92.3	92.3	92.3	...
12	GI_1009257...	599.3	599.3	599.3	...
13	GI_1009258...	99.0	99.0	99.0	...
14	GI_1009259...	122.1	122.1	122.1	...
15	GI_1009260...	3789.0	3789.0	3789.0	...
16	GI_1009260...	85.4	85.4	85.4	...
17	GI_1009260...	96.0	96.0	96.0	...
18	GI_1009261...	93.8	93.8	93.8	...
19	GI_1009261...	455.9	455.9	455.9	...
20	GI_1009261...	135.8	135.8	135.8	...
21	GI_1009263...	100.0	100.0	100.0	...
22	GI_1009265...	71.9	71.9	71.9	...
23	GI_1009266...	95.8	95.8	95.8	

Click to the column title row

3. Import tool: Define columns (identifier,sample)



Column: 1 - TargetID

Showing rows 100 of 47294

	1 - TargetID Identifier	2 - MIN_Signal-1412091085_A Unused	3 - AVG_Signal-1412091085_A Sample 1	4 - MAX_Signal-1412091085_A Unused	5 - N/ Unus
2	GI_10047089-S	73.7	73.7	73.7	1.0
3	GI_10047091-S	312.7	312.7	312.7	1.0
4	GI_10047093-S	170.6	170.6	170.6	1.0
5	GI_10047099-S	98.0	98.0	98.0	1.0
6	GI_10047103-S	354.3	354.3	354.3	1.0
7	GI_10047105-S	213.0	213.0	213.0	1.0
8	GI_10047121-S	90.9	90.9	90.9	1.0
9	GI_10047123-S	92.4	92.4	92.4	1.0
10	GI_10047133-A	83.8	83.8	83.8	1.0
11	GI_10047133-I	92.3	92.3	92.3	1.0
12	GI_10092578-S	599.3	599.3	599.3	1.0
13	GI_10092585-S	99.0	99.0	99.0	1.0
14	GI_10092596-S	122.1	122.1	122.1	1.0
15	GI_10092600-S	3789.0	3789.0	3789.0	1.0
16	GI_10092602-S	85.4	85.4	85.4	
17	GI_10092603-S	96.0	96.0	96.0	1.0
18	GI_10092611-A	93.8	93.8	93.8	1.0
19	GI_10092616-S	455.9	455.9	455.9	1.0
20	GI_10092618-S	135.8	135.8	135.8	1.0
21	GI_10092638-S	100.0	100.0	100.0	1.0
22	GI_10092658-S	71.9	71.9	71.9	1.0

Import tool - which columns should I mark?

- **http://chipster.csc.fi/manual/import-help.html**

- **Agilent**
  - **Identifier (ProbeName, in case of miRNA arrays use GeneName)**
  - **Annotation (Control type)**
  - **Sample (rMeanSignal or rMedianSignal)**
  - **Sample background (rBGMedianSignal)**
  - **Control (gMeanSignal or gMedianSignal)**
  - **Control background (gBGMedianSignal)**

1-color

2-color

- **Illumina BeadStudio version 3 file and GenomeStudio files**
  - **Identifier (ProbeID)**
  - **Sample (text “AVG”)**

- **Illumina BeadStudio version 1-2 file**
  - **Identifier (TargetID)**
  - **Sample (text “AVG”)**

Importing normalized data

- **The data should be tab delimited and preferably log-transformed**
  - If your data is not log-transformed, you can transform it with the tool “Change interpretation”

- **Import the data file to Chipster using the Import tool. Mark the identifier column and all the sample columns.**

- **Run the tool Normalize / Process prenormalized. This**
  - Converts data to Chipster format by adding ”chip.” to expression column names
  - Creates the phenodata file. You need to indicate the chiptype using names given at http://chipster.csc.fi/manual/supported-chips.html

Exercise 1. Start Chipster and open a session with Affymetrix .CEL-files

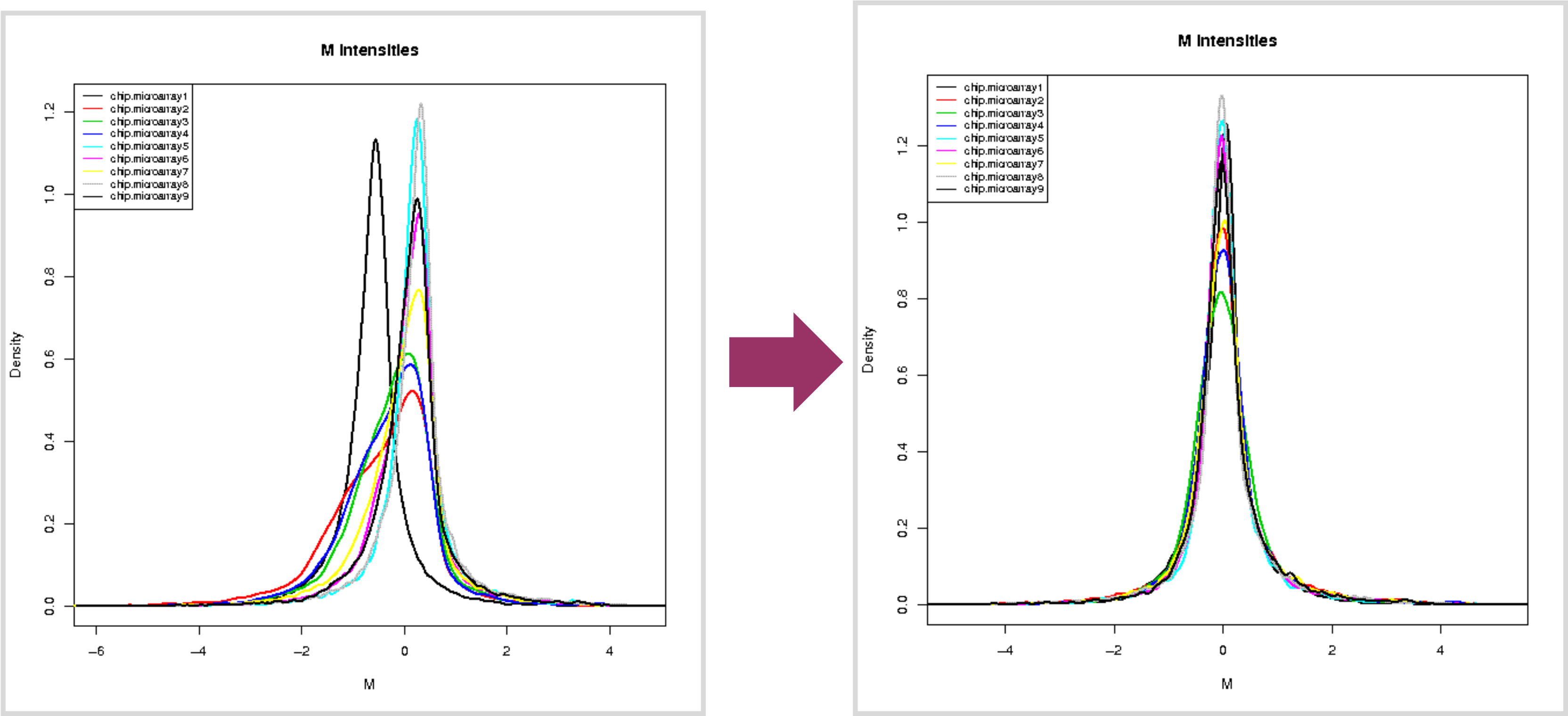
- **Log in to Chipster**
- **Open session containing course data**
  - Select **Open local session** and choose **Affymetrix_kidney_cancer**. The course data contains 17 samples from a kidney cancer study, measured using Affymetrix U133A chips. We want to find genes which are differentially expressed in cancer vs normal tissue.

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Normalization

- **The goal is to make the arrays comparable to each other**
  - Makes the expression value distributions similar
  - Assumes that most genes don’t change expression
- **After normalization the expression values are in <u>log2-scale</u>**
  - Hence for example a fold change of 2 means 4-fold up



Normalization of Affymetrix data

- **Normalization = background correction + expression estimation + summarization**

- **Methods**
  - **RMA** (Robust Multichip Averaging) uses only PM probes, fits a model to them, and gives out expression values after quantile normalization and median polishing. Works nicely if you have more than a few chips
  - **GCRMA** is similar to RMA, but takes also GC% content into account
  - **MAS5** is the older Affymetrix method, **Plier** is a newer one
  - **Li-Wong** is the method implemented in dChip

- **Custom chiptype parameter to use remapped probe information**
  - Because some of the Affymetrix probe-to-transcript mappings can be outdated, probes have been remapped in the Bioconductor project.
  - To use these remappings (alt CDF environments), select the matching chiptype from the Custom chiptype menu.

- **Variance stabilization option makes the variance similar over all the chips**
  - Works only with MAS5 and Plier (the other methods log2-transform the data, which corrects for the same phenomenon)

Quantile normalization procedure

	Sample A	Sample B	Sample C
Gene 1	20	10	350
Gene 2	100	500	200
Gene 3	300	400	30

1. Raw data

	Sample A	Sample B	Sample C	Median
Quantile 1	20	10	30	20
Quantile 2	100	400	200	200
Quantile 3	300	500	350	350

2. Rank data within sample and calculate median intensity for each row

	Sample A	Sample B	Sample C	Median
Quantile 1	20	20	20	20
Quantile 2	200	200	200	200
Quantile 3	350	350	350	350

3. Replace the raw data of each row with its median (or mean) intensity

	Sample A	Sample B	Sample C
Gene 1	20	20	350
Gene 2	200	350	200
Gene 3	350	200	20

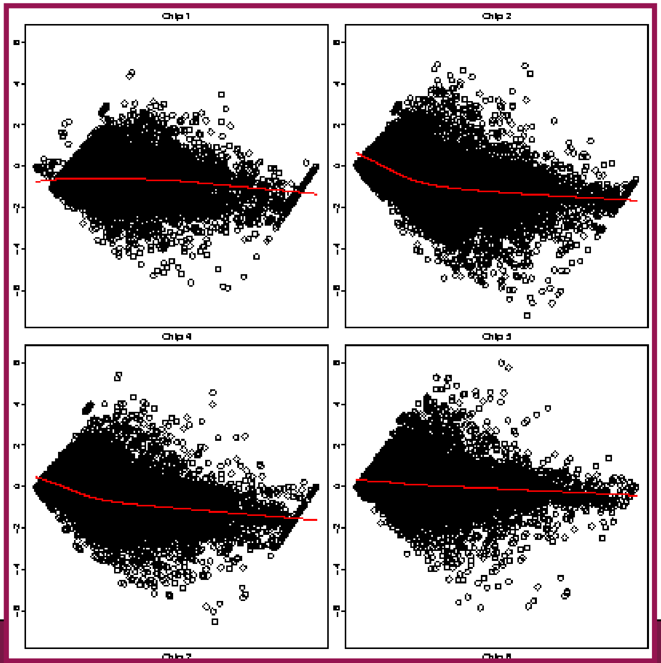
4. Restore the original gene order

Normalization of Agilent data

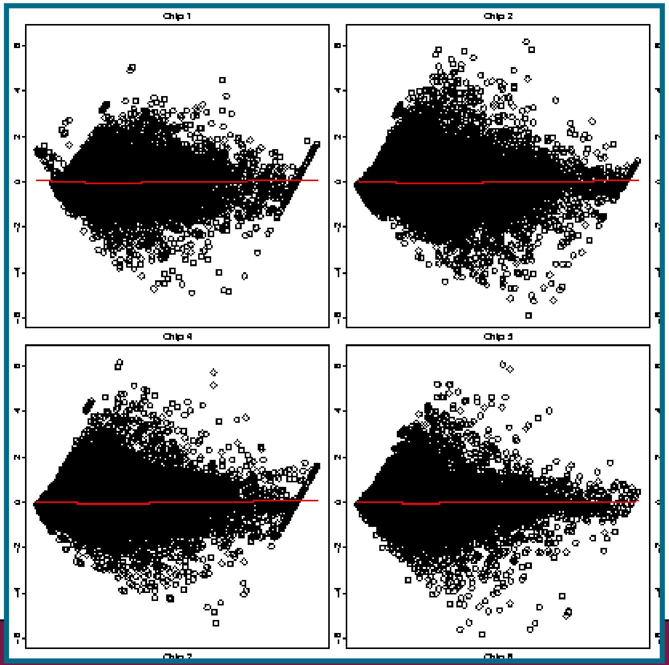
- **Background correction + averaging duplicate spots + normalization**

- **Background subtraction often generates negative values, which are coded as missing values after log2-transformation.**
    - Using normexp + offset 50 will not generate negative values, and it gives good estimates
- **Loess removes curvature from the data (recommended)**

Before



After



Agilent normalization parameters in Chipster

- **Background treatment**
  - Normexp, Subtract, Edwards, None
- **Background offset**
  - 50 or 0
- **Normalize chips**
  - Loess, median, none
- **Chiptype**
  - You must give this information in order to use annotation-based tools later
- **Normalize genes**
  - None, scale (to median), quantile
  - not needed for statistical analysis

Illumina normalization: two analysis tools

1. **Illumina**
   - Normalization method

     Quantile, vsn (variance stabilizing normalization), scale, none
   - Illumina software version

     GenomeStudio or BeadStudio3, BeadStudio2, BeadStudio1
   - Chiptype
   - Identifier type

     Probe ID (for BeadStudio version 3 data and newer), Target ID

2. **Lumi pipeline (data needs to be in one file, imported directly!)**
   - Normalization method

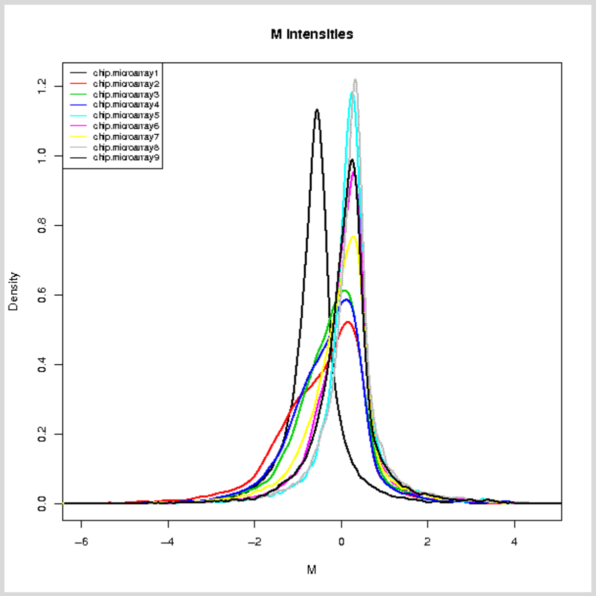
     Quantile, vsn, rsn (robust spline normalization), loess, none
   - Transformation

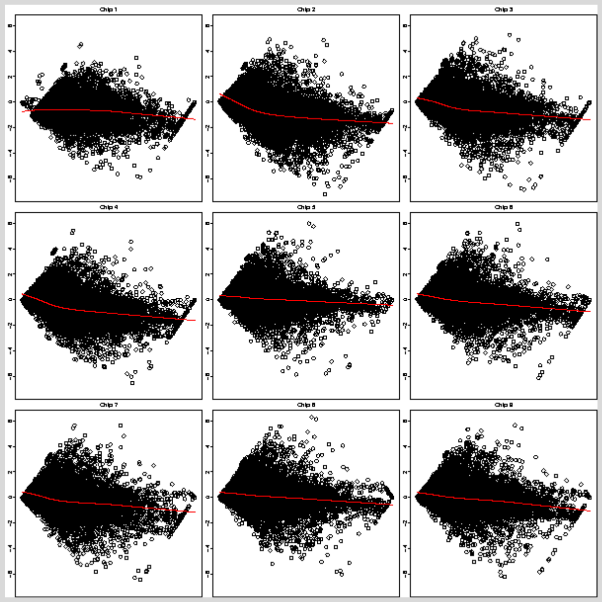
     Log2, vst (variance stabilizing transformation), none
   - Chiptype

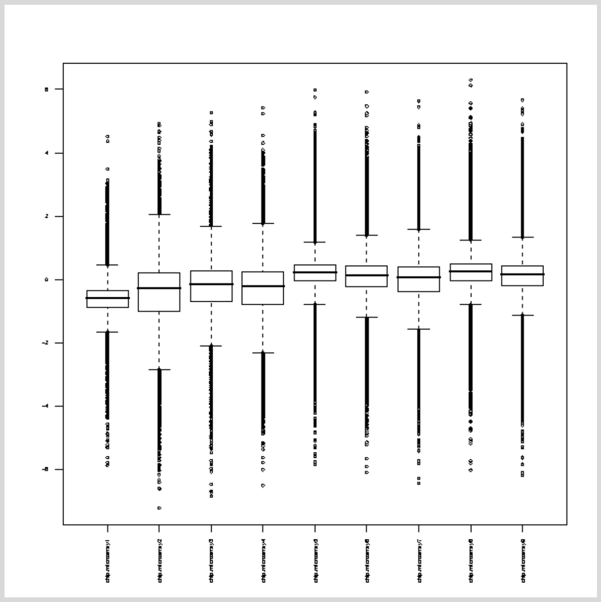
     human, mouse, rat
   - Background correction (usually done already in GenomeStudio)

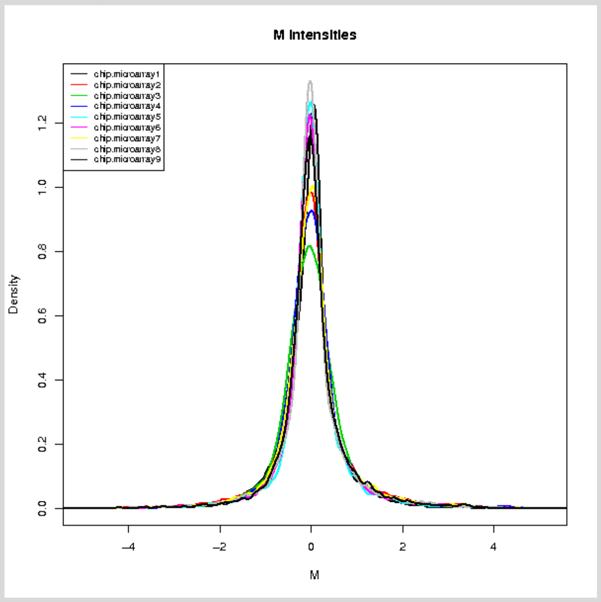
     none, bgAdjust.Affy

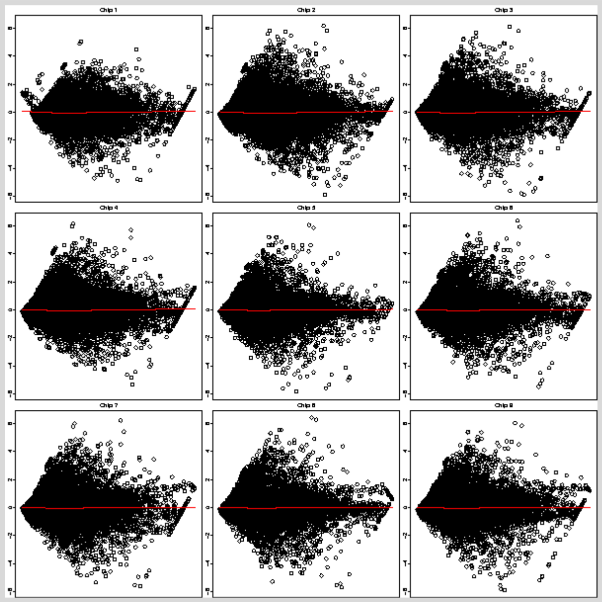
Checking normalization

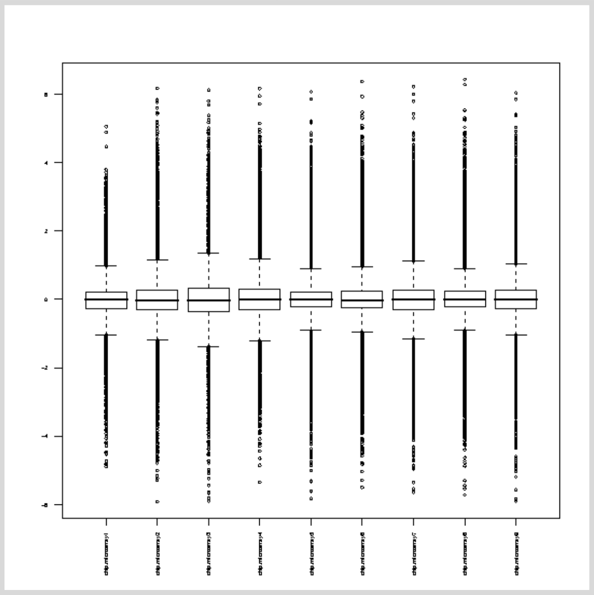












Exercise 2: Normalize Affymetrix data

- Select all the CEL files by clicking on the box ”**17**” in the Workflow view

- Select the tool **Normalisation / Affymetrix**, click **Show parameters**, set **Custom CDF annotation to be used** = **hgu133A**, and click **Run**.

- Repeat the process by setting **Custom CDF annotation to be used** = **Use original Affymetrix annotations**. When the result file **normalized.tsv** comes, rename it to **original_normalized.tsv**

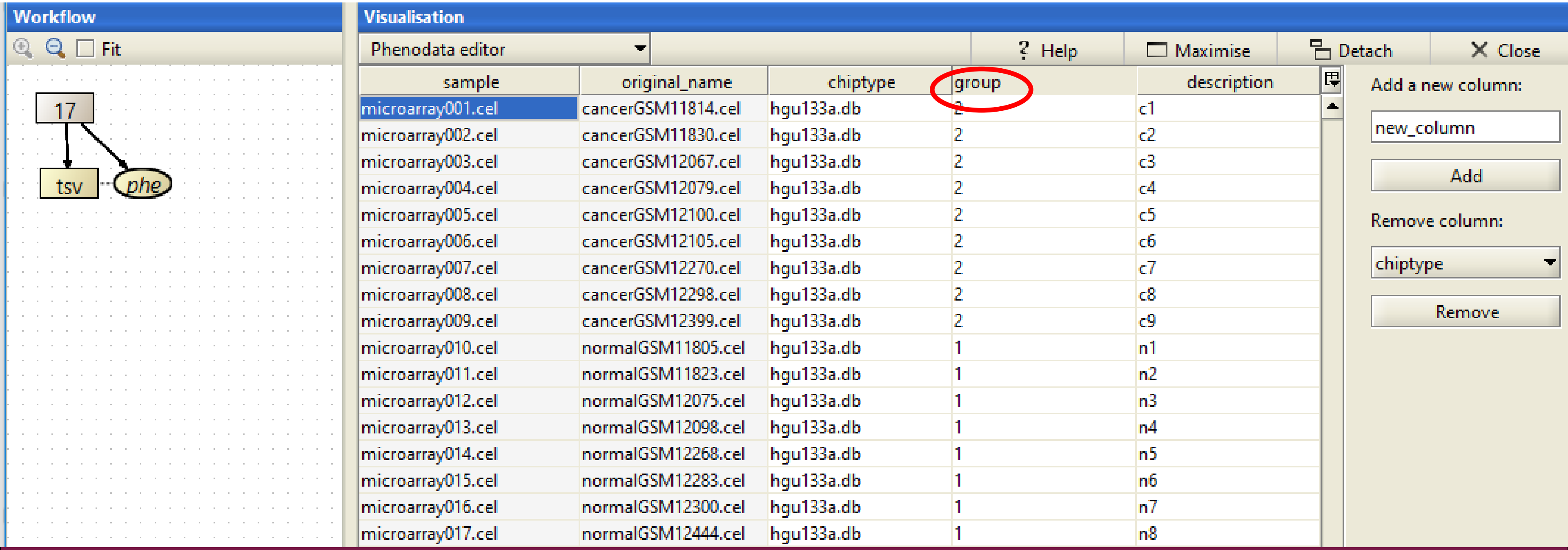
- Open both normalized files and compare them. Do they have the same number of genes (rows)?

Microarray data analysis workflow

- Importing data to Chipster
- Normalization
- **Describing samples with a phenodata file**
- Quality control
  - Array level
  - Experiment level
- Filtering (optional)
- Statistical testing
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- Annotation
- Pathway analysis
- Clustering
- Saving the workflow

Phenodata file

- **Experimental setup is described with a phenodata file, which is created during normalization**
- **<u>Fill in the group column</u> with numbers describing your experimental groups**
  - e.g. 1 = control sample, 2 = cancer sample
  - necessary for the statistical tests to work
  - note that you can sort a column by clicking on its title
- **Change sample names in Description column for visualizations**



sample	original_name	chiptype	group	description
microarray001.cel	cancerGSM11814.cel	hgu133a.db	2	c1
microarray002.cel	cancerGSM11830.cel	hgu133a.db	2	c2
microarray003.cel	cancerGSM12067.cel	hgu133a.db	2	c3
microarray004.cel	cancerGSM12079.cel	hgu133a.db	2	c4
microarray005.cel	cancerGSM12100.cel	hgu133a.db	2	c5
microarray006.cel	cancerGSM12105.cel	hgu133a.db	2	c6
microarray007.cel	cancerGSM12270.cel	hgu133a.db	2	c7
microarray008.cel	cancerGSM12298.cel	hgu133a.db	2	c8
microarray009.cel	cancerGSM12399.cel	hgu133a.db	2	c9
microarray010.cel	normalGSM11805.cel	hgu133a.db	1	n1
microarray011.cel	normalGSM11823.cel	hgu133a.db	1	n2
microarray012.cel	normalGSM12075.cel	hgu133a.db	1	n3
microarray013.cel	normalGSM12098.cel	hgu133a.db	1	n4
microarray014.cel	normalGSM12268.cel	hgu133a.db	1	n5
microarray015.cel	normalGSM12283.cel	hgu133a.db	1	n6
microarray016.cel	normalGSM12300.cel	hgu133a.db	1	n7
microarray017.cel	normalGSM12444.cel	hgu133a.db	1	n8

How to describe pairing, replicates, time, etc?

- **You can add new columns to the phenodata file**
- **How to describe different variables**
  - **Time:** Use either real time values or recode with group codes
  - **Replicates:** All the replicates are coded with the same number
  - **Pairing:** Pairs are coded using the same number for each pair
  - **Gender:** Use numbers
  - **Anything else:** Use numbers

Creating phenodata for normalized data

- **When you import data which has been already normalized, you need to create a phenodata file for it**
  - Use Import tool to bring the data in
  - Use the tool Normalize / Process prenormalized to create phenodata
    - Remember to give the chiptype
  - Fill in the group column

- **Note: If you already have a phenodata file, you can import it too**
  - Choose ”Import directly” in the Import tool
  - Right click on normalized data, choose ”Link to phenodata”

Exercise 3: Describe the experiment

- **Double click the phenodata file of the normalized.tsv**

- **In the phenodata editor, fill in the group column so that you enter**
  - 1 for normal samples
  - 2 for cancer samples

- **For the interest of visualizations later on, give shorter names for the samples in the Description column**
  - Name the normal samples n1, n2,…
  - Name the cancer samples c1, c2 ,…

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Array level quality control

- **Allows you to check if arrays are comparable to each other**
- **Tools in Chipster**
  - Affymetrix basic: RNA degradation and Affy QC
  - Affymetrix RLE and NUSE: fit a model to expression values
  - Agilent 1-color: density plot and boxplot
  - Agilent 2-color: MA-plot, density plot and boxplot
  - Illumina: density plot and boxplot

Affymetrix array level QC tools

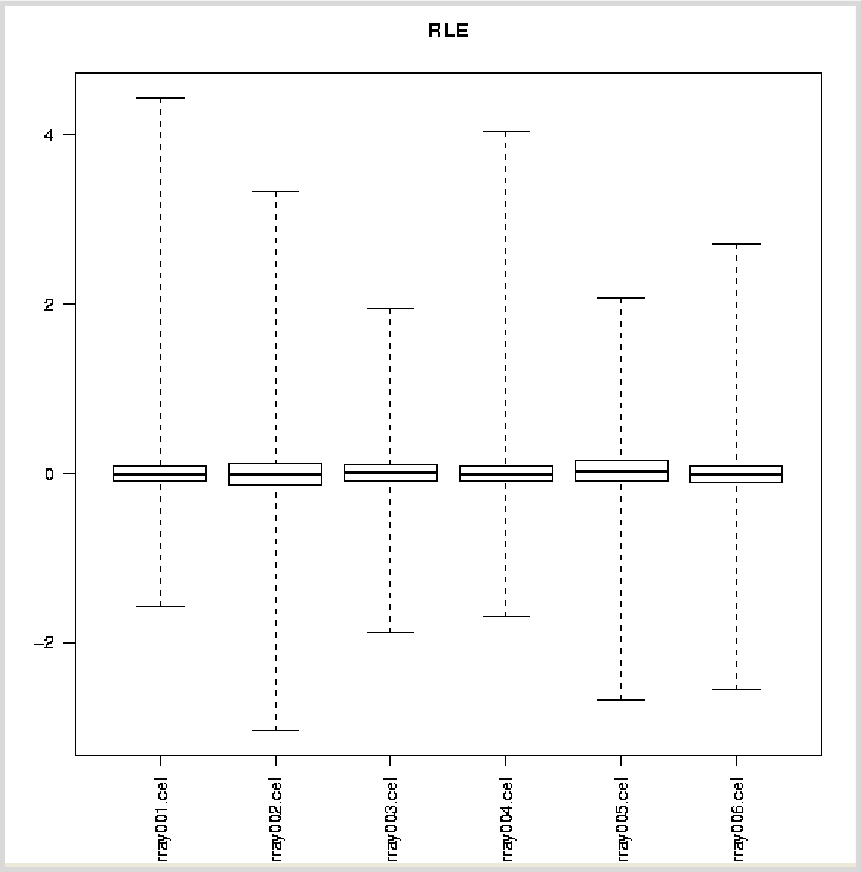
- **Note that these tools use raw data (CEL files), not normalized data**

- **Affymetrix basic**
  - Produces 3 plots:
    - QC stats plot
    - RNA degradation plot
    - Spike-in controls linearity plot
  - Note that this tool uses the original probe set definitions from Affymetrix, not the alternative CDFs

- **Affymetrix RLE and NUSE**
  - RLE (relative log expression)
  - NUSE (normalized unscaled standard error plot)

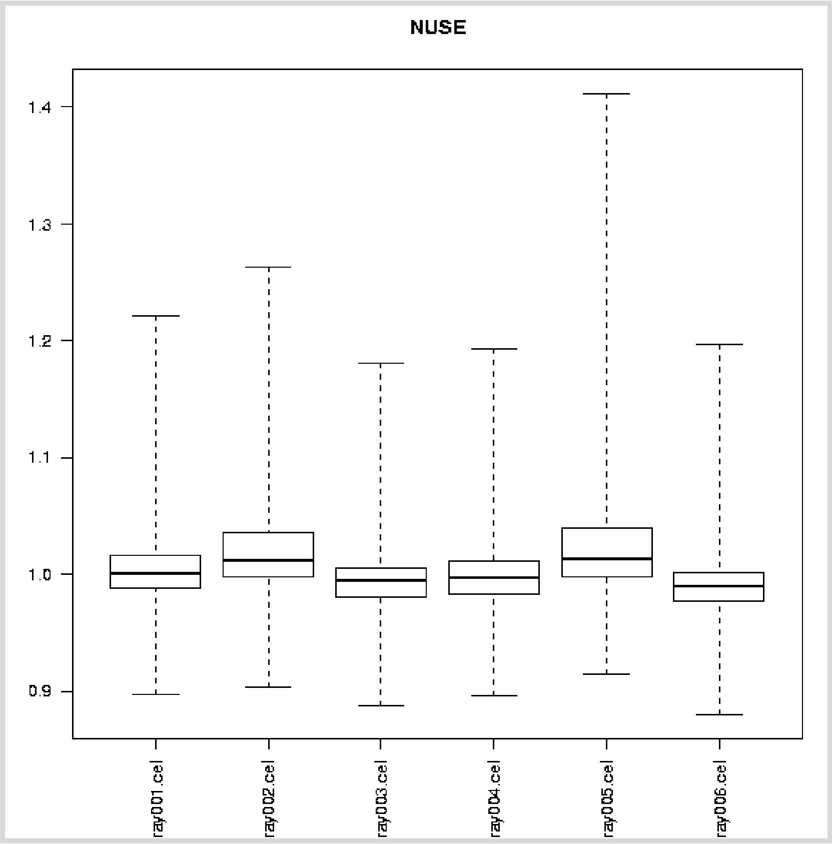
- **Affymetrix RLE and NUSE for exon/gene arrays**

Relative log expression, RLE



- RLE is the difference between log summarized expression of each chip to the log summarized expression on the median chip values.
- Boxes should be centered near 0 and have similar spread.

Normalized Unscaled Standard Error, NUSE

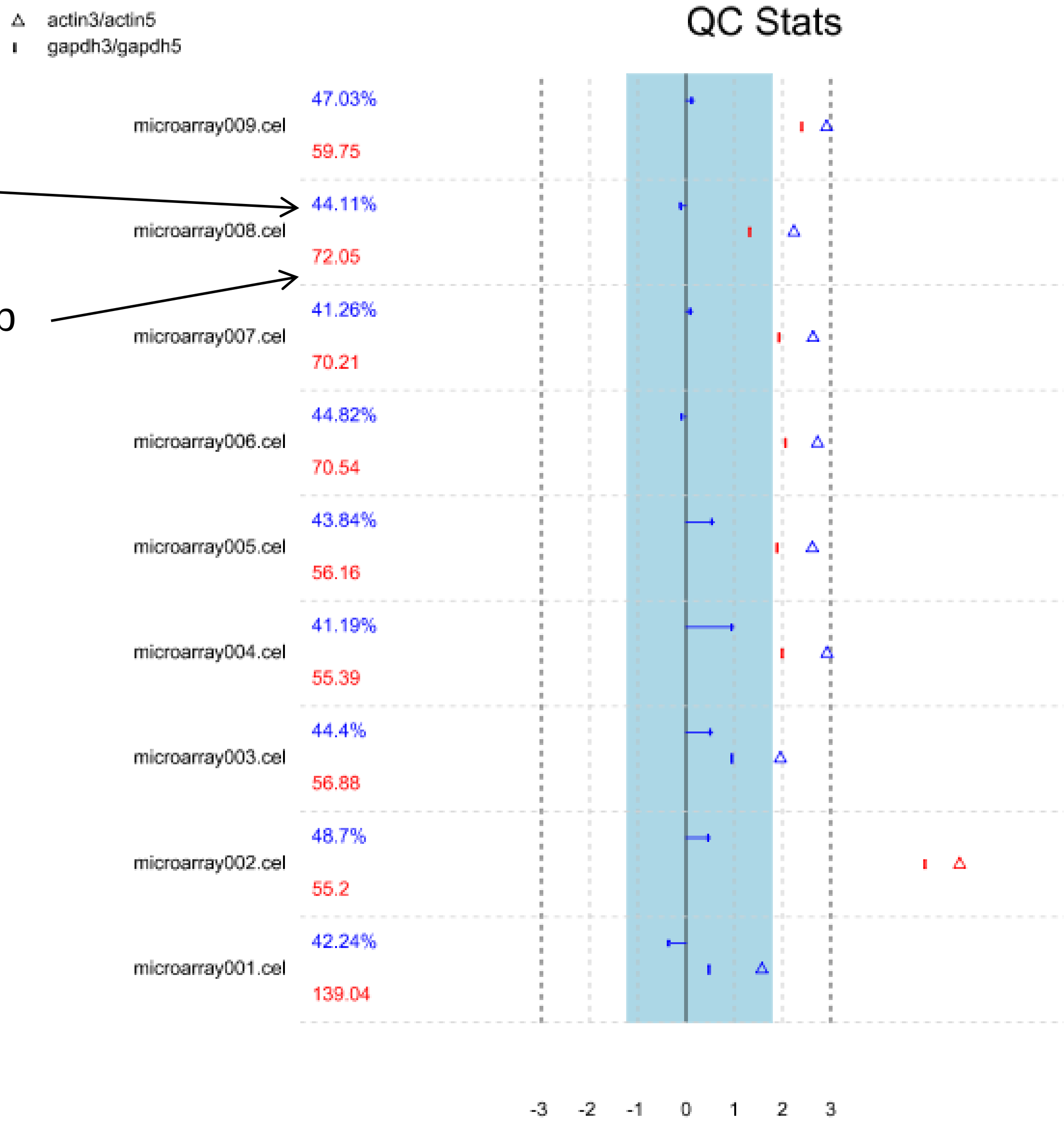


- NUSE is the individual probe error fitting the Probe-Level Model.
- Good chips have median values close to one, while bad ones are above 1.1.
- Check also if some chips show higher spread of NUSE distribution than others.

Affymetrix QC

Proportion of probesets with present flag

average background on the chip



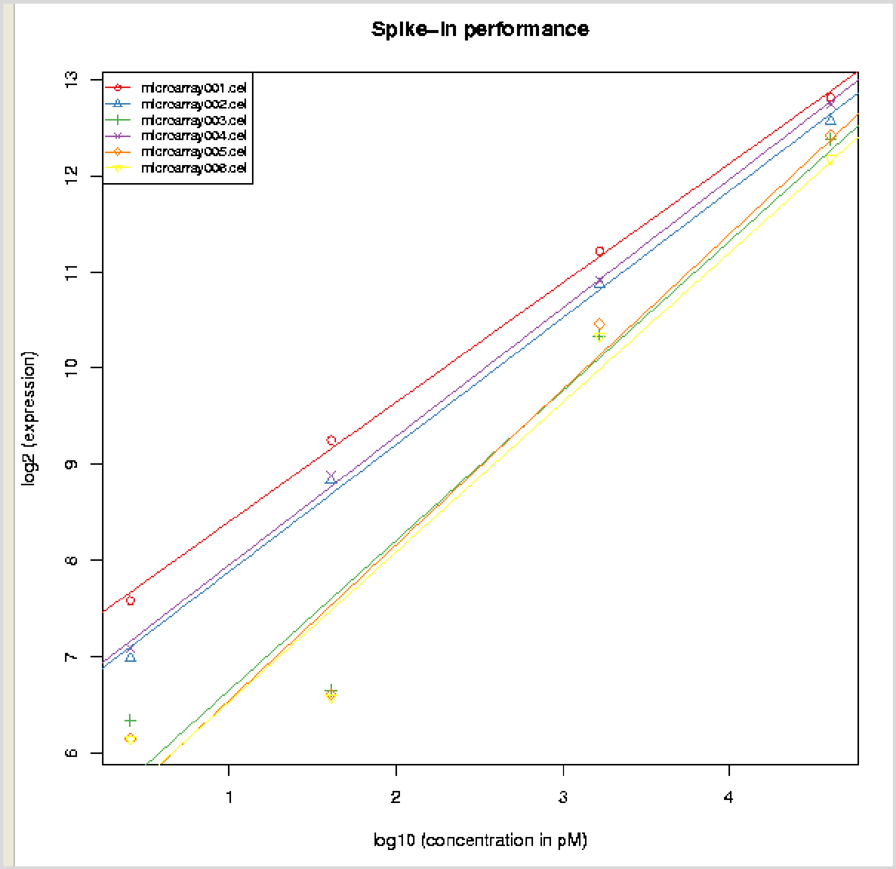
─● scaling factors for the chips
▲ beta-actin 3':5' ratio
● GAPDH 3':5' ratio

Blue area shows where scaling factors are less than 3-fold of the mean.

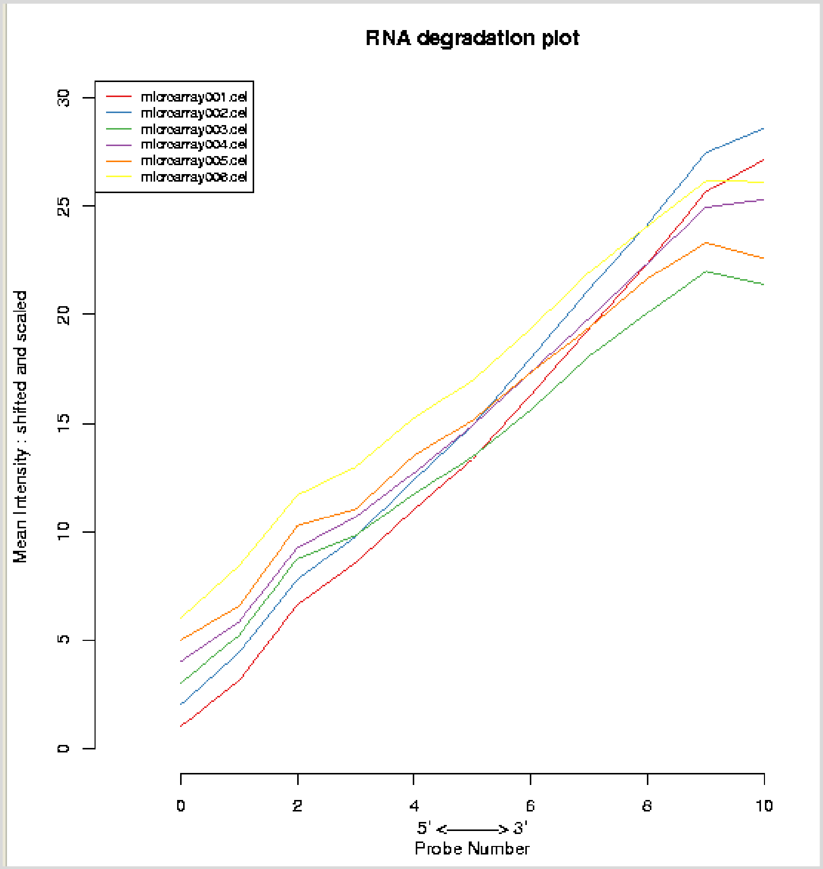
- If the scaling factors or ratios fall within this region (1.25-fold for GAPDH), they are colored **blue,** otherwise **red**

Affymetrix spike-ins and RNA degradation

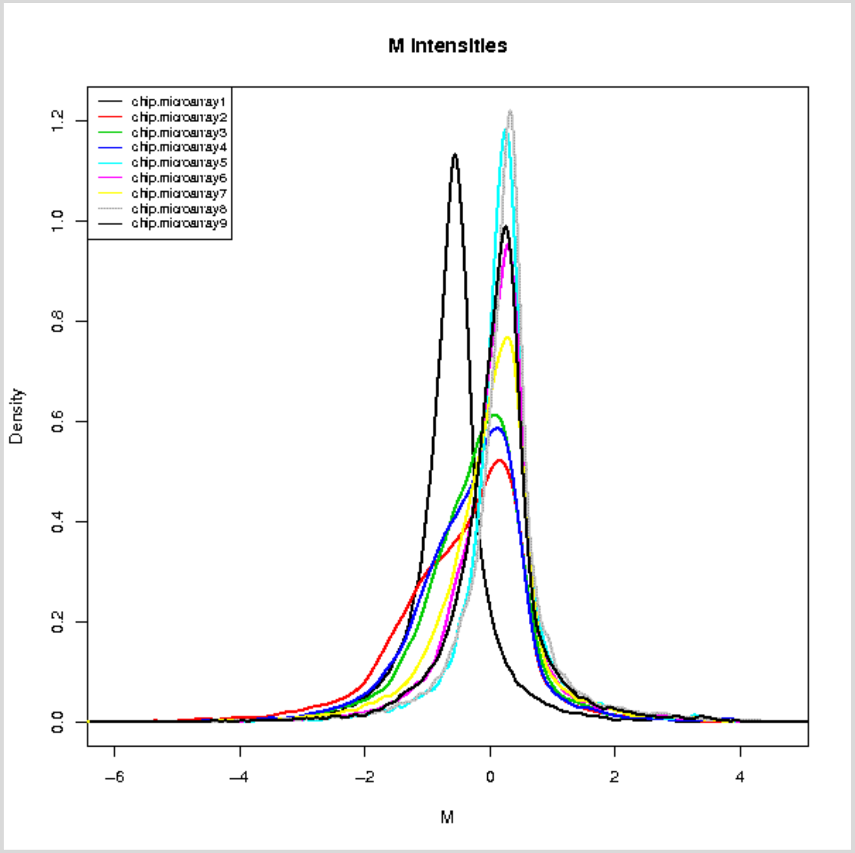
Spike-in linearity

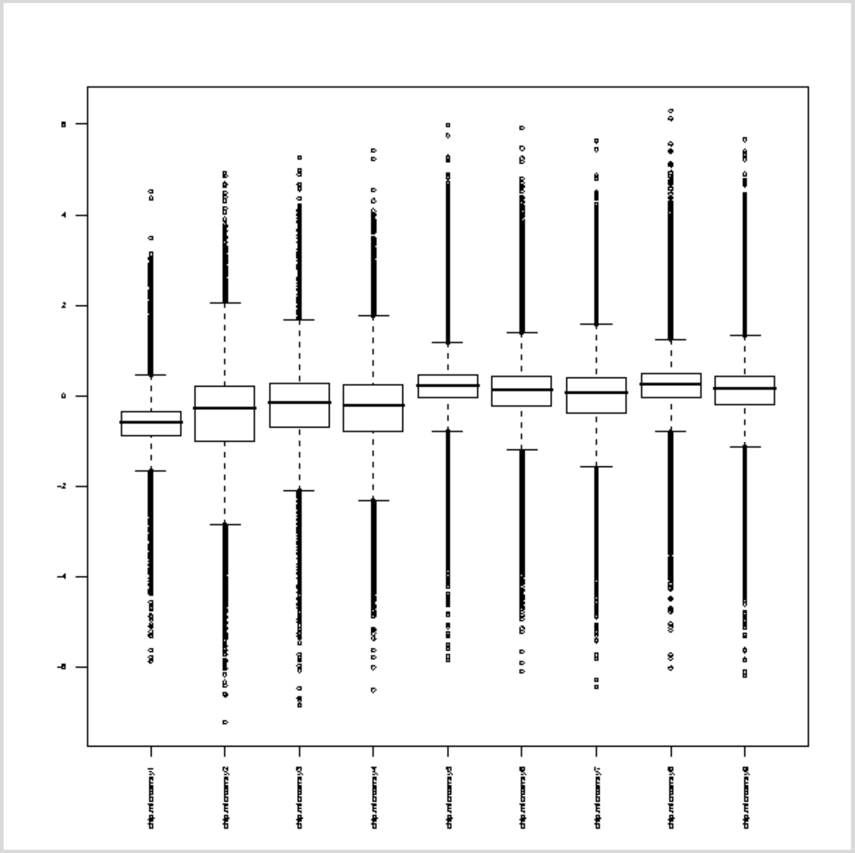


RNA degradation plot

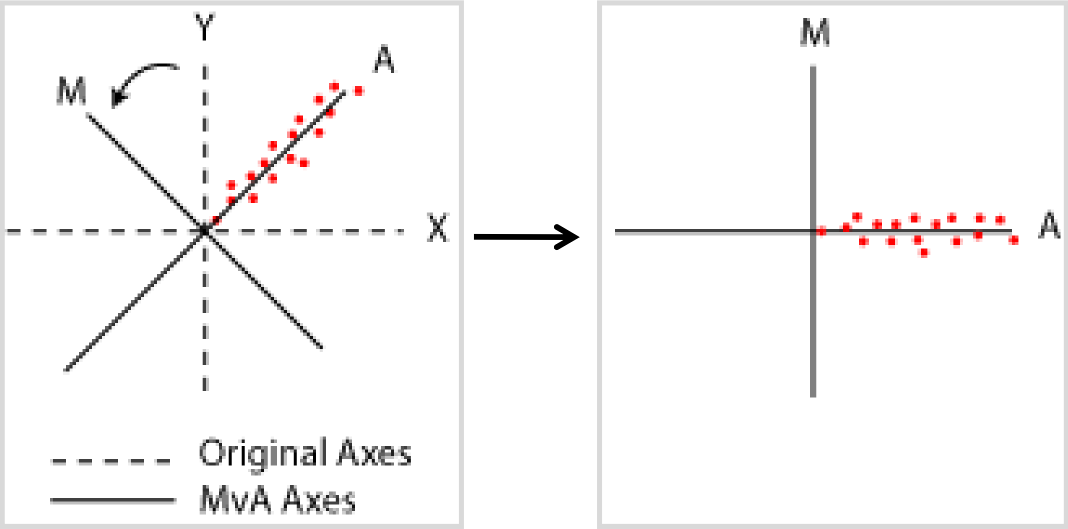


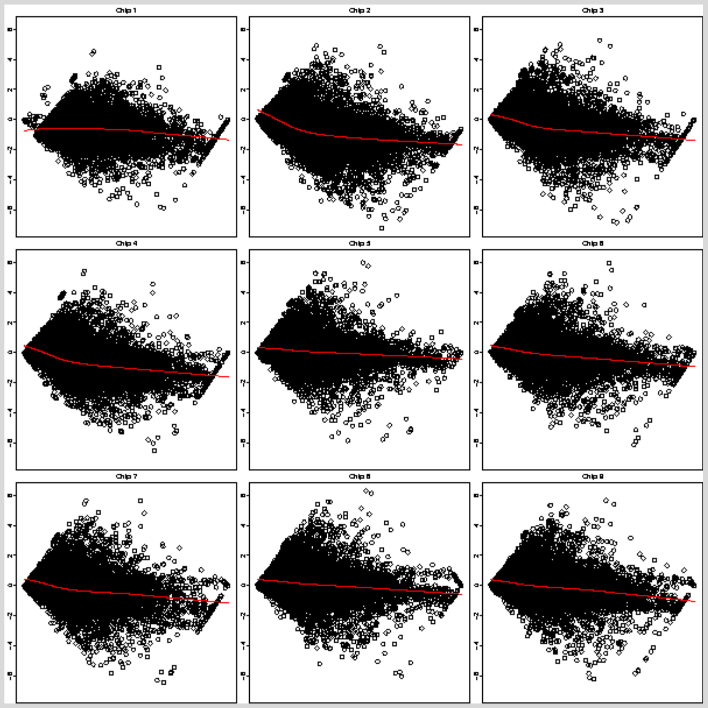
Density plot and box plot





Agilent QC: MA-plot





- Scatter plot of log intensity ratios **M=log2(R/G)** versus average log intensities **A = log2 √(R*G**), where R and G are the intensities for the sample and control, respectively
- M is a mnemonic for minus, as M = log R – log G
- A is mnemonic for add, as A = (log R + log G) / 2

Exercise 4: Affymetrix array level quality control

- Select the **17 CEL files** and run the tool **Quality control / Affymetrix basic**. Please note that this tool uses the original probe set definitions from Affymetrix
  - Inspect the three pdf image files. Are there outlier samples?

- Select the **17 CEL files** and run the tool **Quality control / Affymetrix – using RLE and NUSE** setting **Custom chiptype = hgu133ahsentrezg(hgu133a)**
  - Inspect the RLE and NUSE images. Are there outlier samples?

- Select **normalized.tsv** and run the tool **Quality control / Illumina** which produces a boxplot and density plot
  - Inspect the plots. Are there outlier samples?

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Experiment level quality control

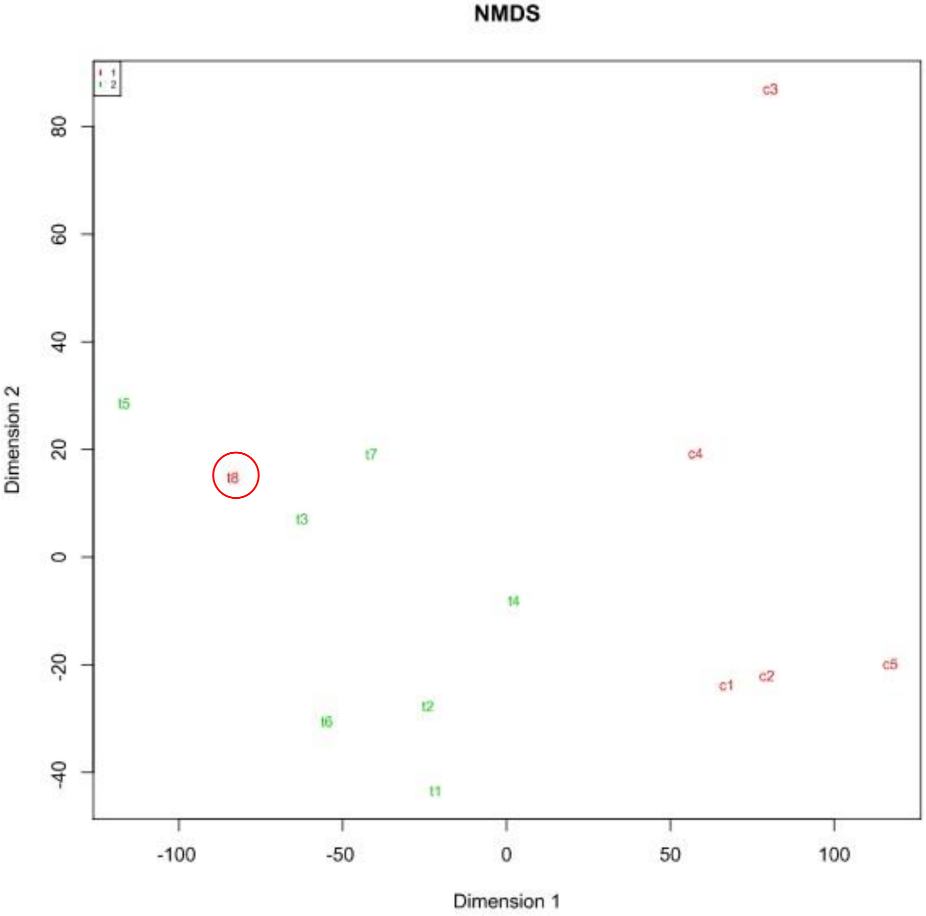
- **Getting an overview of similarities and dissimilarities between samples allows you to check**
  - Do the experimental groups separate from each other?
  - Is there a confounding factor (e.g. batch effect) that should be taken into account in the statistical analysis?
  - Are there sample outliers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

- **Several methods available**
  - NMDS (non-metric multidimentional scaling)
  - PCA (principal component analysis)
  - Clustering
  - Dendrogram
  - Correlogram

Non-metric multidimensional scaling (NMDS)

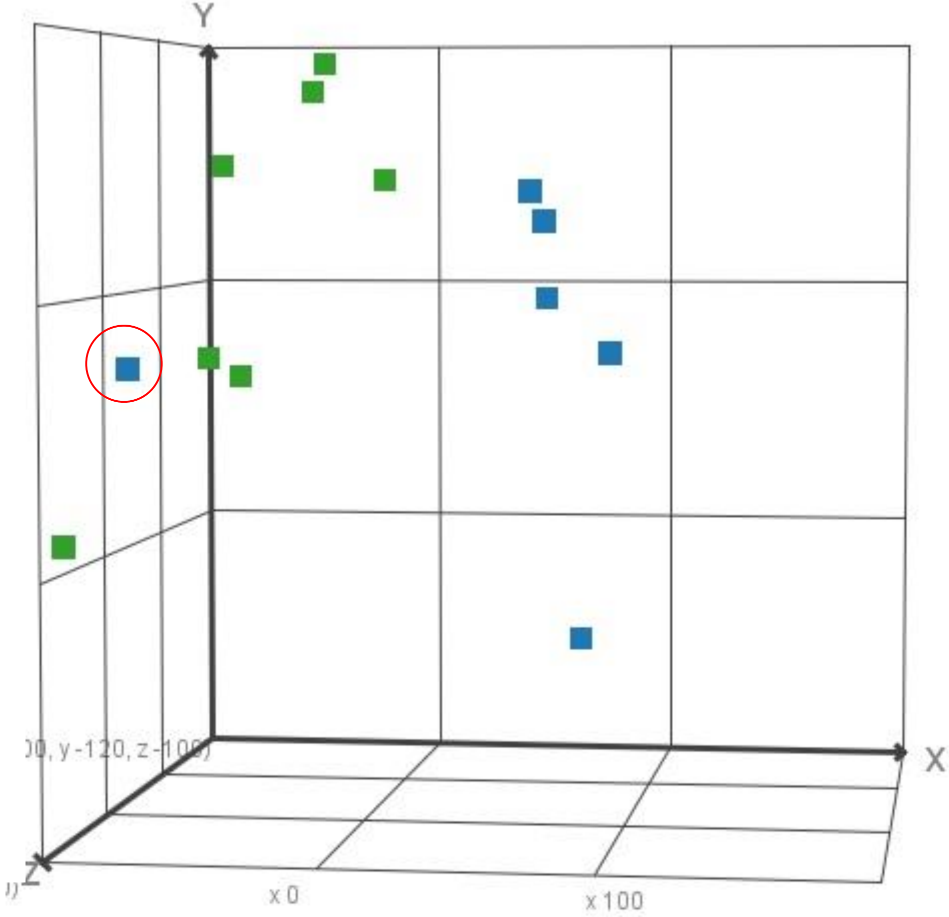
- **Goal is to reduce dimensions from several thousands to two**
  - High dimensional space is projected into a 2-dimensional space
- **Check that the experimental groups separate on dimension 1**
  - Do the samples separate according to something else on dimension 2?

- **Method**
  - Computes a distance matrix for all genes
  - Constructs the dimensions so that the similarity of distances between the original and the 2-dimensional space is maximized



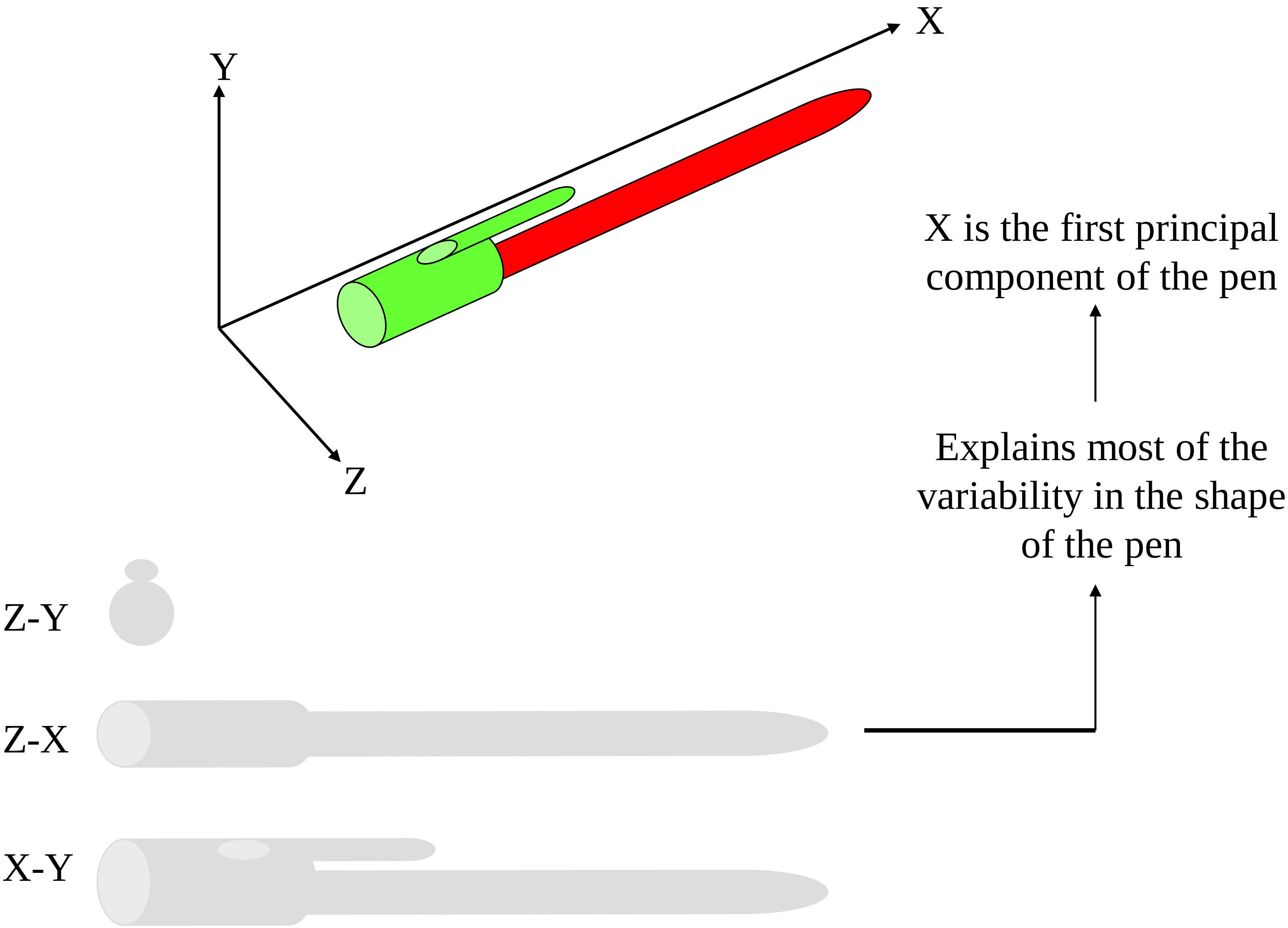
Principal component analysis (PCA)

- **Goal is to reduce dimensions**
  - High dimensional space is projected into a lower dimensional space
- **Check the percentage of variance explained by each component**
  - If PC2 explains only a small percentage of variance, it can be ignored.



- **Method**
  - Computes a variance-covariance matrix for all genes
  - PC1, the first principal component, is the linear combination of variables that maximizes the variance
  - PC2 is a linear combination orthogonal to the previous one which maximizes variance.
  - etc

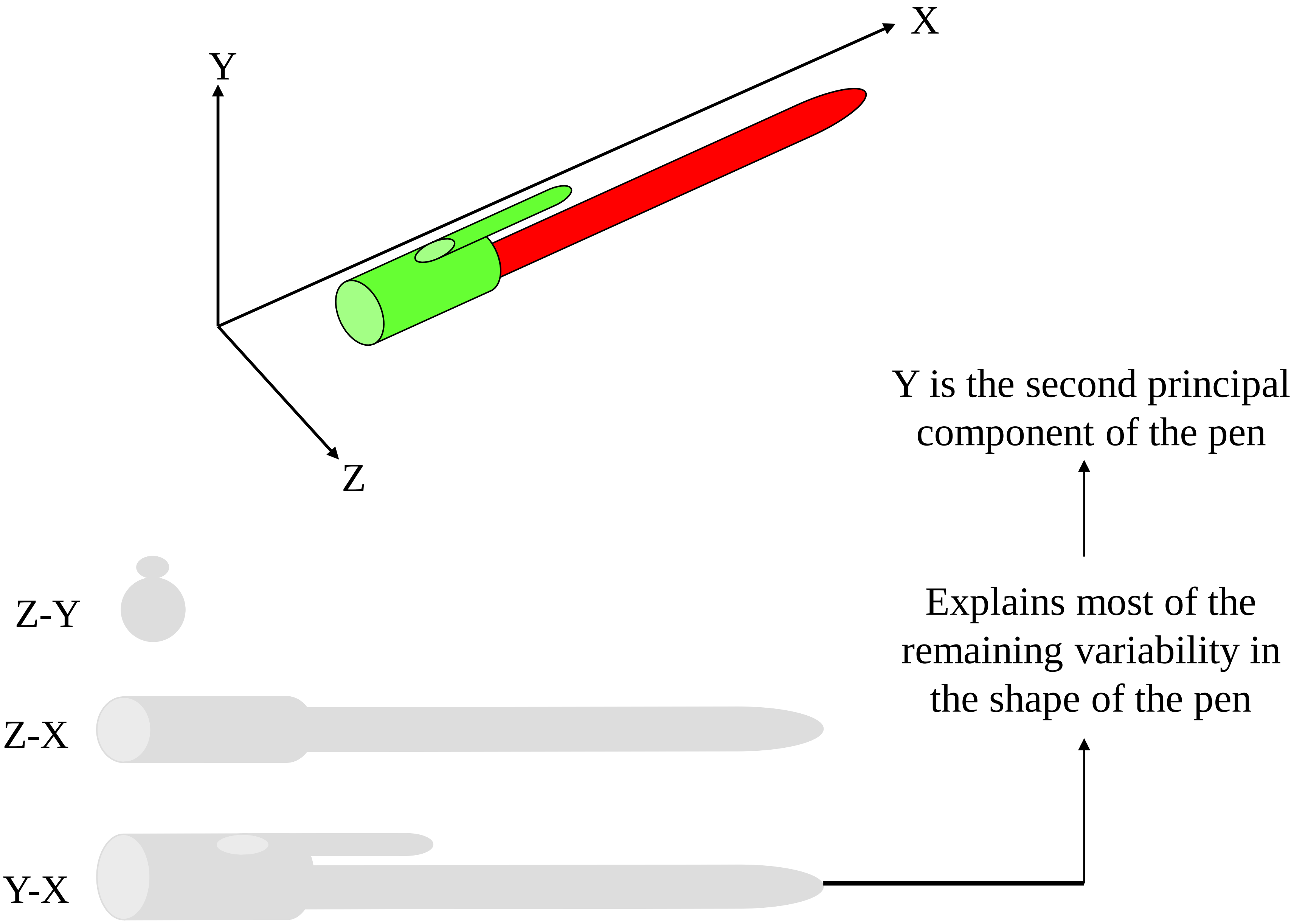
PCA illustration



X is the first principal component of the pen

Explains most of the variability in the shape of the pen

PCA illustration, continued

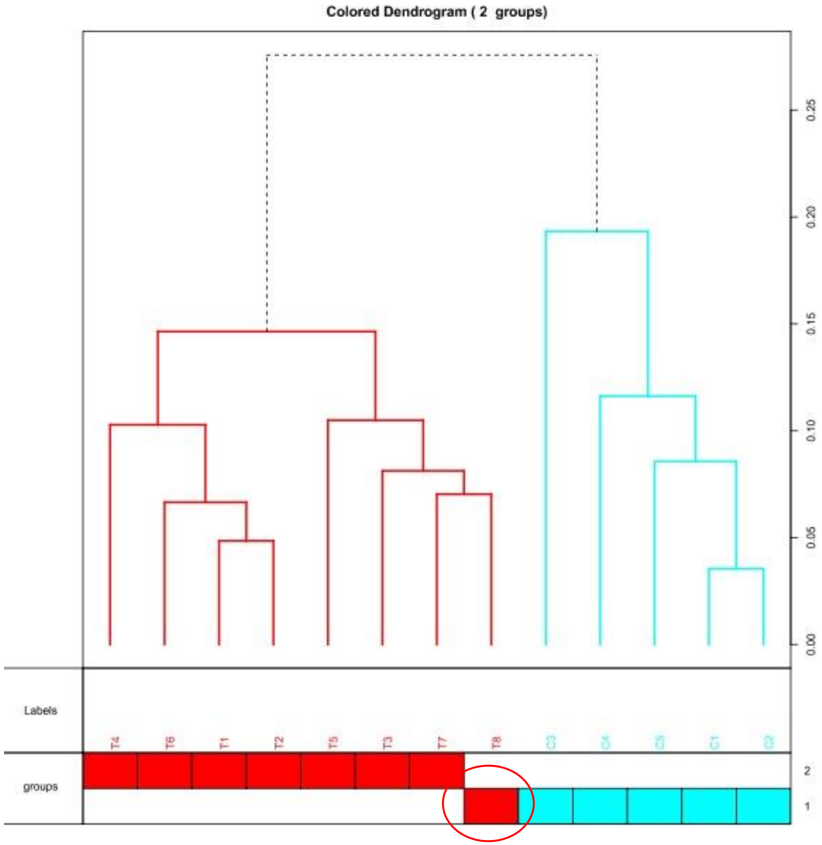


Y is the second principal component of the pen

Explains most of the remaining variability in the shape of the pen

Dendrogram

Dendrogram



Exercise 5: Experiment level quality control

- **Run Statistics / NMDS for the normalized data (normalized.tsv)**
  - Do the groups separate along the first dimension?

- **Run Statistics / PCA on the normalized data.**
  - View **pca.tsv** as **3D scatter plot for PCA**. Can you see 2 groups?
  - Check in **variance.tsv** how much variance the first principal component explains? And the second one?

- **Run Visualization / Dendrogram for the normalized data**
  - Do the groups separate well?

- **Save the analysis session with name sessionKidneyCancer.zip**

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Filtering

- **Why?**
  - Reducing the number of genes tested for differential expression reduces the severity of multiple testing correction of p-values. As the p-values remain better, we detect more differentially expressed genes.
- **Why not?**
  - Some statistical testing methods (inc. the empirical Bayes option in Chipster) need many genes, because they estimate variance by borrowing information from other genes which are expressed at similar level. Hence the more genes the better.

- **Filtering should**
  - remove genes which don’t have any chance of being differentially expressed: genes that are not expressed or don’t change
  - be <u>independent</u>: should not use the sample group information

Filtering tools in Chipster

- **Filter by standard deviation (SD)**
  - Select the percentage of genes to be filtered out

- **Filter by coefficient of variation (CV = SD / mean)**
  - Select the percentage of genes to be filtered out

- **Filter by interquartile range (IQR)**
  - Select the IQR

- **Filter by expression**
  - Select the upper and lower cut-offs
  - Select the number of chips required to fulfil this rule

- **Filter by flag** (Affymetrix P, M and A flags)
  - Flag value and number of arrays

Exercise 6: Filtering

- **Select the normalized data and play with the SD filter and CV filter.**
  - Set the cutoffs so that you filter out 90% of genes (Percentage to filter out = 0.9).
  - Preprocessing / Filter by standard deviation
  - Preprocessing / Filter by coefficient of variation

- **Select the result files and compare them using the interactive Venn diagram visualization**
  - Save the genes specific to SD filter to a new file. Rename it sd.tsv.
  - Save the genes specific to CV filter to a new file. Rename it cv.tsv.
  - View both as expression profiles. Is there a difference in expression levels of the two sets?

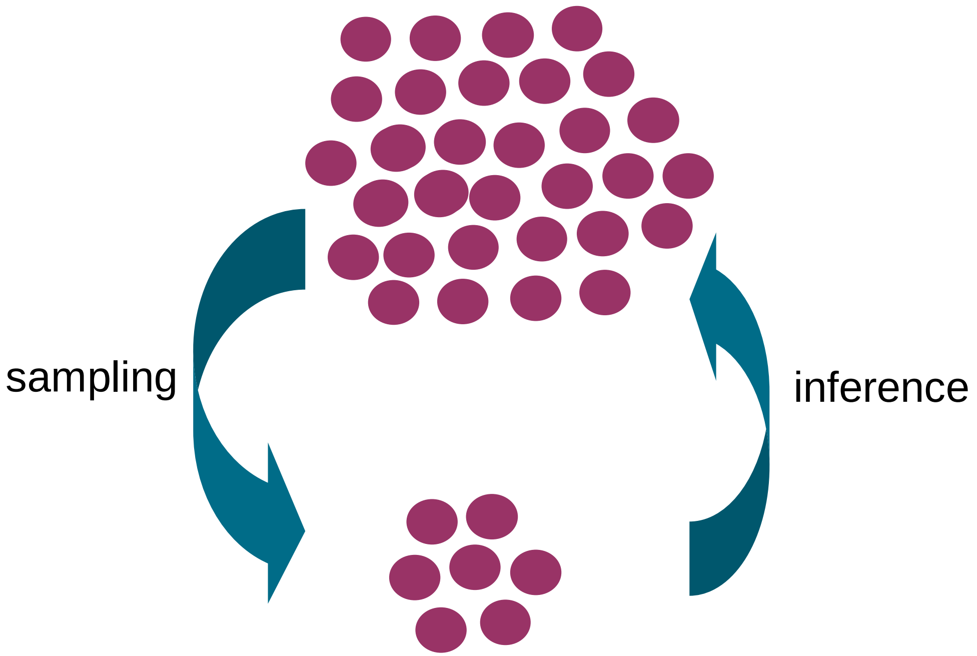
Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Statistical analysis: Why?

- **Distinguish the treatment effect from biological variability and measurement noise**
  - replicates
  - estimation of uncertainty (variability)

- **Generalisation of results**
  - representative sample
  - statistical inference

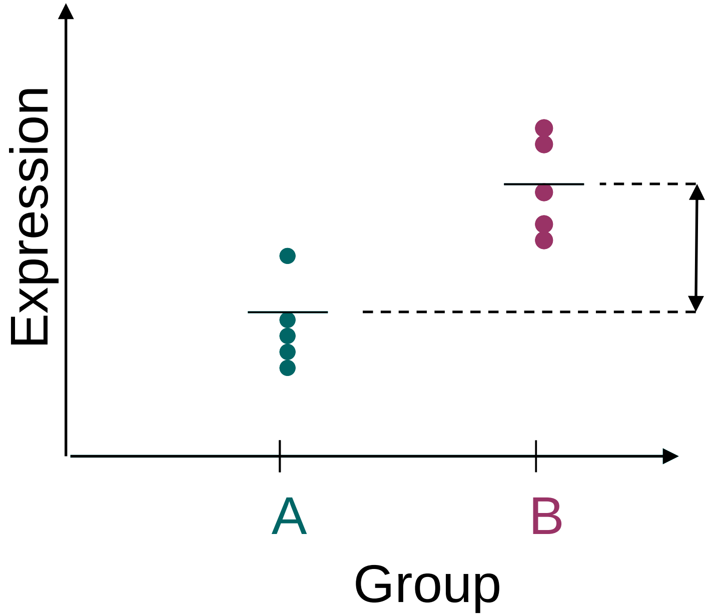


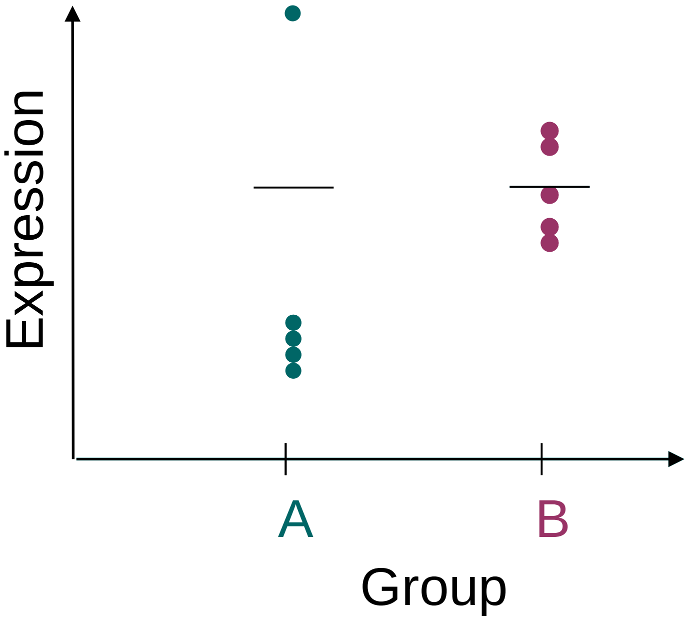
Parametric statistical methods

- **Comparing means of 1-2 groups**
  - student's t-test

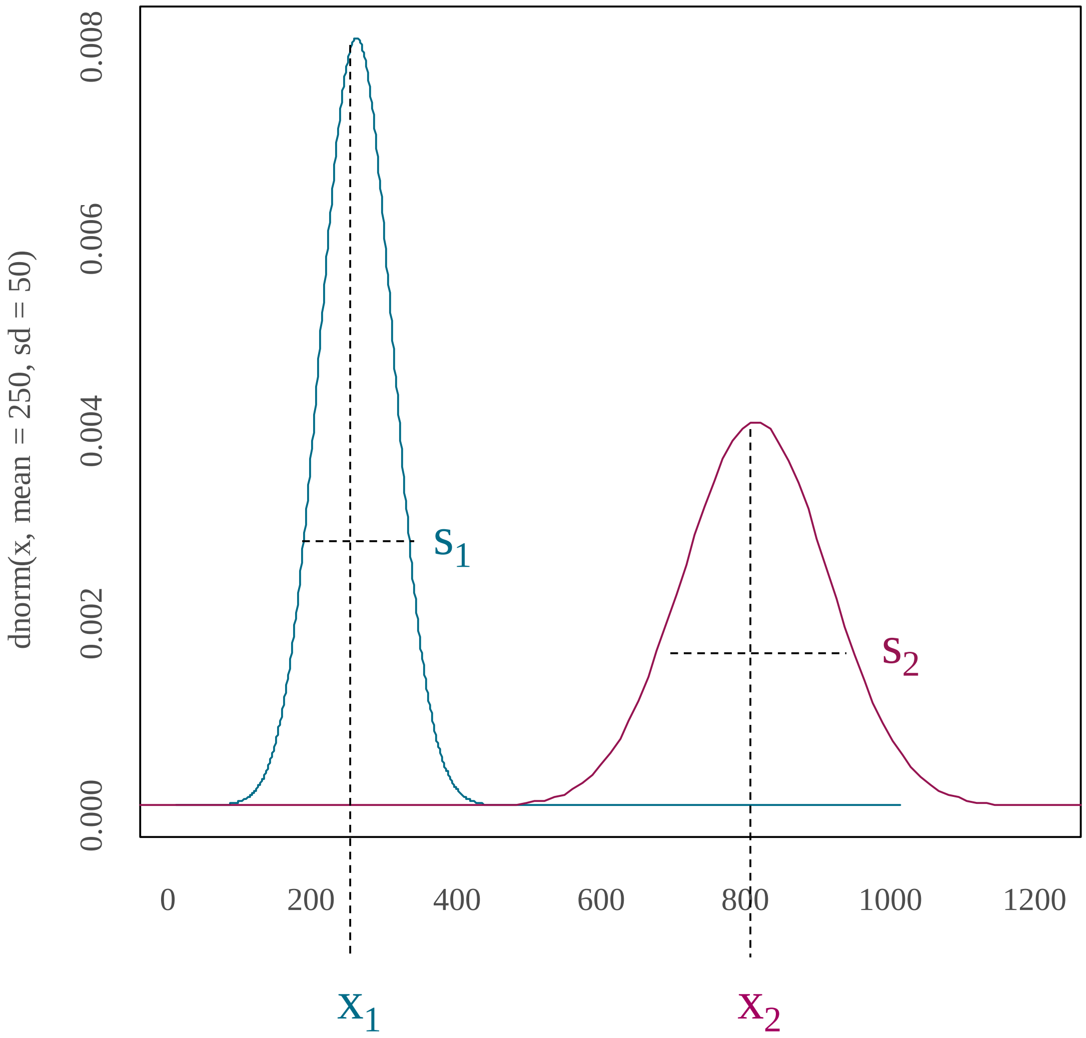
- **Comparing means of more than 2 groups**
  - 1-way ANOVA

- **Comparing means in a multifactor experiment**
  - 2-way ANOVA





Parametric statistics



$$t = \frac{x_1 - x_2}{\sqrt{\frac{s_1{}^2}{n_1} + \frac{s_2{}^2}{n_2}}}$$

$H_0 : \mu_A = \mu_B \text{ , } \mu_A - \mu_B = 0$

$H_1 : \mu_A \neq \mu_B$

Type 1 error, α

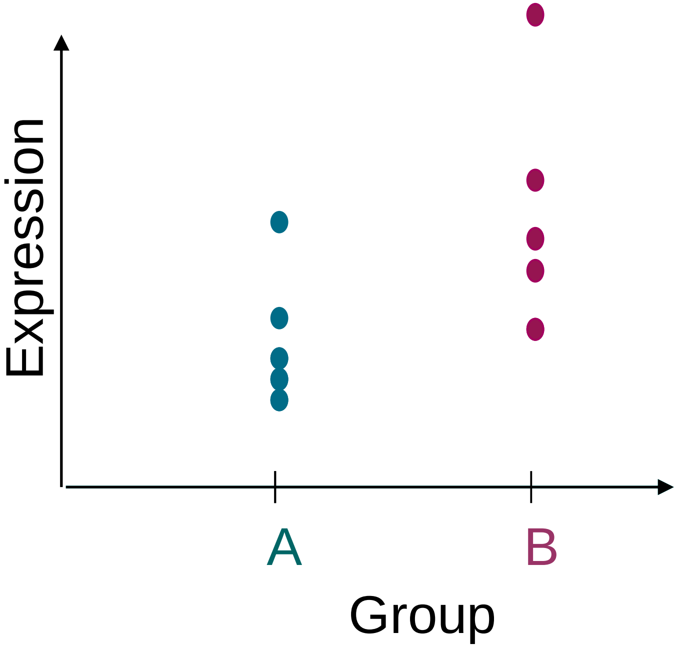
Type 2 error, β

Power = **1 - β**

Non-parametric statistical methods

- **Comparing <u>ranks</u> of 2 groups**
  - Mann-Whitney

- **Comparing <u>ranks</u> of more than 2 groups**
  - Kruskal-Wallis



Ranks

group A	group B
1	4
2	6
3	7
5	9
8	10

$$U_1 = n_1 * n_2 + \frac{n_1 * (n_1 + 1)}{2} - R_1$$

$$U_2 = n_1 * n_2 + \frac{n_2 * (n_2 + 1)}{2} - R_2$$

Non-parametric tests compared to parametric

Benefits

- Do not make any assumptions on data distribution
  - ⇒ robust to outliers
  - ⇒ allow for cross-experiment comparisons

Drawbacks

- Lower power than parametric counterpart
- Granular distribution of calculated statistic
  - ⇒ many genes get the same rank
  - ⇒ requires at least 6 samples / group

How to improve statistical power?

- **Need more accurate estimates of variability and effect size**
- **Improved analysis methods**
  - Variance shrinking: Empirical Bayes method
  - Partitioning variability: ANOVA, linear modeling
- **Improved experimental design**
  - Increase number of biological replicates
  - Use paired samples if possible
  - Randomization
  - Blocking

Pairing = matched samples from the same individual

Unpaired analysis

	Before	After
	2	3
	2	4
	3	2
	1	3
Mean	**2**	**3**
Stdev	**0.8**	**0.8**

Paired analysis

Before	After	Difference
2	3	1
2	3	1
3	4	1
1	2	1

Improving power with variance shrinking

- **Concept**
  - Borrow information from other genes which are expressed at similar level, and form a pooled error estimate

- **How?**
  - models the error - intensity dependence by comparing replicates
  - uses a smoothing function to estimate the error for any given intensity
  - calculates a weighted average between the observed gene specific variance and the model-derived variance (pooling)
  - incorporates the pooled variance estimate in the statistical test (usually t- or F-test)

- **Available in Chipster**
  - Two group test: Select empirical Bayes as the test
  - Linear modeling tool

Exercise 7: Statistical testing

- **Run different two group tests**
  - Select the file **normalized.tsv** and **Statistics / Two group test.** What is the default value of the parameter “test”? How many differentially expressed genes do you get?
  - Repeat the run but change **test** = **t-test**. Rename the result file to **t.tsv.** How many differentially expressed genes do you get now?
  - Repeat the run but change **test** = **Mann-Whitney**. Rename the result file to **MW.tsv.** How many differentially expressed genes do you get now?

- **Compare the results with a Venn diagram**
  - Do the gene lists overlap?

Exercise 8: Visualize and filter results

- **Filter genes based on fold change**
  - Select **two-sample.tsv** and the tool **Utilities / Filter using a column value.** Keep genes whose expression changes more than 4-fold:
    - Column = FC
    - Cut-off = 2 (remember that the fold change values are in log2 scale)
    - Smaller or larger = outside (we want both up and down-regulated genes)

- **View results in interactive visualizations**
  - Select the **column-value-filter.tsv** and visualization method **Volcano plot**
  - Visualize the file also as **Expression profile**

Exercise 9: Use paired samples in testing

- **Use pre-filled phenodata which contains more information about the samples**
  - Select **normalized.tsv** and **phenodata.tsv**, right click, and select **Links between selected / Unlink**.
  - Select **normalized.tsv** and right click to link it to **phenodata_moreSampleInfo**.
  - Inspect the new phenodata for sample information. Note that sample pairing information is in the patient column.

- **Repeat statistical testing so that you include pairing information**
  - Select the file **normalized.tsv** and **Statistics / Two group test** and set the parameter **Column with pairing information** = **patient**.
  - Does the number of differentially expressed genes change?
  - Rename the result file to **paired.tsv**

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Linear modeling

- **Models the expression of a gene as a linear combination of explanatory factors (e.g. group, gender, time, patient,…)**

$$\boxed{y = a + (b \cdot \text{group}) + (c \cdot \text{gender}) + (d \cdot \text{group} \cdot \text{gender})}$$

y = gene’s expression

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

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

b and c = main effects

d = interaction effect

Taking multiple factors into account

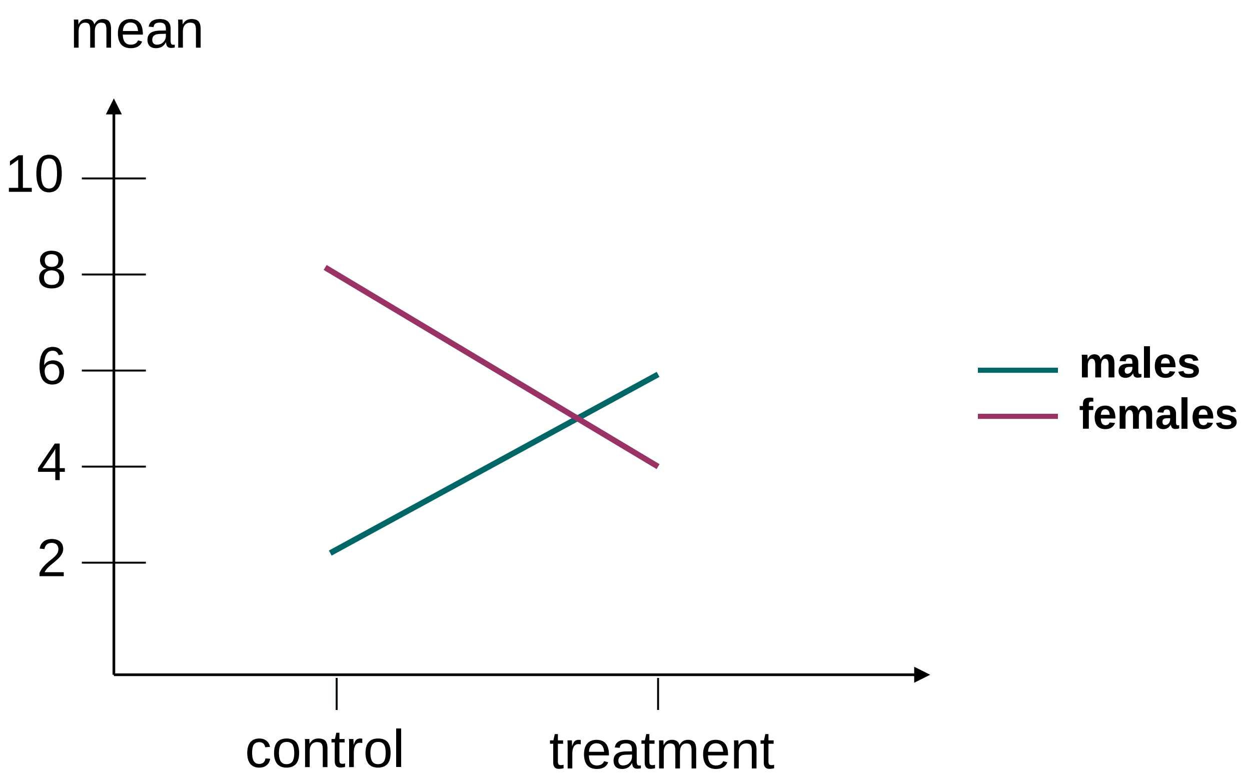
1 factor: treatment

	Control	Treatment
	2	5
	9	7
	1	3
	7	5
	8	4
	3	6
Mean	**5**	**5**

2 factors: treatment and <u>gender</u>

	Control	Treatment
	2	6
Males	3	7
	1	5
Mean	**2**	**6**
	8	4
Females	9	5
	7	3
Mean	**8**	**4**

Linear modeling: Interaction effect



Linear modeling tool in Chipster

- **Linear modeling tool in Chipster can take into account**
  - 3 main effects
  - Their interactions
  - Pairing
  - Technical replication (one sample is hybridized to several arrays)

- **Main effects can be treated as**
  - Linear = is there a trend towards higher numbers?
  - Factor = are there differences between the groups?

  If the main effect has only two levels (e.g. gender), selecting linear or factor gives the same result

- **Note that the result table contains all the genes, so in order to get the differentially expressed genes <u>you have to filter it</u>**
  - Use the tool **Utilities / Filter using a column value**
  - Select the column **p.adjusted** that corresponds to the comparison of your interest

Exercise 10: Linear modeling

- **Perform linear modeling so that the analysis takes into account group and gender.**
  - Select **normalized.tsv** and **Statistics / Linear modelling**
  - Set **Main effect 2 = gender** and **treat both main effects as factors**.
  - Open **limma.tsv** and inspect the result columns.

- **Retrieve differentially expressed genes for the group comparison**
  - Select **limma.tsv** and the tool **Utilities / Filter using a column value.** Keep genes whose adjusted p-value < 0.05:
    - Column = p.adjusted.main12
    - Cut-off = 0.05
    - Smaller or larger = smaller-than

- **Perform linear modeling so that the analysis takes into account group, gender and pairing.**
  - As above but include **pairing = patient**.
  - Open **limma.tsv** and inspect the result columns.
  - Retrieve differentially expressed genes as before.

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Multiple testing correction

- **Problem: When thousands of genes are tested for differential expression, a gene can get a good p-value just by chance.**

1 gene, $\alpha = 0.05$

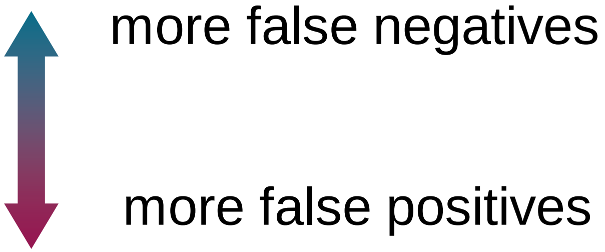
$\Rightarrow$ false positive incidence = 1 / 20

30 000 genes, $\alpha = 0.05$

$\Rightarrow$ false positive incidence = 1500

- **Solution: Correct the p-values for multiple testing. Methods:**

  - Bonferroni
  - Holm (step down)
  - Westfall & Young
  - Benjamini & Hochberg

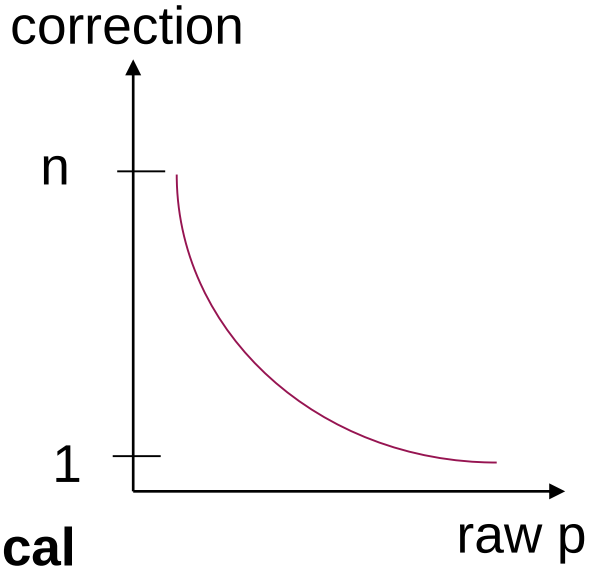


more false negatives

more false positives

Benjamini & Hochberg method (BH)

- **How does it work?**
  - rank p-values from largest to smallest
  - largest p-value remains unaltered
  - second largest p-value = p * n / (n-1)
  - third largest p-value = p * n / (n-2)
  - …
  - smallest p-value = p * n / (n-n+1) = p * n



- **Some adjusted p-values can become identical**
  - Adjusting should not change the order of p-values, so if $pa_{i+1} > pa_i$ then $pa_{i+1} = pa_i$

- **We can reduce the severity of multiple testing correction by reducing the number of genes tested (n)**
  - use independent filtering

- **The adjusted p-value is FDR (false discovery rate)**
  - Tells what proportion of <u>results</u> can be false positives

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Annotation

- **Gene annotation = information about biological function, pathway involvement, chromosal location etc**

- **Annotation information is collected from different biological databases to a single database by the Bioconductor project**
  - Bioconductor provides annotation packages for many microarrays

- **Annotation package is required by many analysis tools**
  - Annotation, GO/KEGG enrichment, promoter analysis, chromosomal plots
  - These tools don’t work for those chiptypes which don’t have Bioconductor annotation packages

Annotations for the selected gene list

Probe	Symbol	Description	Chromosome	Chromosome Location	GenBank	Gene	Cytoband	UniGene	PubMed	Gene Ontology	Pathway
205626_s_at	CALB1	calbindin 1, 28kDa	8	-91140013	NM_004929	793	8q21.3-q22.1	Hs.65425	22	locomotory behavior cytoplasm vitamin D binding calcium ion binding protein binding	
220281_at	SLC12A1	solute carrier family 12 (sodium/potassium/chloride transporters), member 1	15	46285789	AI632015	6557	15q15-q21.1	Hs.123116	13	ion transport potassium ion transport sodium ion transport chloride transport membrane fraction plasma membrane membrane integral to membrane transporter activity sodium:potassium:chloride symporter activity symporter activity potassium ion binding sodium ion binding	
206054_at	KNG1	kininogen 1	3	187917813	NM_000893	3827	3q27	Hs.77741	86	smooth muscle contraction inflammatory response negative regulation of cell adhesion elevation of cytosolic calcium ion concentration blood coagulation diuresis natriuresis negative regulation of blood coagulation vasodilation positive regulation of apoptosis extracellular region cysteine protease inhibitor activity receptor binding heparin binding zinc ion binding	Complement and coagulation cascades
										behavior gamma-aminobutyric acid catabolic process neurotransmitter catabolic	Glutamate

Alternative CDF environments for Affymetrix

- **CDF is a file that links individual probes to gene transcripts**
- **Affymetrix default annotation uses older CDF files which may map many probes to wrong genes**
- **Alternative CDFs fix this problem**
- **In Chipster selecting ”custom chiptype” in Affymetrix normalization takes altCDFs to use**
- **For more information see**
  - Dai et al, (2005) Nuc Acids Res, 33(20):e175: *Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data*
  - http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/genomic_curated_CDF.asp

Exercise 11: Annotation

- **Annotate genes**
  - Select the file **two-sample.tsv**
  - Run **Annotation / Agilent, Affymetrix or Illumina gene list** so that you include the FC and p-value information to the result file
  - Run **Annotation / Add annotations to data**

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Pathway analysis – why?

- **Statistical tests can yield thousands of differentially expressed genes**

- **It is difficult to make ”biological” sense out of the result list**

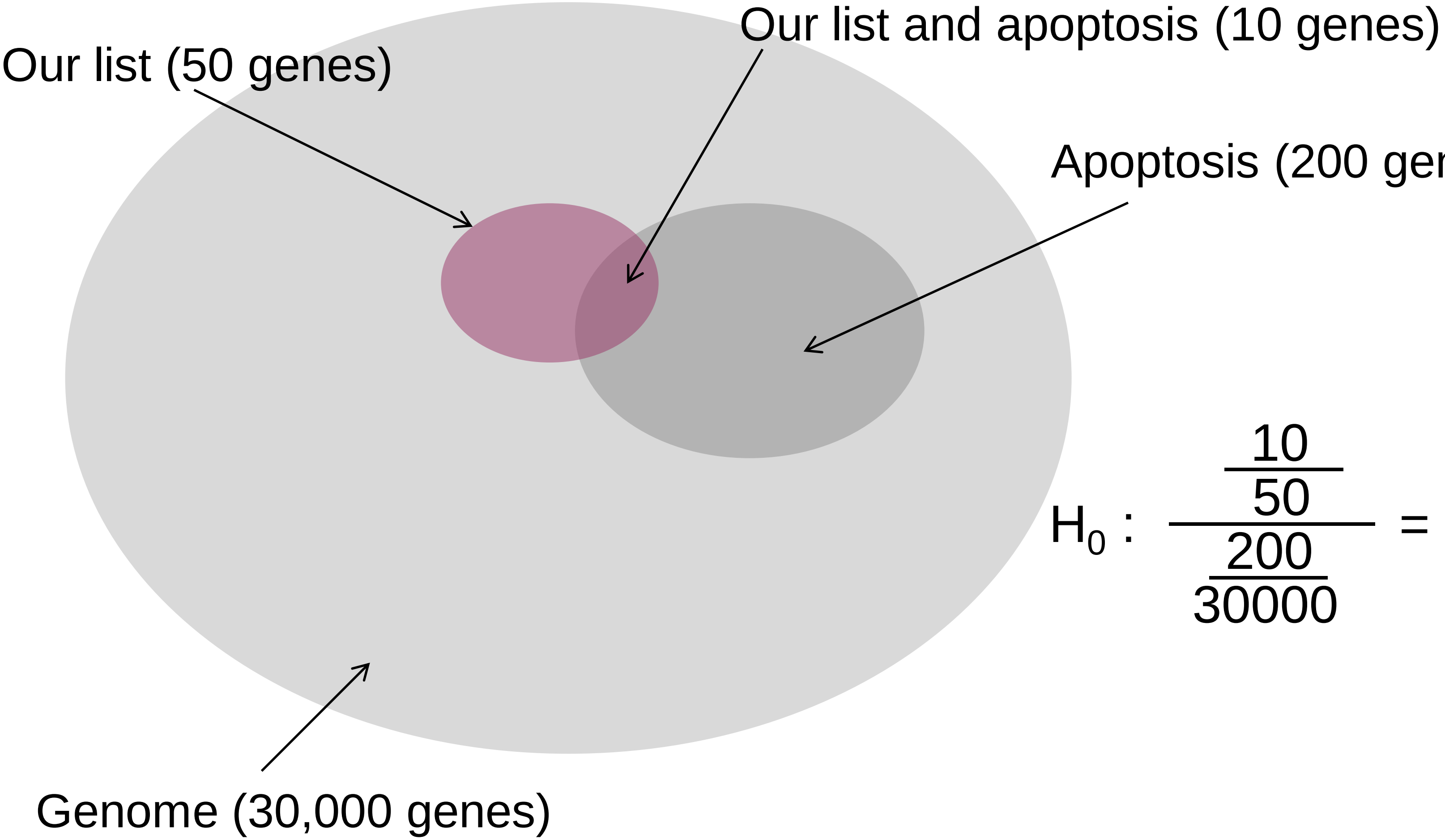
- **Looking at the bigger picture can be helpful, e.g. which pathways are differentially expressed between the experimental groups**

- **Databases such as KEGG, GO, Reactome and ConsensusPathDB provide grouping of genes to pathways, biological processes, molecular functions, etc**

- **Two approaches to pathway analysis**
  - Gene set enrichment analysis
  - Gene set test

Approach I: Gene set enrichment analysis

1. **Perform a statistical test to find differentially expressed genes**
2. **Check if the list of differentially expressed genes is ”enriched” for some pathways**



$$H_0 : \frac{\frac{10}{50}}{\frac{200}{30000}} = 30 >> 1$$

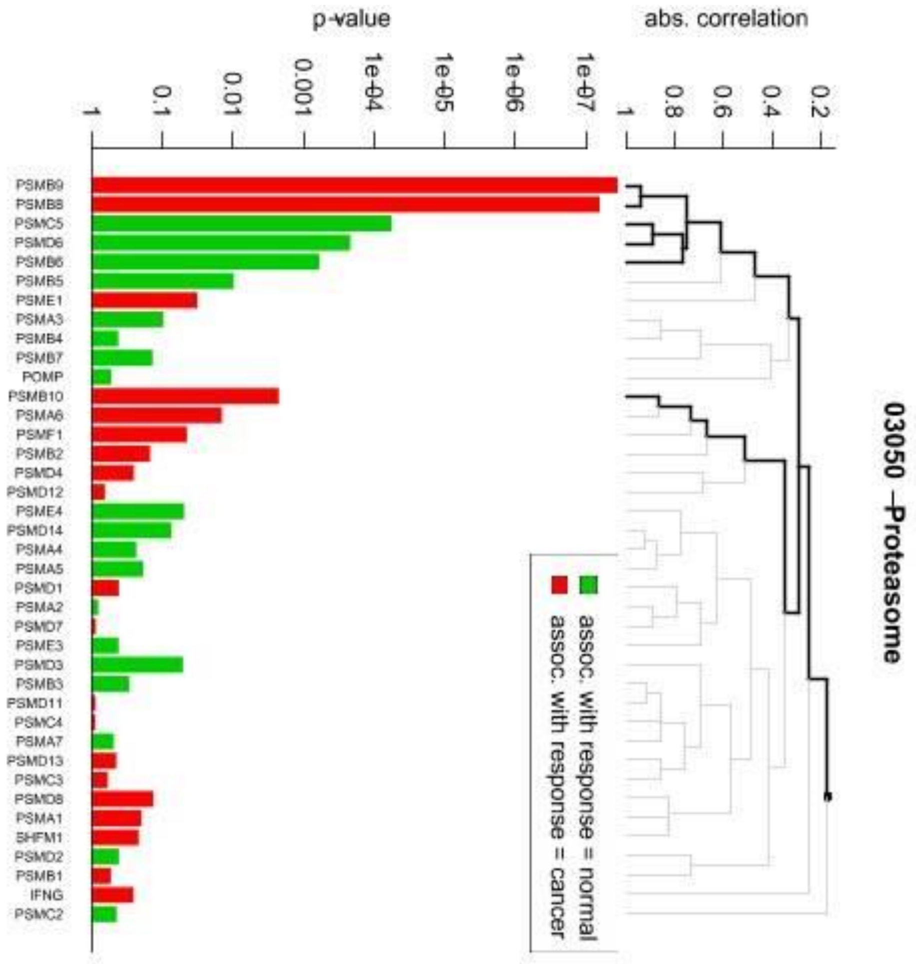
Approach II: Gene set test

1. **Do NOT perform differential gene expression analysis**
2. **Group genes to pathways and perform differential expression analysis for the whole pathway**

➢ **Advantages**

- More sensitive than single gene tests
- Reduced number of tests → less multiple testing correction

  → increased power



ConsensusPathDB

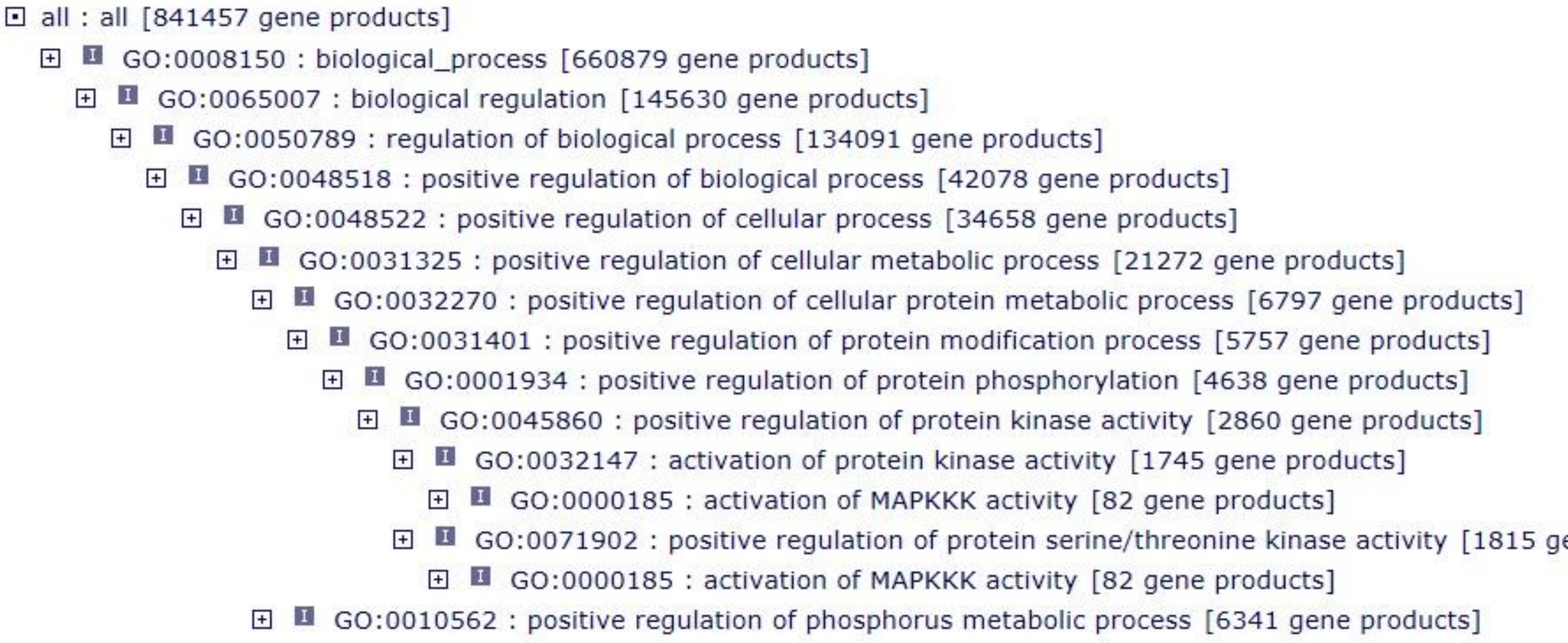
- **One-stop shop: Integrates pathway information from 32 databases covering**
  - biochemical pathways
  - protein-protein, genetic, metabolic, signaling, gene regulatory and drug-target interactions

- **Developed by Ralf Herwig’s group at the Max-Planck Institute in Berlin**

- **ConsensusPathDB over-representation analysis tool is integrated in Chipster**
  - runs on the MPI server in Berlin

GO (Gene Ontology)

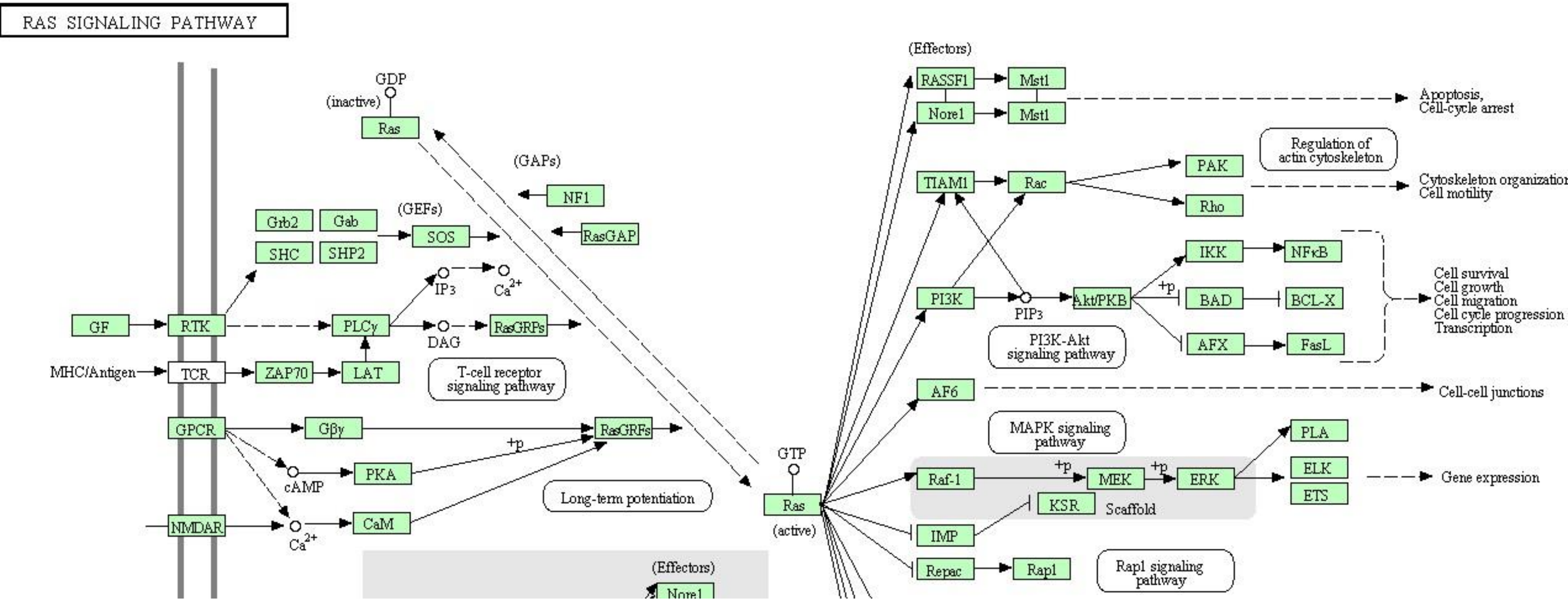
- **Controlled vocabulary of terms for describing gene product characteristics**
- **3 ontologies**
  - Biological process
  - Molecular function
  - Cellular component
- **Hierarchical structure**



- ⊡ all : all [841457 gene products]
  - ⊞ GO:0008150 : biological_process [660879 gene products]
    - ⊞ GO:0065007 : biological regulation [145630 gene products]
      - ⊞ GO:0050789 : regulation of biological process [134091 gene products]
        - ⊞ GO:0048518 : positive regulation of biological process [42078 gene products]
          - ⊞ GO:0048522 : positive regulation of cellular process [34658 gene products]
            - ⊞ GO:0031325 : positive regulation of cellular metabolic process [21272 gene products]
              - ⊞ GO:0032270 : positive regulation of cellular protein metabolic process [6797 gene products]
                - ⊞ GO:0031401 : positive regulation of protein modification process [5757 gene products]
                  - ⊞ GO:0001934 : positive regulation of protein phosphorylation [4638 gene products]
                    - ⊞ GO:0045860 : positive regulation of protein kinase activity [2860 gene products]
                      - ⊞ GO:0032147 : activation of protein kinase activity [1745 gene products]
                        - ⊞ GO:0000185 : activation of MAPKKK activity [82 gene products]
                      - ⊞ GO:0071902 : positive regulation of protein serine/threonine kinase activity [1815 ge
                        - ⊞ GO:0000185 : activation of MAPKKK activity [82 gene products]
              - ⊞ GO:0010562 : positive regulation of phosphorus metabolic process [6341 gene products]

KEGG

- **Kyoto Encyclopedia for Genes and Genomes**
- **Collection of pathway maps representing molecular interaction and reaction networks for**
  - metabolism
  - cellular processes
  - diseases, etc



Exercise 12: Gene set enrichment analysis

- **Identify over-represented GO terms**
  - Select the **two-sample.tsv** file and run **Pathways / Hypergeometric test for GO.** Open **hypergeo.html** and read about the first term. Check in **hypergeo.tsv** how many terms do you get.

- **Extract genes for a specific GO term**
  - Copy the GO identifier for the top term (GO:0006082).
  - Select **two-sample.tsv** and run tool **Utilities / Extract genes for GO term**, pasting the GO identifier in the parameter field.
  - Open **extracted-from-GO.tsv.** How many genes do you get? Are they up- or down-regulated (use also Volcano plot and Expression profile)?

- **Identify over-represented ConsensusPathDB pathways**
  - Select **two-sample.tsv** and run **Pathways / Hypergeometric test for ConsensusPathDB**.
  - Click on the links in the **cpdb.html** file to read about the pathways.

Exercise 13: Gene set test

- **Identify differentially expressed KEGG pathways**
  - Select the **normalized.tsv** file and **Pathways / Gene set test**. Set the **Number of pathways to visualize = 4**
  - Explore **global-test-result-table.tsv.** How many differentially expressed KEGG pathways do you get?
  - Explore **multtest.pdf.** Which gene contributes most to the first pathway?

Microarray data analysis workflow

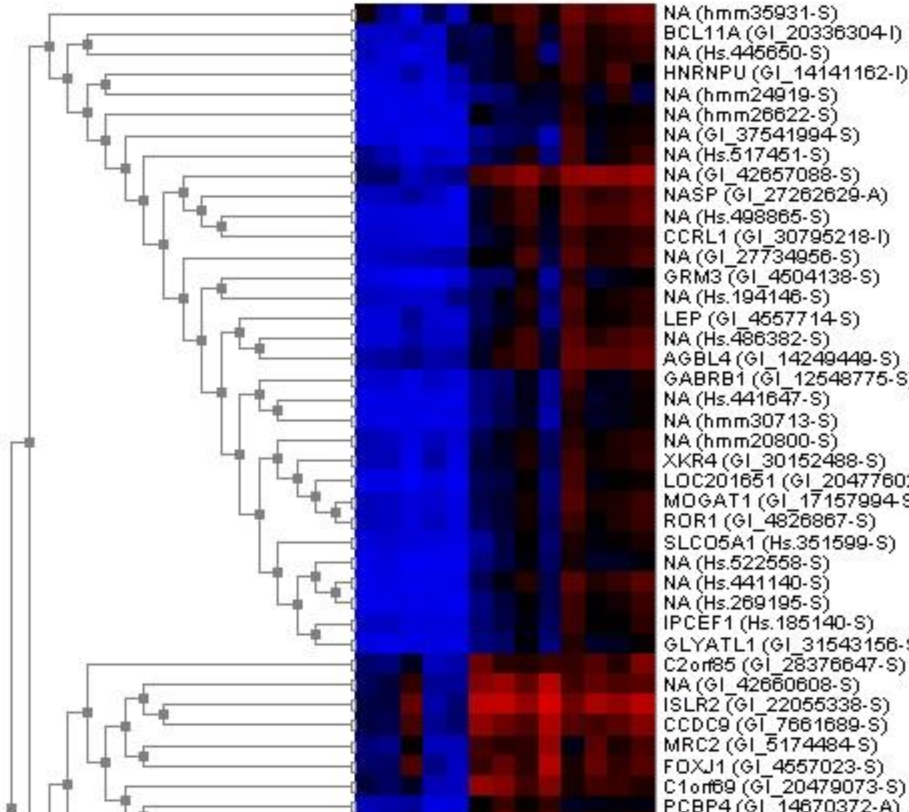
- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

Clustering in Chipster

- **Hierarchical**
  - Includes reliability checking of the resulting tree with bootstrapping
- **K-means**
  - Additional tool to estimate K
- **Quality threshold**
- **Self-organizing maps**
- **K-nearest neighbor (KNN)**
  - Classification aka class prediction

Hierarchical clustering

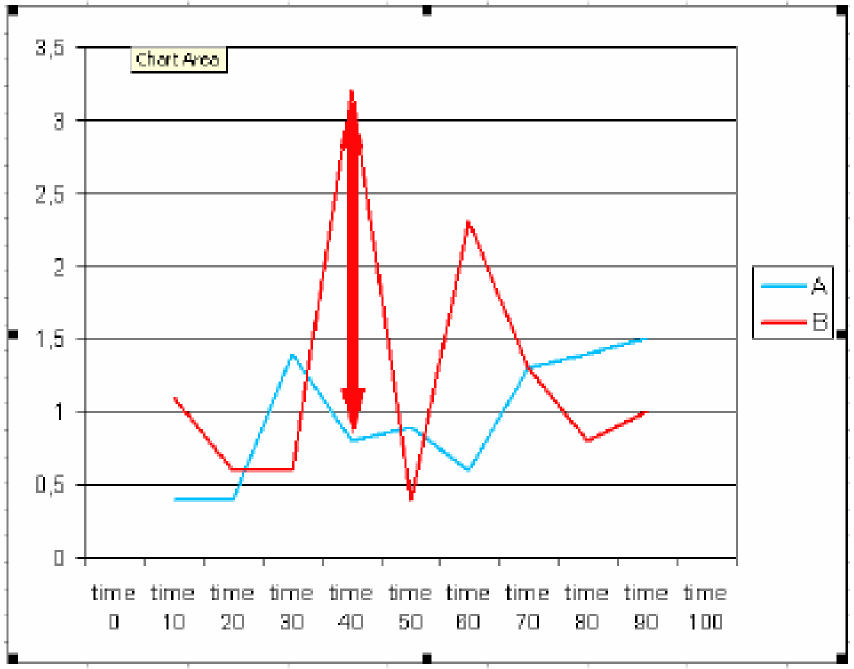
- **Provides stable clusters**
- **Assumes pairwise correlations**
- **Early mistakes cannot be corrected**
- **Computationally intensive**
- **Drawing methods**
  - Single / average / complete linkage
- **Distance methods**
  - Euclidean distance
  - Pearson / Spearman correlation



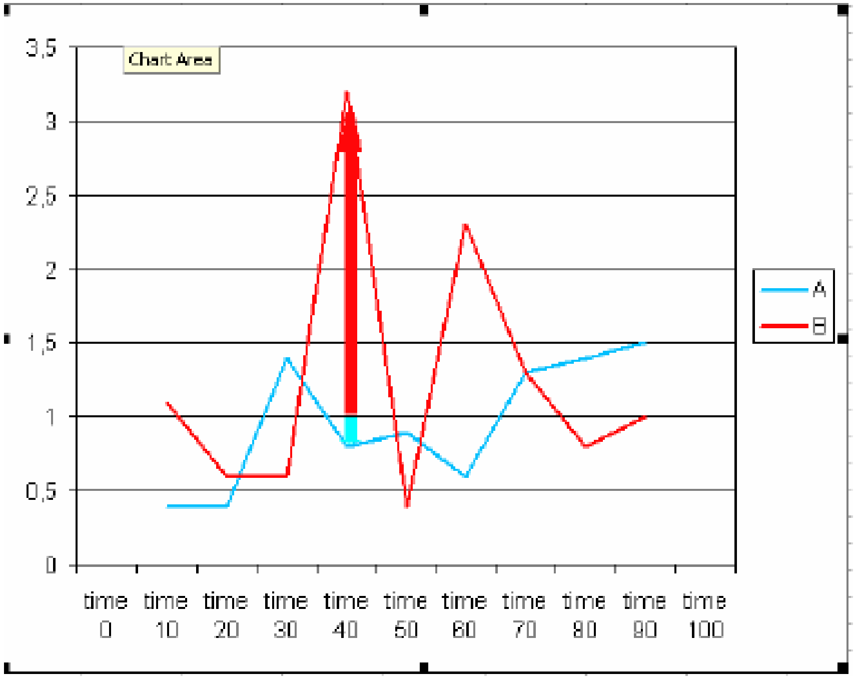
Hierarchical clustering: distance methods

One can either calculate the distance between two pairs of data sets (e.g. samples) or the similarity between them

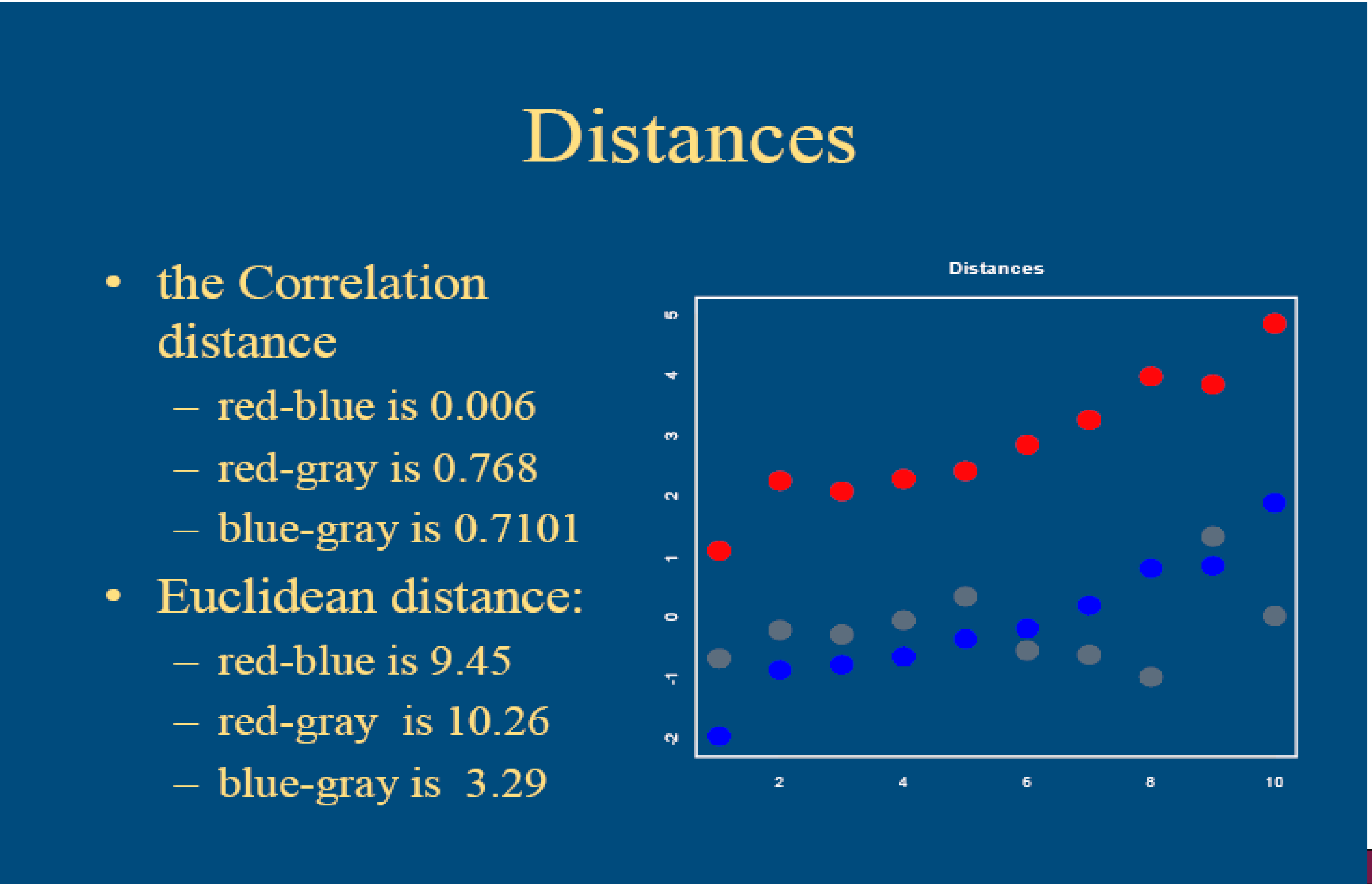
Euclidean distance



Pearson correlation



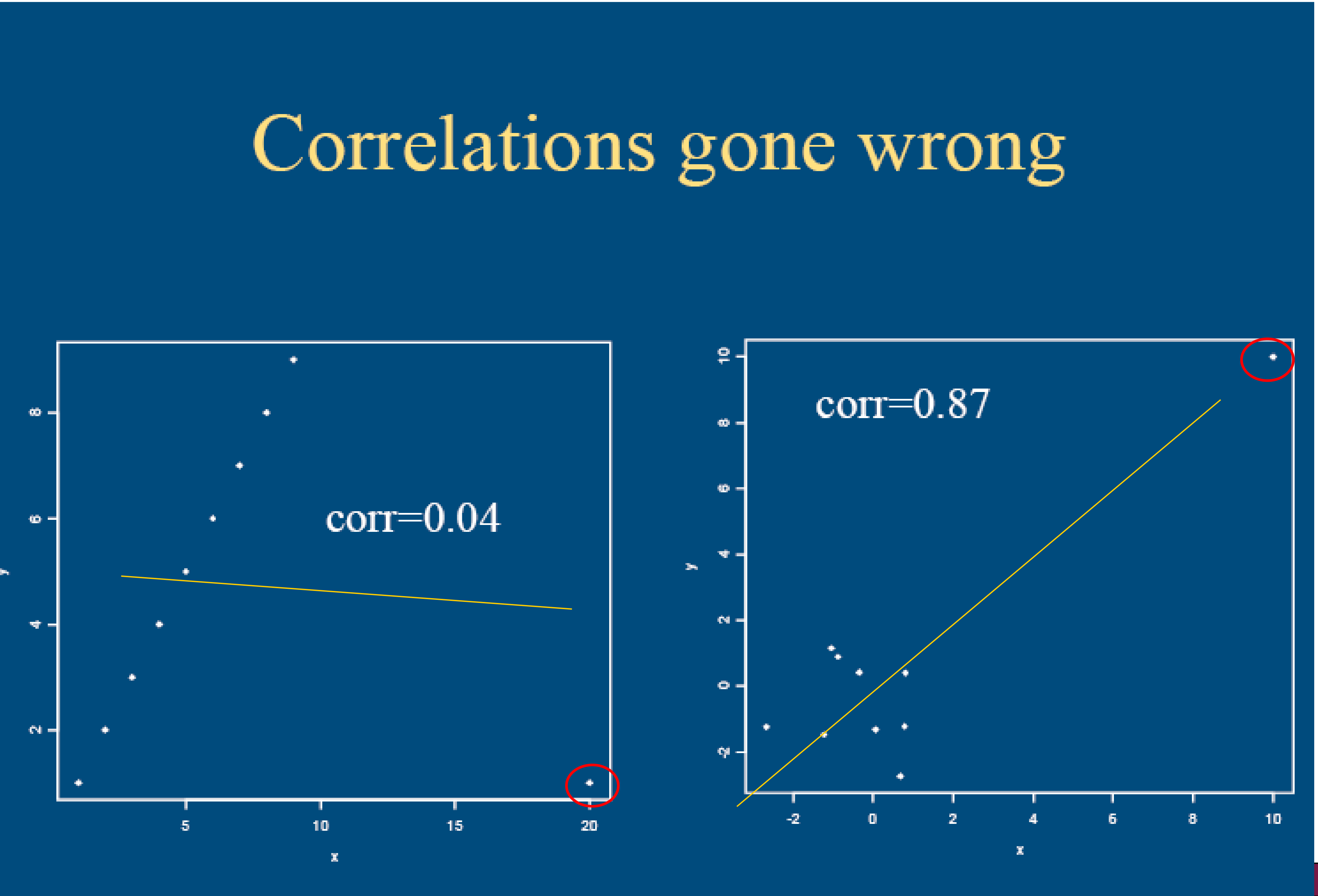
Distance methods can yield very different results



Distances

- the Correlation distance
  - red-blue is 0.006
  - red-gray is 0.768
  - blue-gray is 0.7101
- Euclidean distance:
  - red-blue is 9.45
  - red-gray is 10.26
  - blue-gray is 3.29

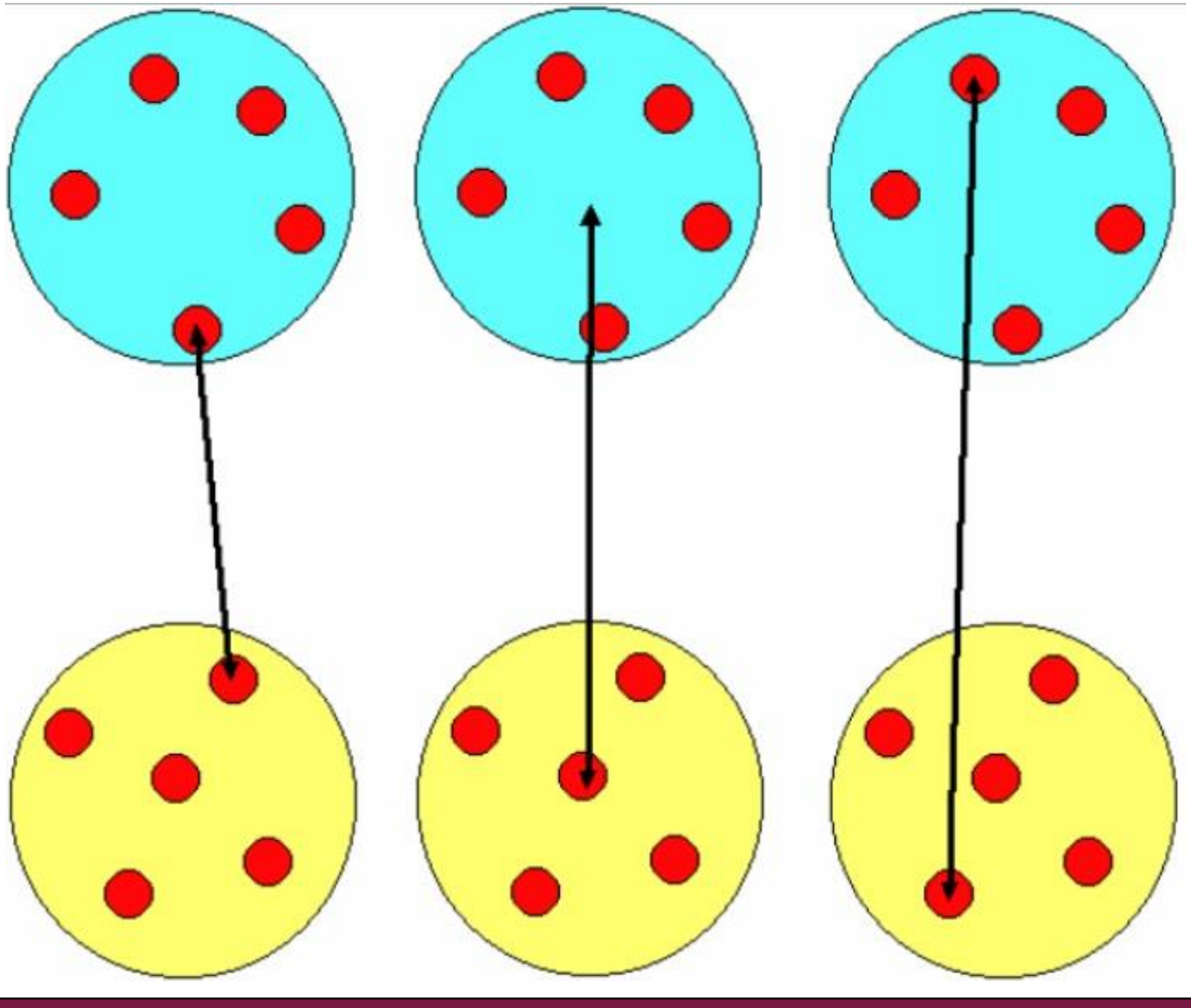
Correlations are sensitive to outliers (use Spearman)!



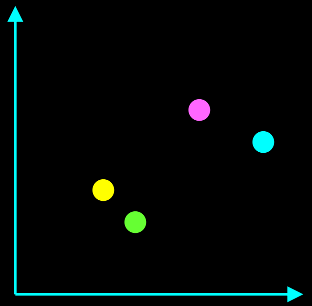
Correlations gone wrong

Hierarchical clustering: drawing methods

single linkage   average linkage   complete linkage



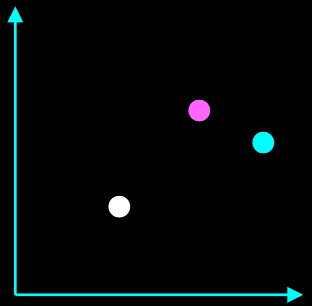
Hierarchical clustering (euclidean distance)



calculate distance matrix

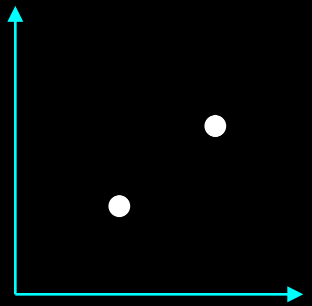
	gene 1	gene 2	gene 3	gene 4
gene 1	0			
gene 2	2	0		
gene 3	8	7	0	
gene 4	10	12	4	0

calculate averages of most similar



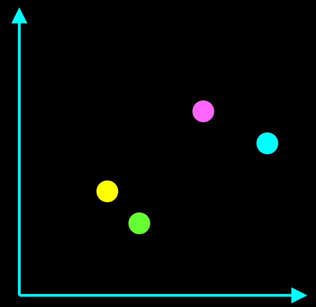
	gene 1,2	gene 3	gene 4
gene 1,2	0		
gene 3	7.5	0	
gene 4	11	4	0

calculate averages of most similar



	gene 1,2	gene 3,4
gene 1,2	0	
gene 3,4	9.25	0

Hierarchical clustering (avg. linkage)



calculate distance matrix

	gene 1	gene 2	gene 3	gene 4
gene 1	0			
gene 2	2	0		
gene 3	8	7	0	
gene 4	10	12	4	0

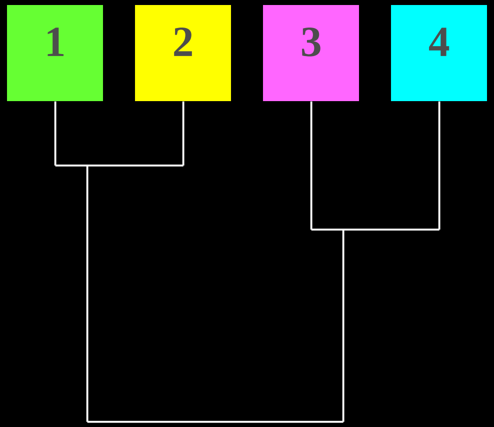
calculate averages of most similar

	gene 1,2	gene 3	gene 4
gene 1,2	0		
gene 3	7.5	0	
gene 4	11	4	0

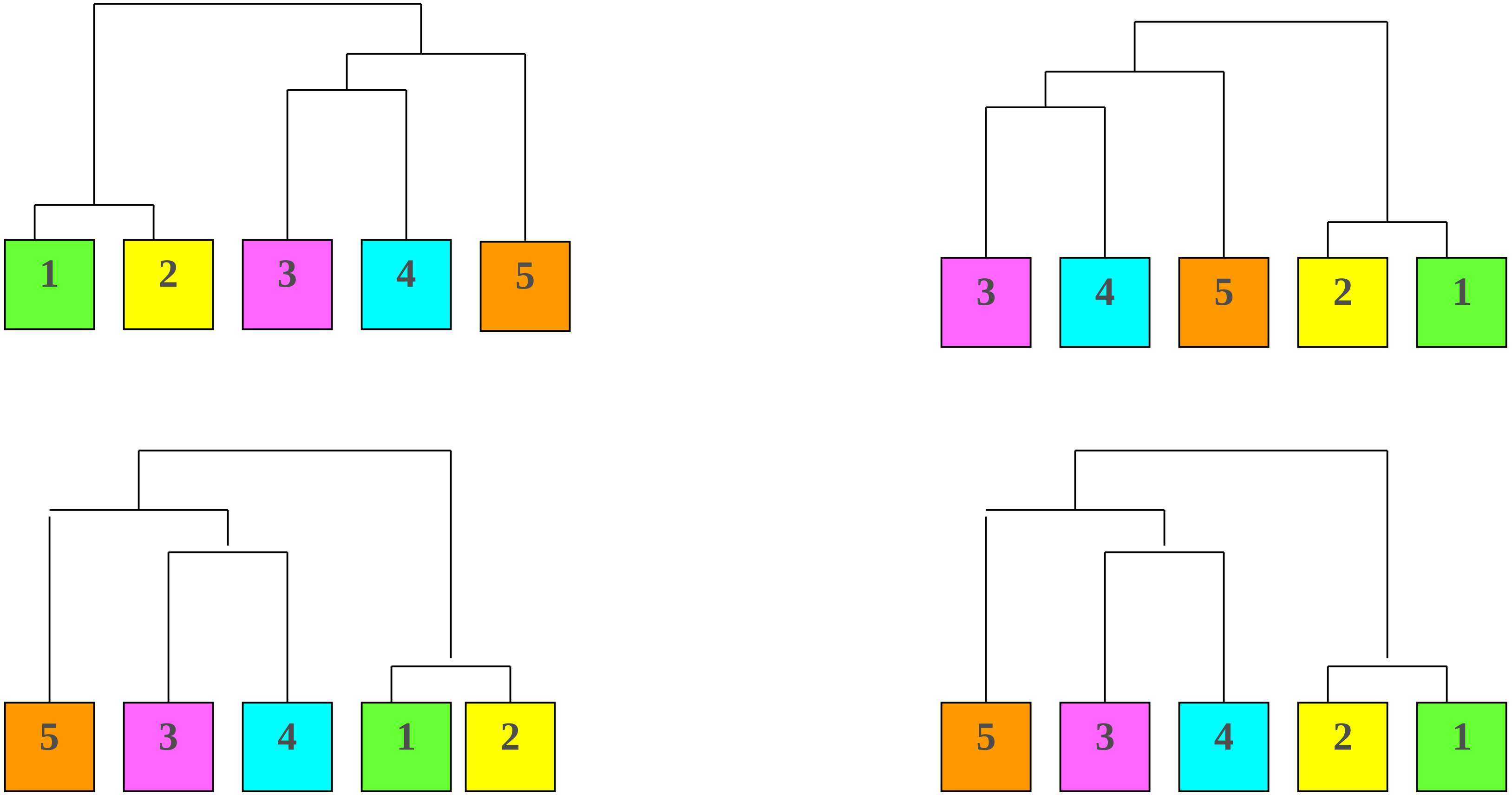
calculate averages of most similar

	gene 1,2	gene 3,4
gene 1,2	0	
gene 3,4	9.25	0

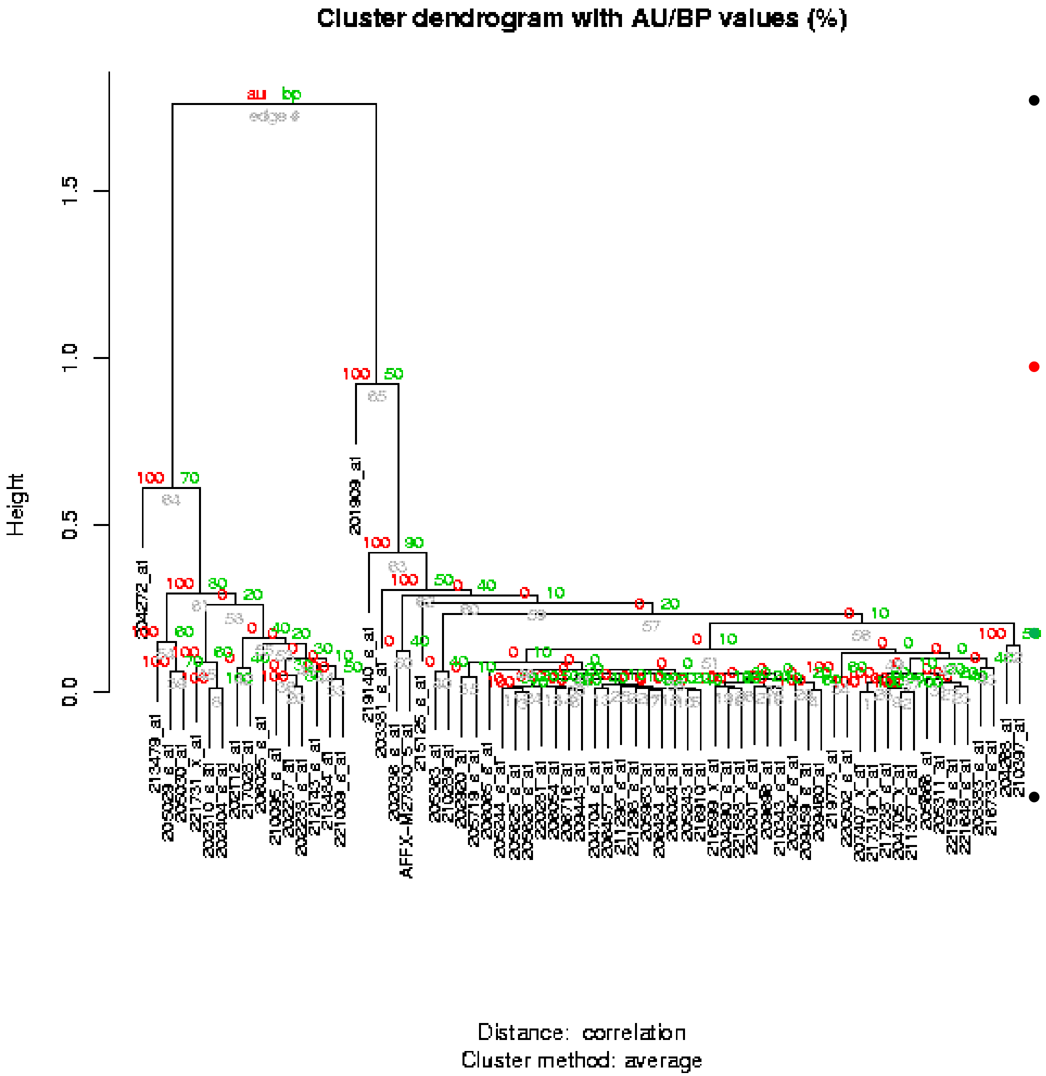
Dendrogram



When assessing similarity, look at the branching pattern instead of sample order

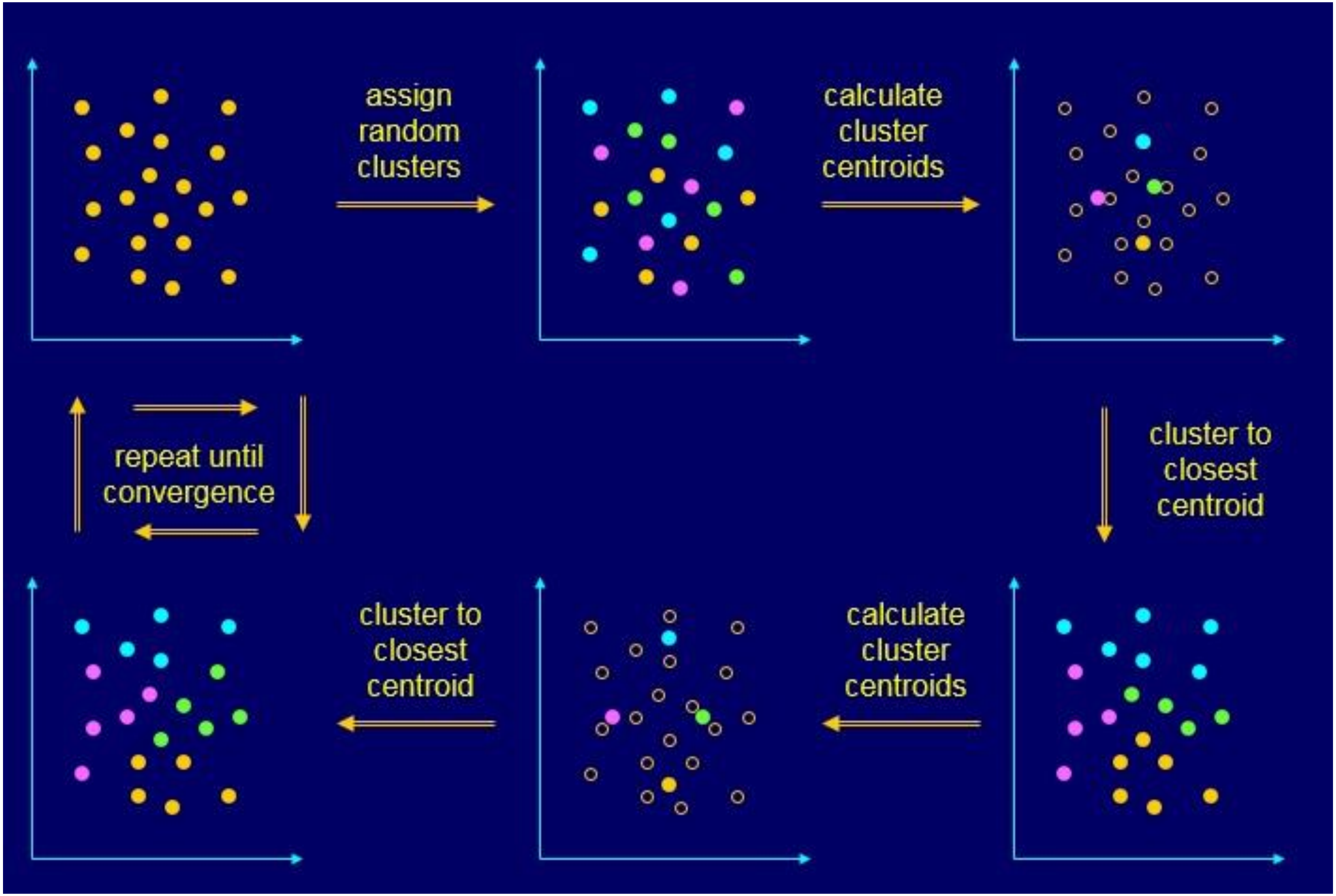


Assessing the certainty of the branching pattern by bootstrapping



- You can get this plot by setting the parameter Resample = bootstrap. Increasing the number of resamplings increases the accuracy but makes the analysis slower. You can start with 100.
- AU = approximately unbiased p-value, computed by multiscale bootstrap resampling. Clusters with AU larger than 95% are strongly supported by data.
- BP = bootstrap probability p-value, computed by normal bootstrap resampling
- Developers recommend to use the AU values

K-means clustering


assign random clusters
calculate cluster centroids
cluster to closest centroid
calculate cluster centroids
cluster to closest centroid
repeat until convergence

Exercise 14: Hierarchical clustering

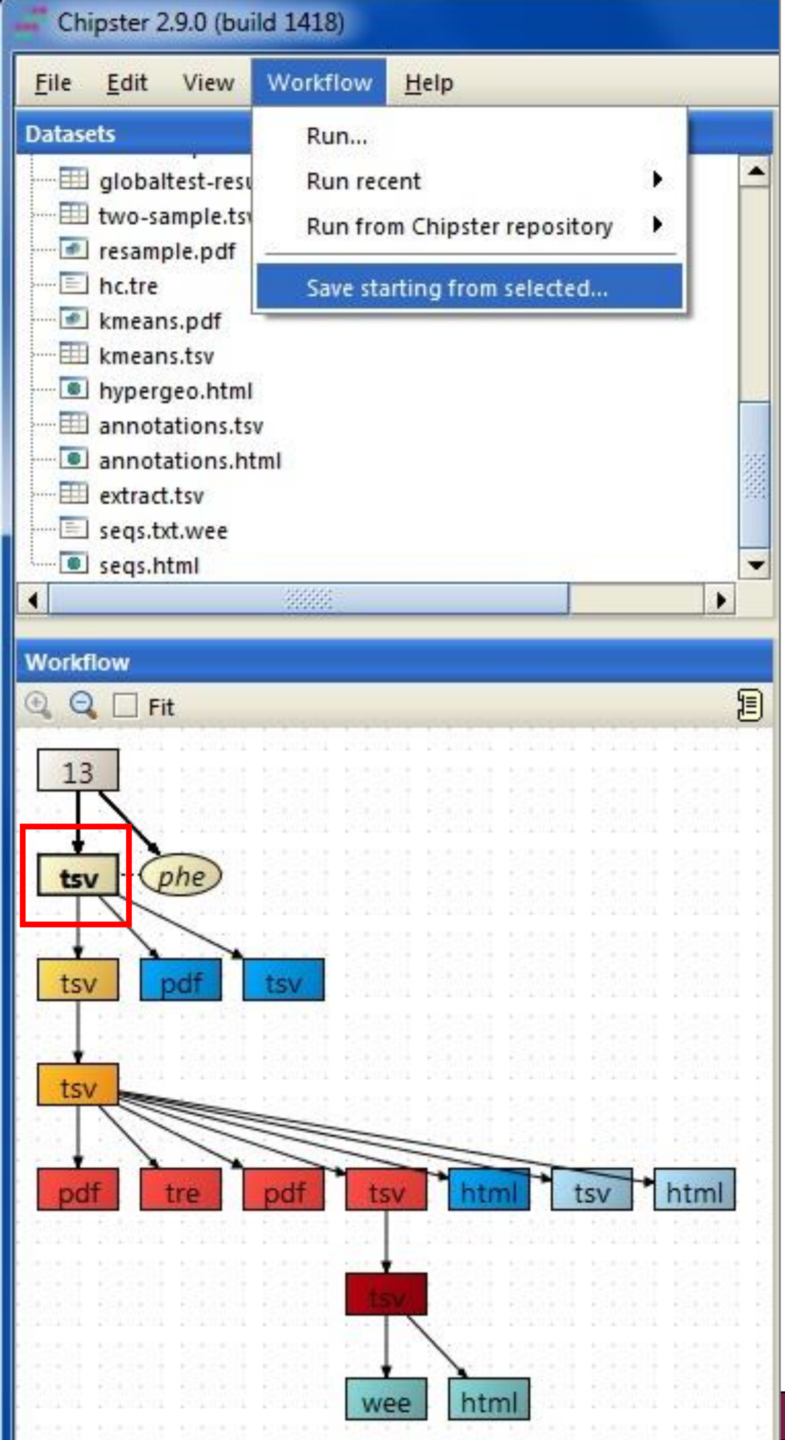
- **Cluster genes**
  - Select the **column-value-filter.tsv** and run **Clustering / Hierarchical**.
  - View the resulting file **hc.tre** as **Hierarchical clustering**.

- **Cluster genes and samples**
  - Select the **column-value-filter.tsv** and run the tool **Visualization / Heatmap.**
  - Select the **column-value-filter.tsv** and run the tool **Visualization / Annotated heatmap,** using parameters
    - Coloring scheme = Blue - white – red
    - Cluster samples only = no

Microarray data analysis workflow

- **Importing data to Chipster**
- **Normalization**
- **Describing samples with a phenodata file**
- **Quality control**
  - Array level
  - Experiment level
- **Filtering (optional)**
- **Statistical testing**
  - Parametric and non-parametric tests
  - Linear modeling
  - Multiple testing correction
- **Annotation**
- **Pathway analysis**
- **Clustering**
- **Saving the workflow**

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 on another dataset, select**
  - Workflow → Open and run
  - Workflow → Run recent (if you saved the workflow recently).

Exercise 15: Saving a workflow

- **Prune your workflow if necessary by removing**
  - cyclic structures
  - files produced by visual selection (gray boxes)

- **Save the workflow**
  - Select **normalized.tsv** and click on **Workflow / Save starting from selected**. Give your workflow a meaningful name and save it.

Microarray data analysis summary

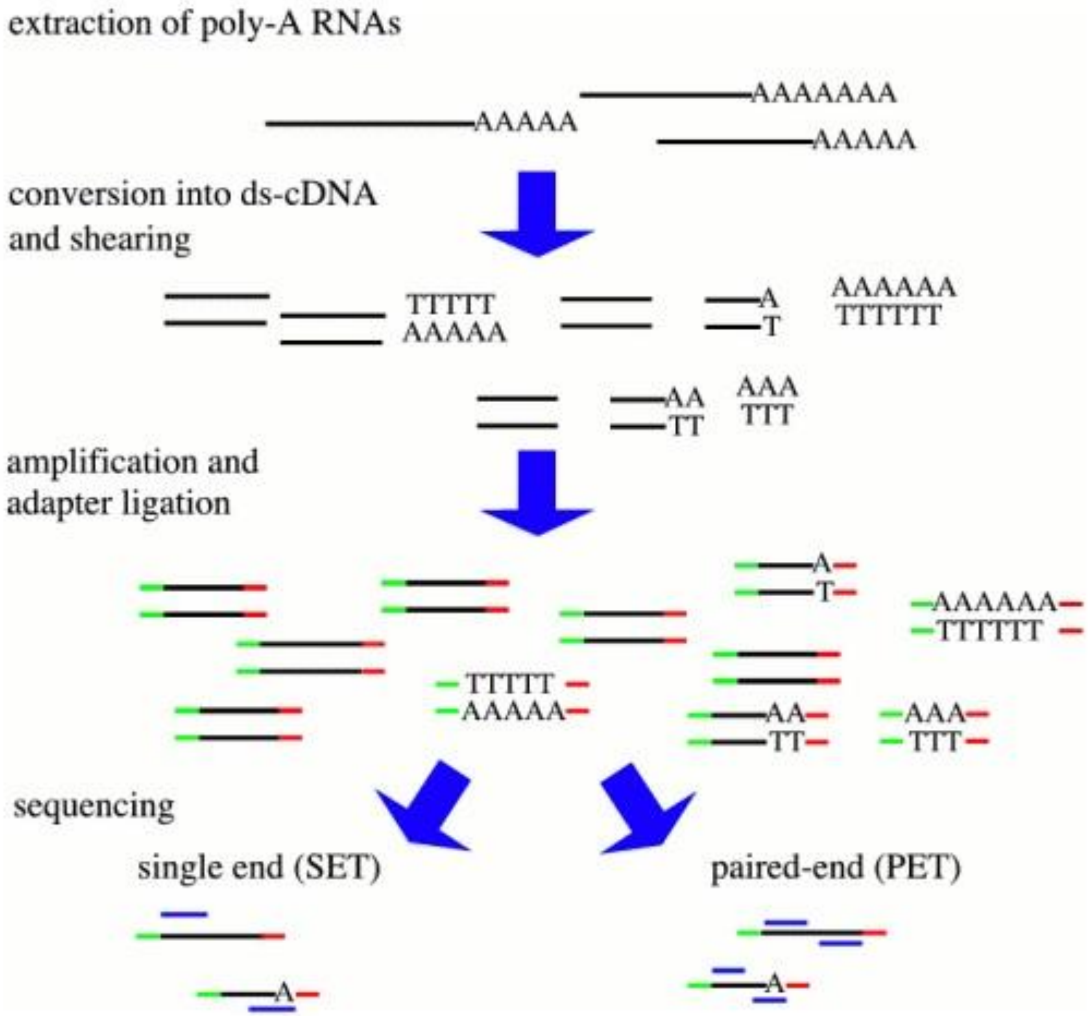
- **Normalization**
  - RMA for Affy
- **Quality control at array level: are there outlier arrays?**
  - RLE, NUSE
- **Quality control at experiment level: do the sample groups separate? Are there batch effects or outliers?**
  - PCA, NMDS, dendrogram
- **(Independent filtering of genes)**
  - e.g. 50% based on coefficient of variation
  - Depends on the statistical test to be used later
- **Statistical testing**
  - Empirical Bayes method (two group test / linear modeling)
- **Annotation, pathway analysis, promoter analysis, clustering, classification…**

Introduction to RNA-seq

What can I investigate with RNA-seq?

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

How was your data produced?



PolyA purification

cDNA generation & fragmentation

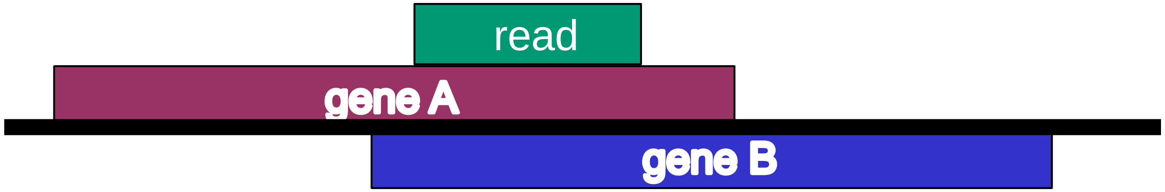
Library construction

Size selection

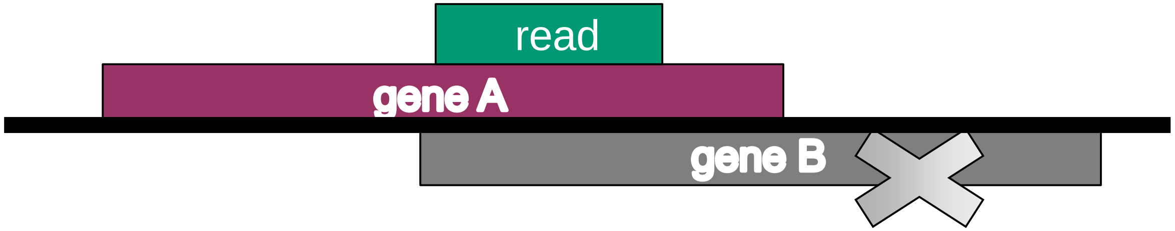
http://cmb.molgen.mpg.de/2ndGenerationSequencing/Solas/RNA-seq.html

Stranded RNA-seq data

- **Tells if a read maps to 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…



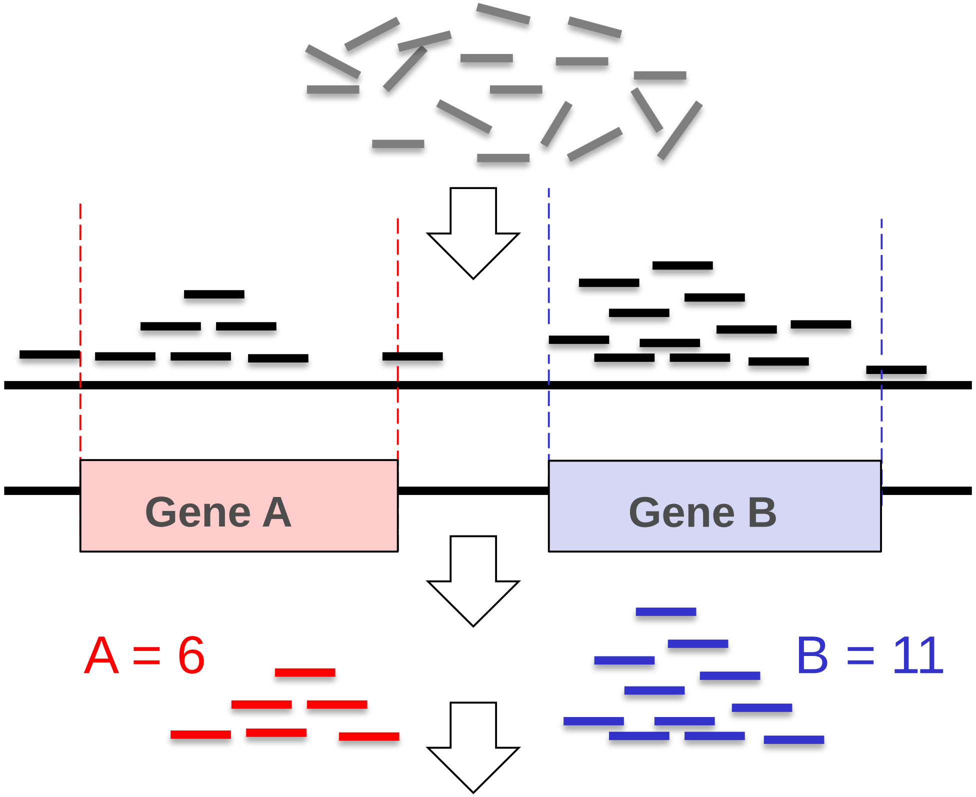
Unstranded data: Does the read come from geneA or geneB?



Stranded data
→ the read comes from geneA

RNA-seq data analysis

RNA-seq data analysis: typical steps



Raw data (reads)

Align reads to reference genome

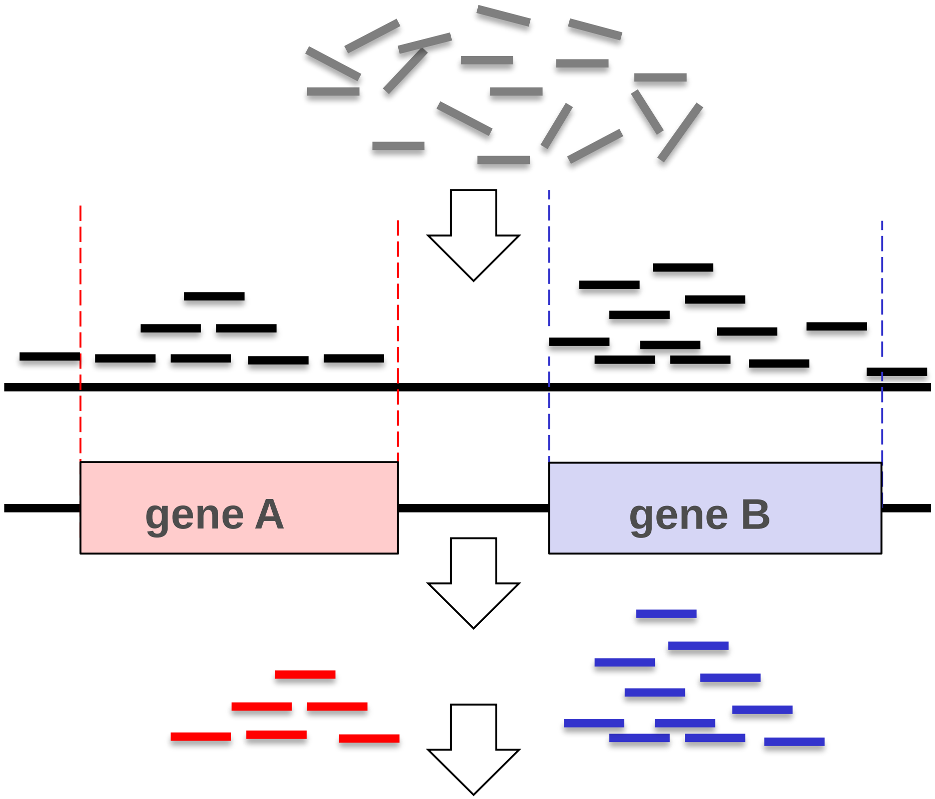
Match alignment positions with known gene positions

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

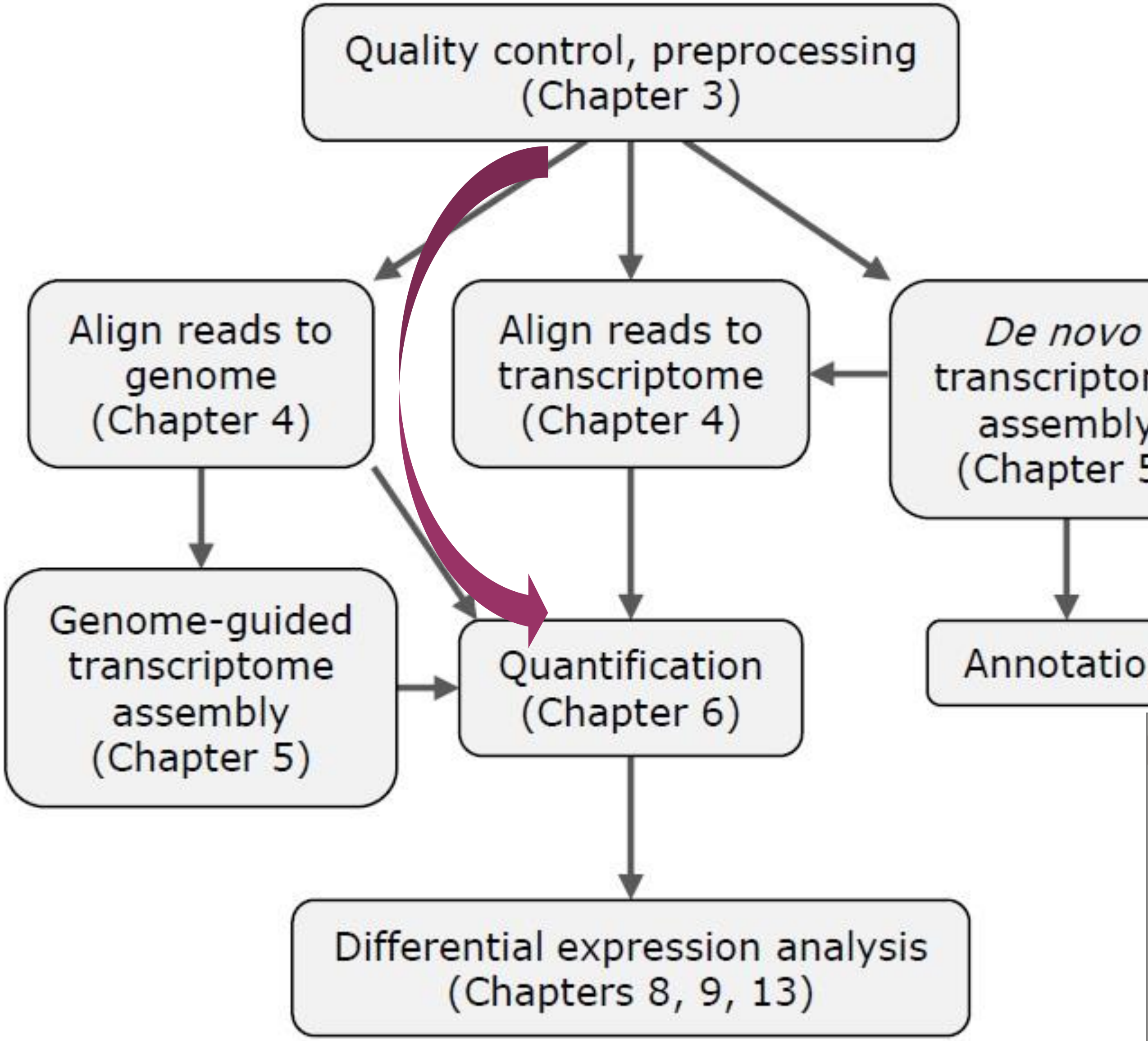
RNA-seq data analysis: steps, tools and files



	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	Trimmomatic	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)

RNA-seq data analysis workflow



Quality control, preprocessing (Chapter 3)

Align reads to genome (Chapter 4)

Align reads to transcriptome (Chapter 4)

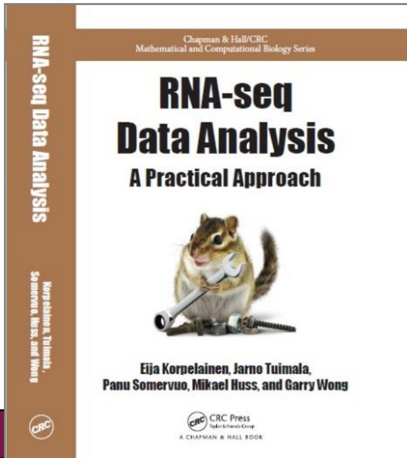
De novo transcriptome assembly (Chapter 5)

Genome-guided transcriptome assembly (Chapter 5)

Quantification (Chapter 6)

Annotation

Differential expression analysis (Chapters 8, 9, 13)



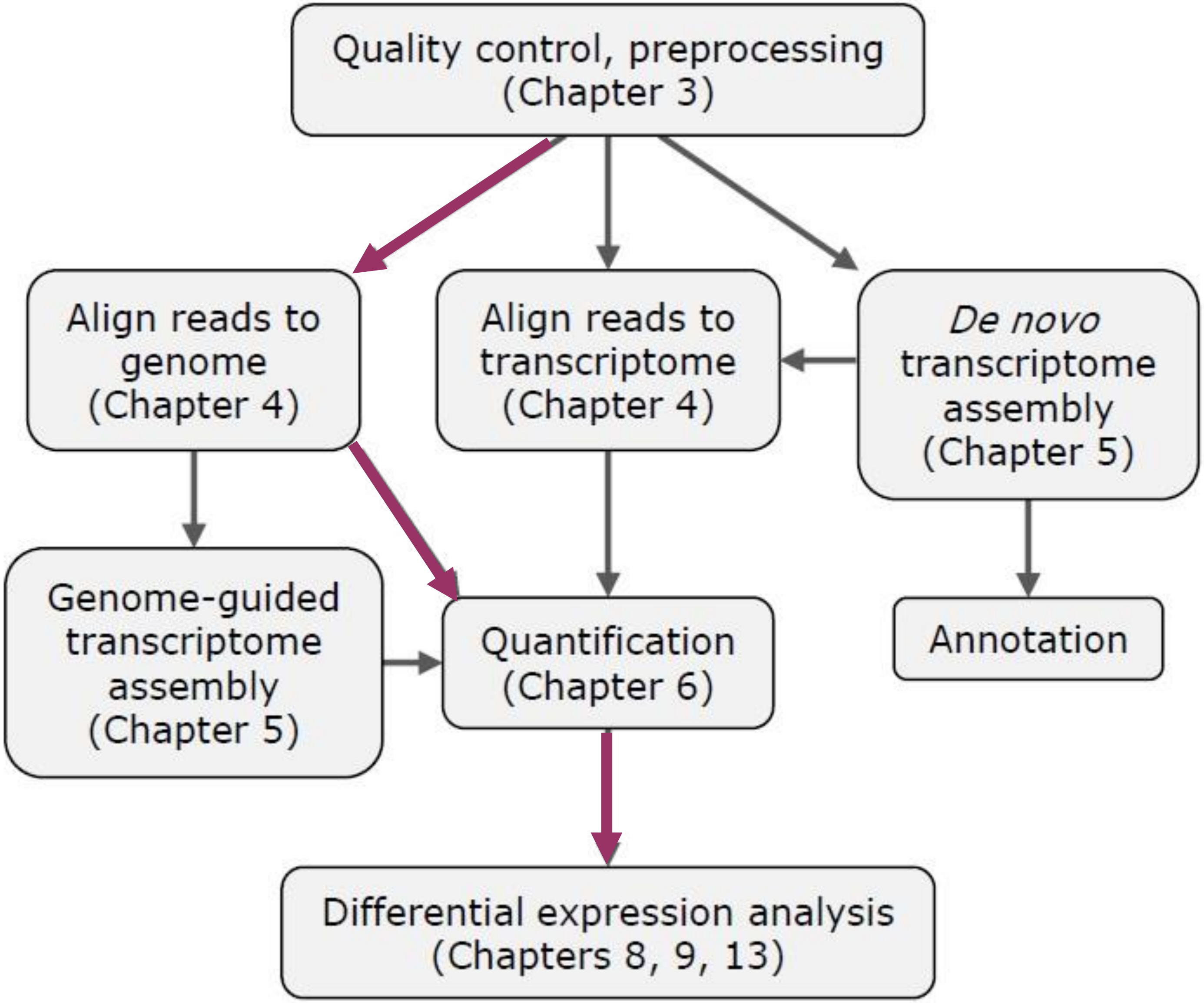
Chapman & Hall/CRC
Mathematical and Computational Biology Series

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

The steps we practise during the course


Quality control, preprocessing (Chapter 3)

Align reads to genome (Chapter 4)

Align reads to transcriptome (Chapter 4)

De novo transcriptome assembly (Chapter 5)

Genome-guided transcriptome assembly (Chapter 5)

Quantification (Chapter 6)

Annotation

Differential expression analysis (Chapters 8, 9, 13)

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**

Raw reads: FASTQ file format

- **Four lines per read:**

```
@read name
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+ 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)

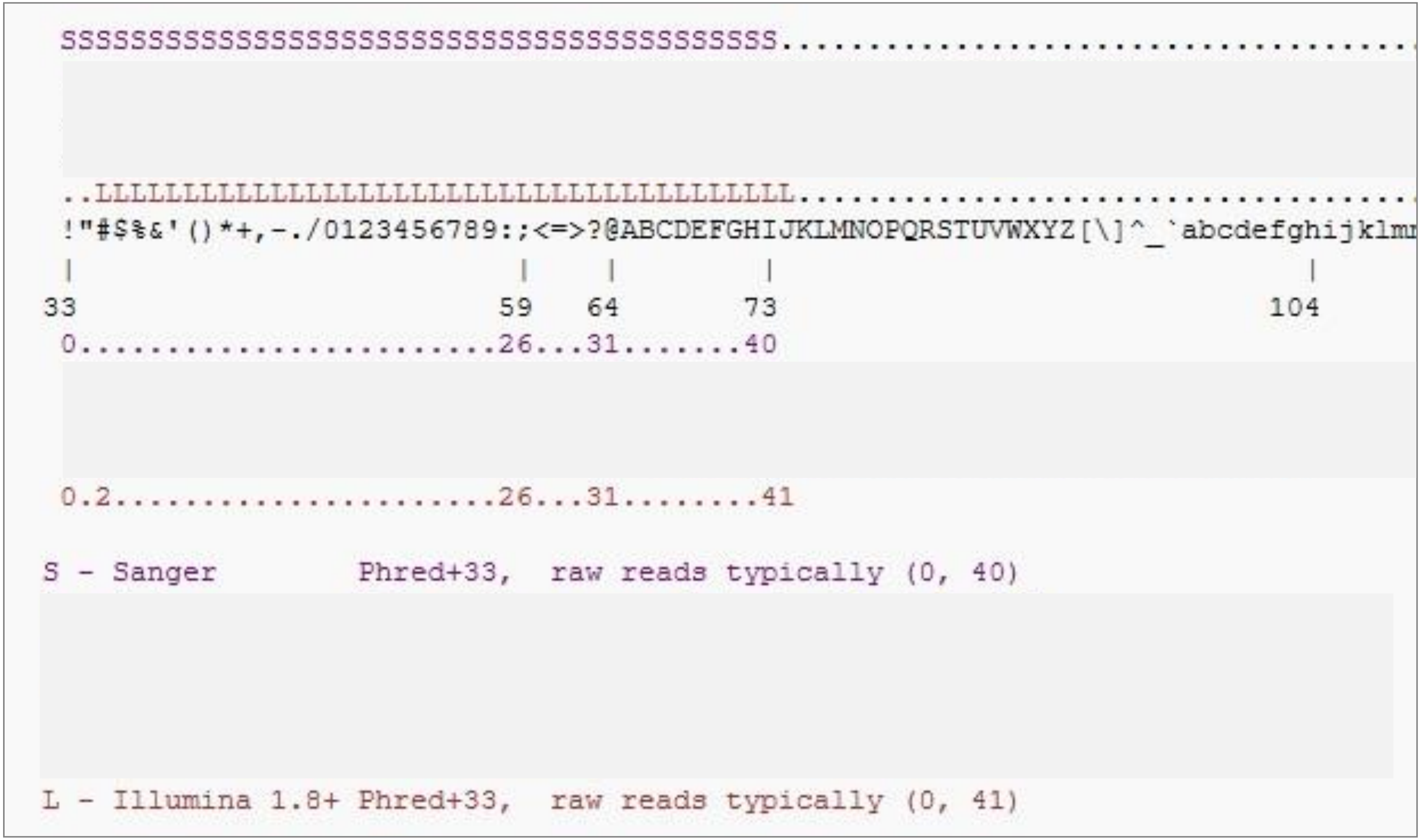
Base qualities

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

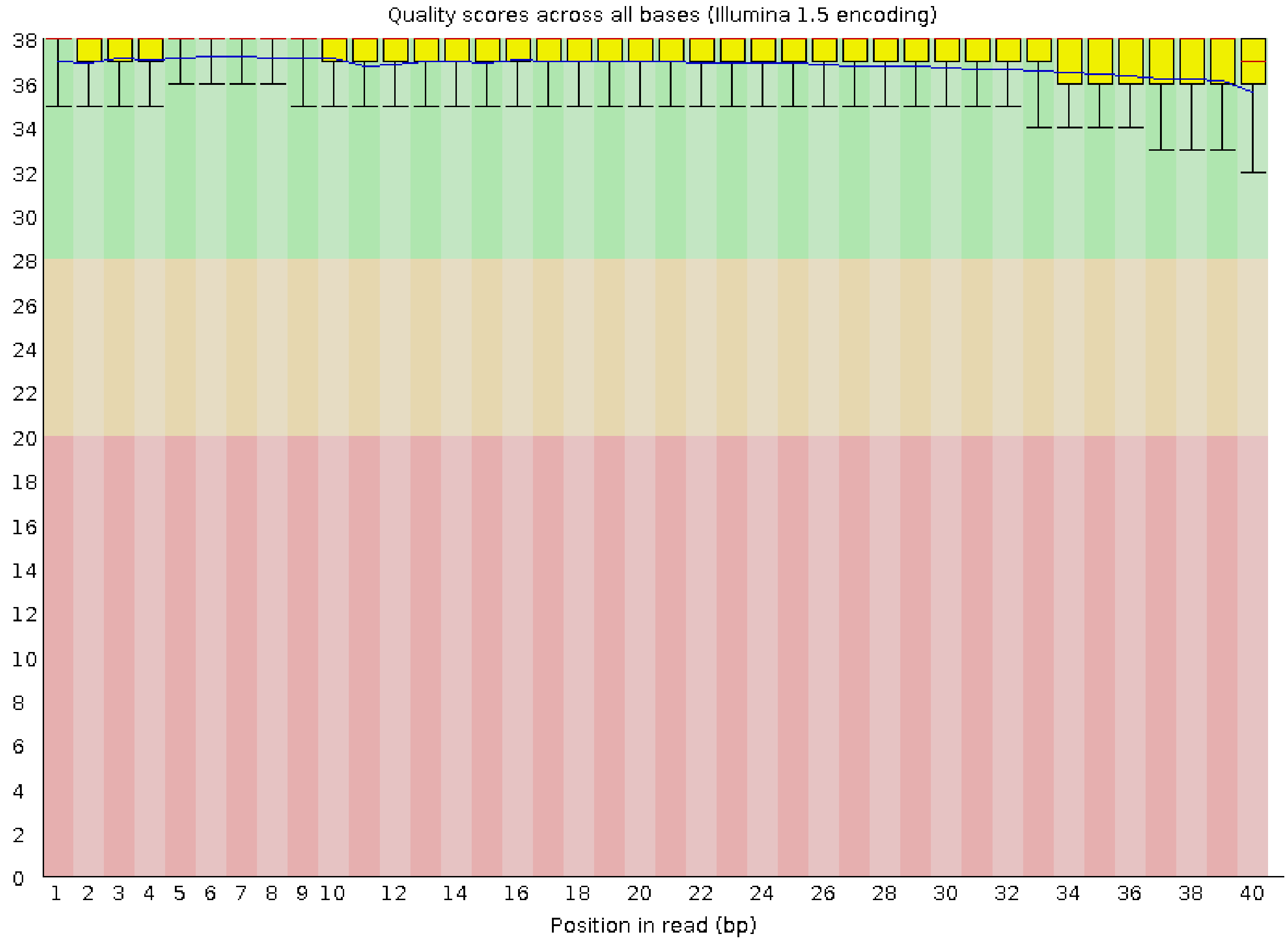
Base quality encoding systems



```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS....................................
 ..LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...................................
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklm
 |                         |    |         |                               |
33                        59   64        73                              104
 0........................26...31.......40
 0.2......................26...31........41

S - Sanger        Phred+33,  raw reads typically (0, 40)
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

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

Per position base quality (FastQC)

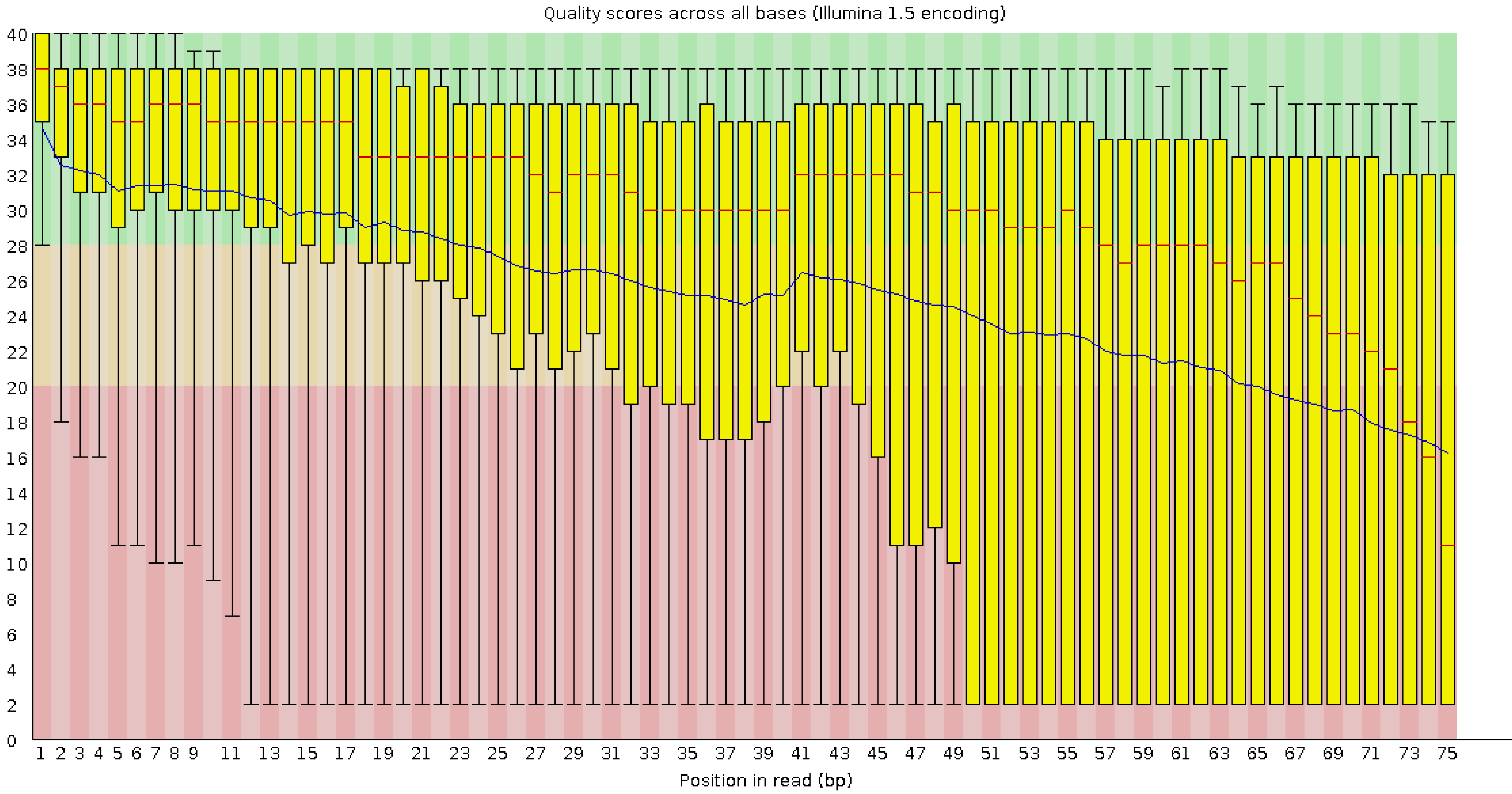


good

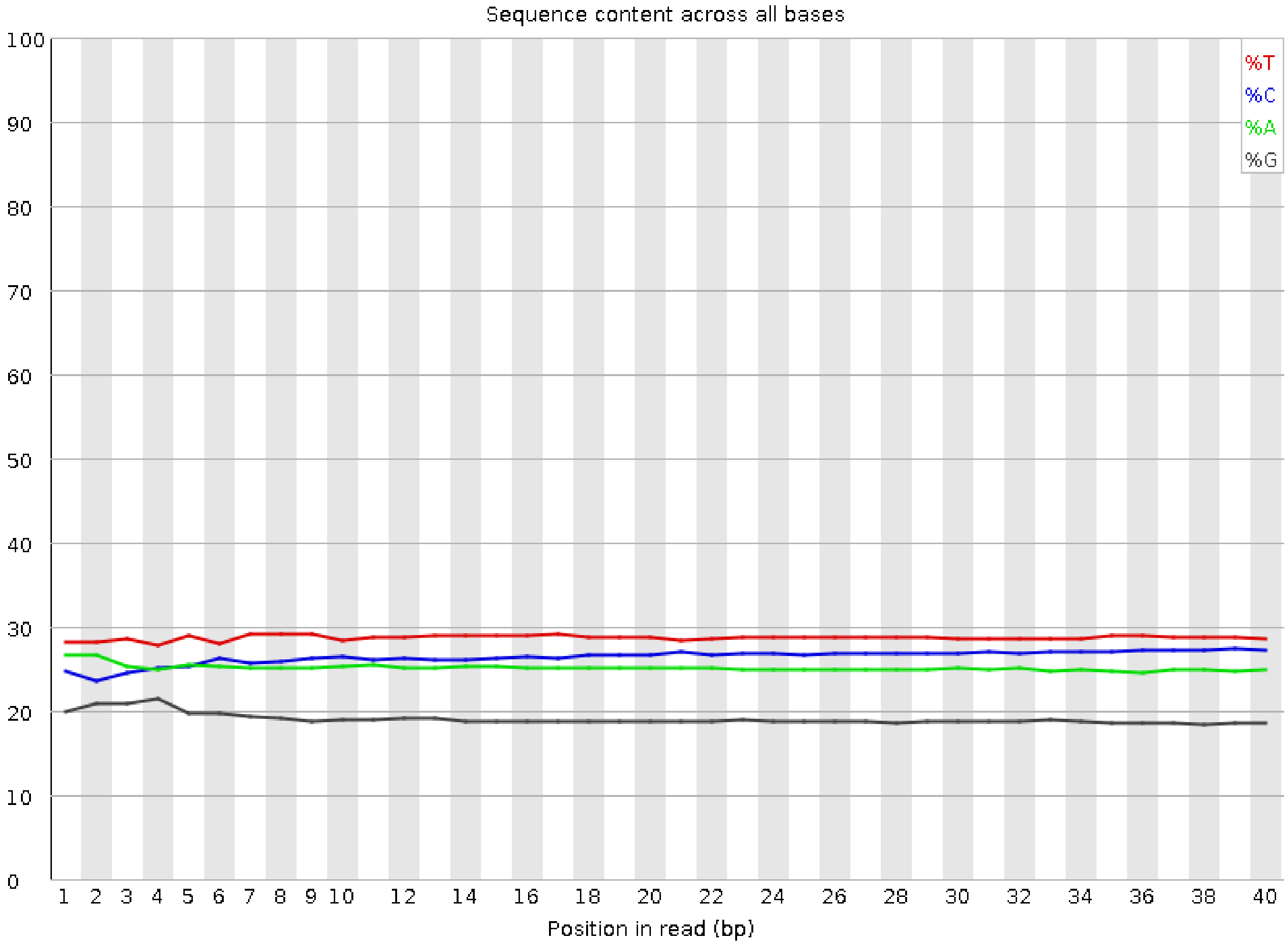
ok

bad

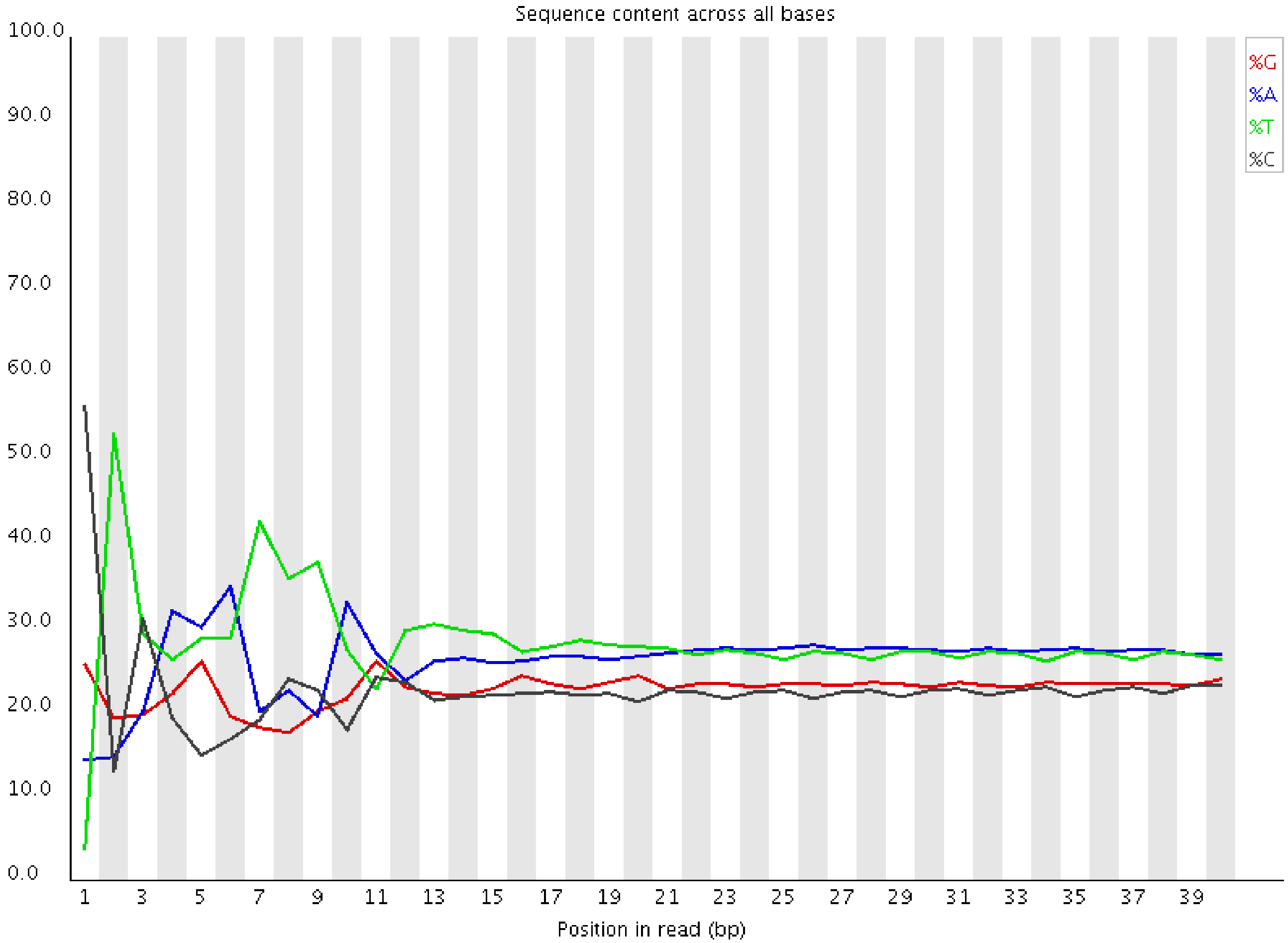
Per position base quality (FastQC)



Per position sequence content (FastQC)



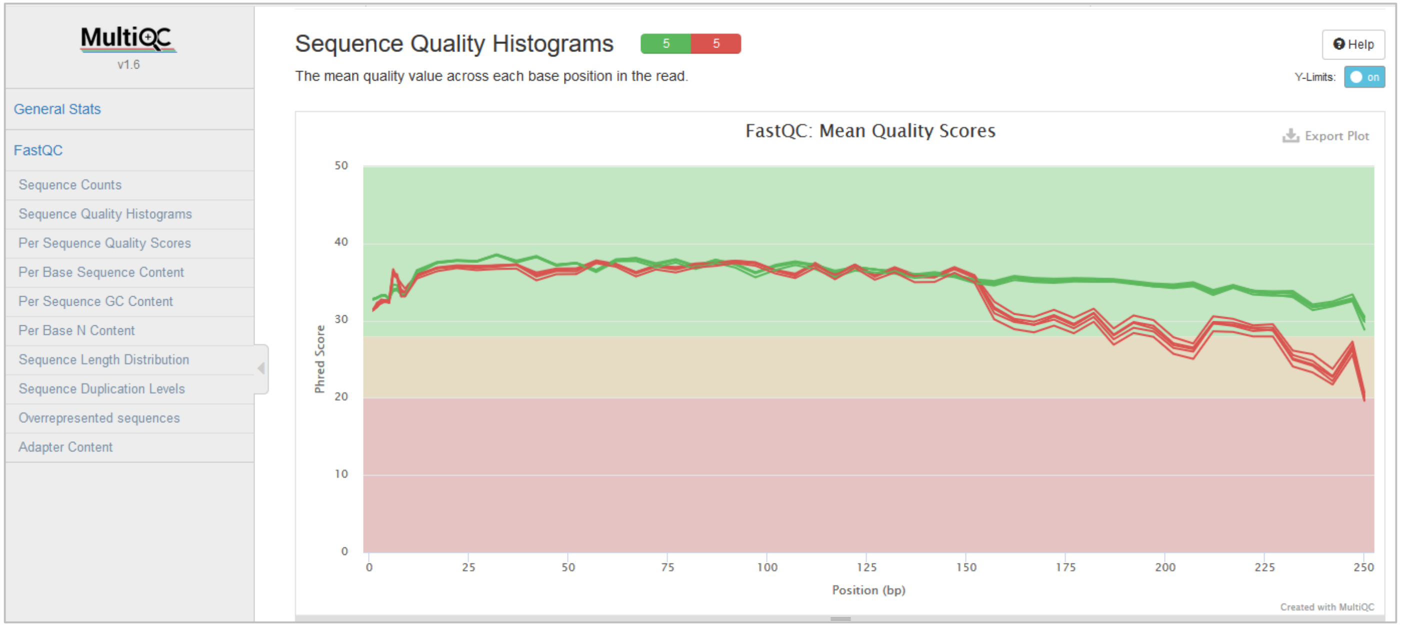
Per position sequence content (FastQC)



- 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**



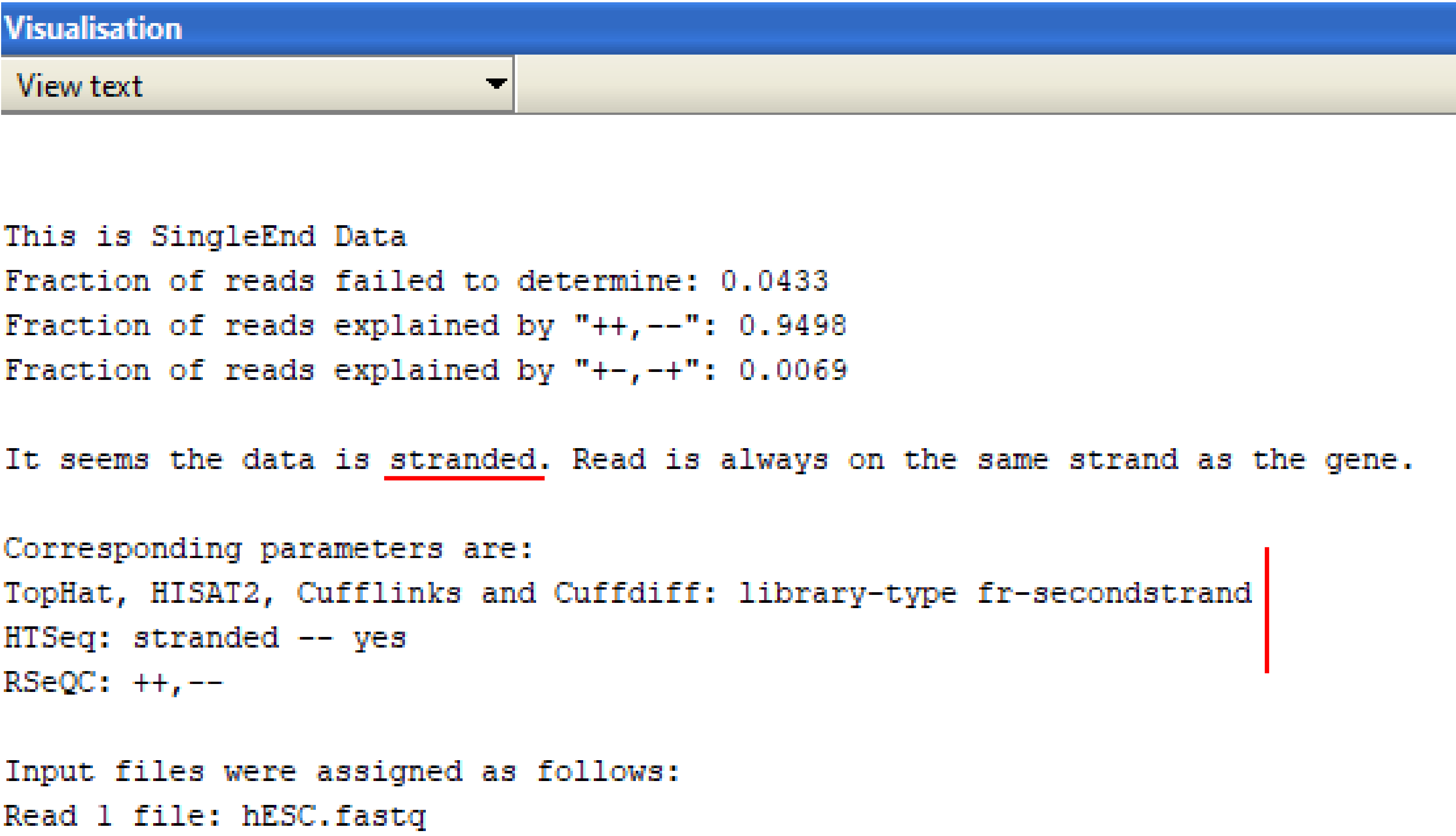
Sequence Quality Histograms

The mean quality value across each base position in the read.

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

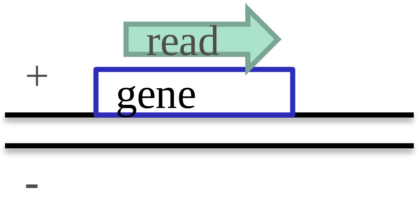


```
This is SingleEnd Data
Fraction of reads failed to determine: 0.0433
Fraction of reads explained by "++,--": 0.9498
Fraction of reads explained by "+-,-+": 0.0069

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
```

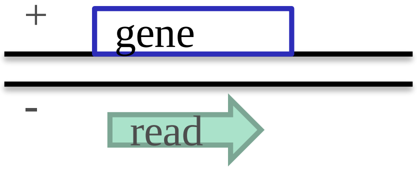
What does this ++, - - mean?

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
- Manipulation of alignment files
- Alignment level quality control
- Quantitation
- Experiment level quality control
- Visualization of reads and results in genomic context
- Differential expression analysis

Filtering and 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
- **Base quality threshold for trimming is a trade-off between having good quality reads and having enough sequence**
- **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 removed as well

RNA-seq data analysis workflow

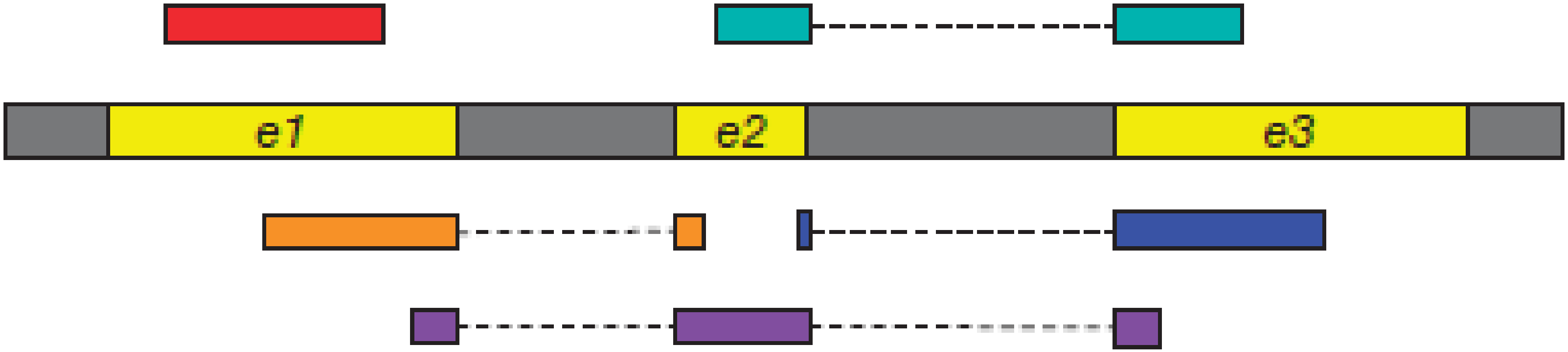
- **Quality control of raw reads**
- **Preprocessing (trimming / filtering) if needed**
- **Alignment (=mapping) to reference genome**
- **Manipulation of alignment files**
- **Alignment level quality control**
- **Quantitation**
- **Experiment level quality control**
- **Visualization of reads and results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

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



Modified from Kim et al (2015) Nature methods 12:358

Alignment programs

- **Many aligners have been developed over the years**
  - Convert genome fasta file to a data structure which 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

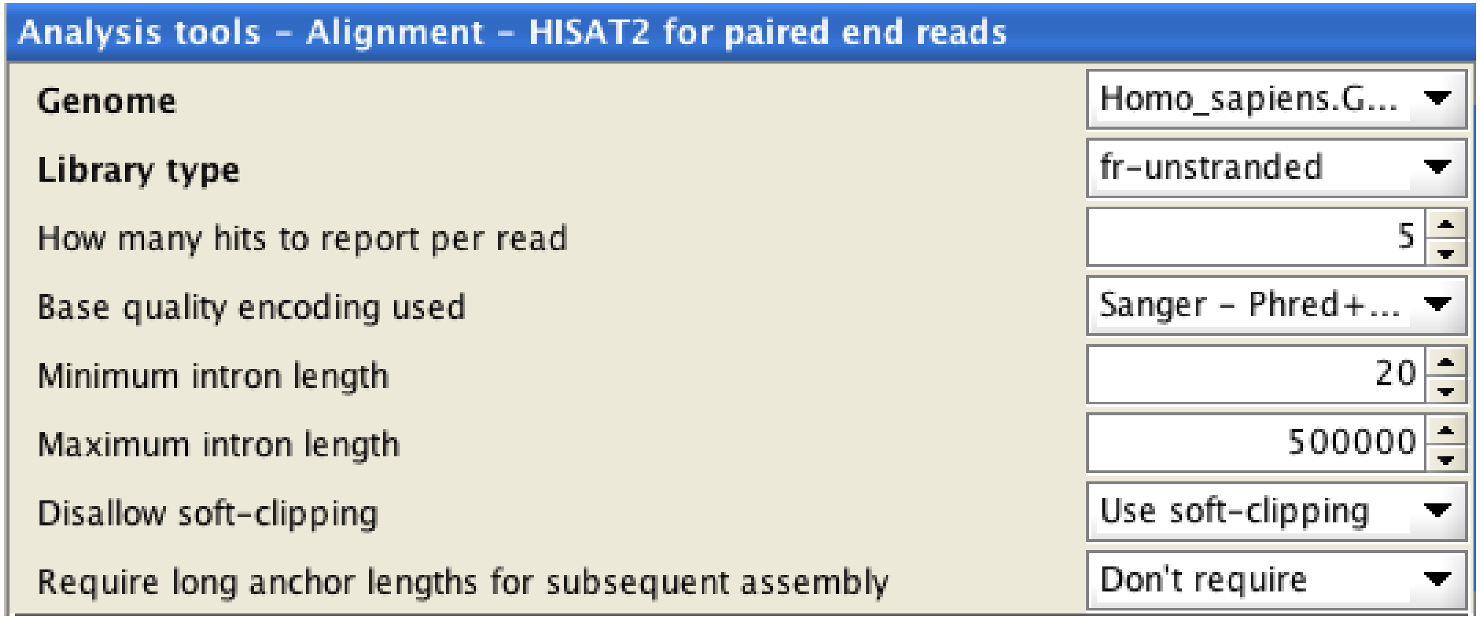
- **STAR**
  - Human genome available
- **HISAT2**
  - Human and mouse genome available
  - You can also supply own genome if it is small
- **TopHat2**
  - Many genomes available
  - You can also supply own genome

- **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

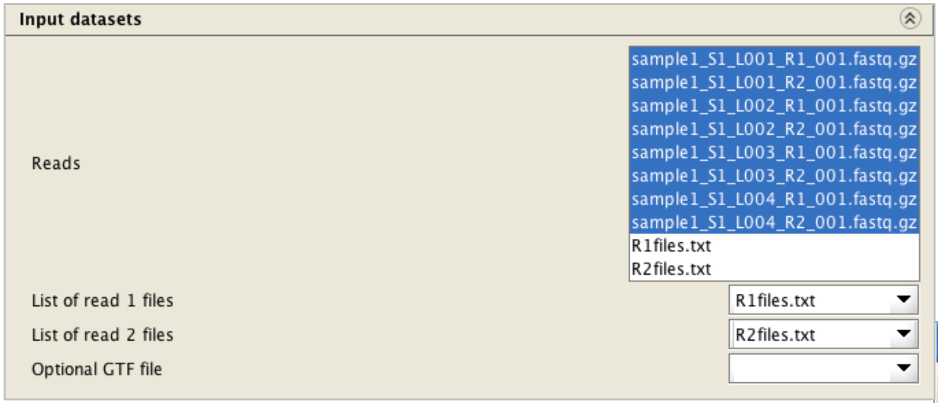
- Remember to set the strandedness (library type) correctly!
- Note that there can be alignments that are better than the 5 reported ones
- Require long anchors (> 16 bp) if you are going to do transcript assembly
- Soft-clipping = read ends don’t need to align to the genome, if this maximizes the alignment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



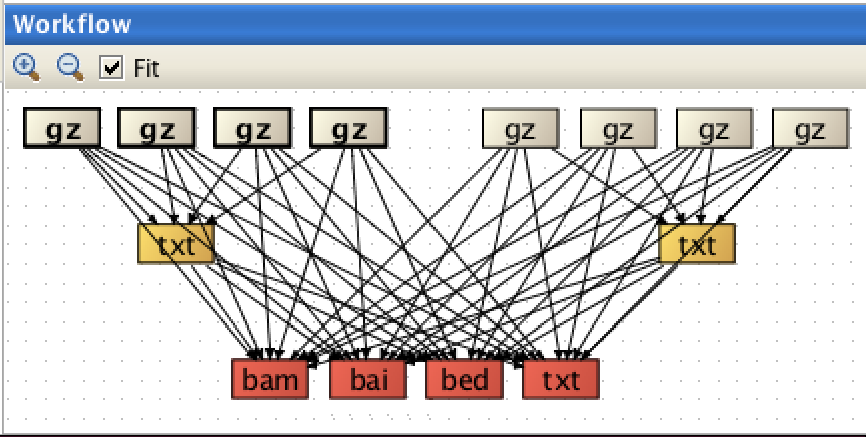
Input datasets

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

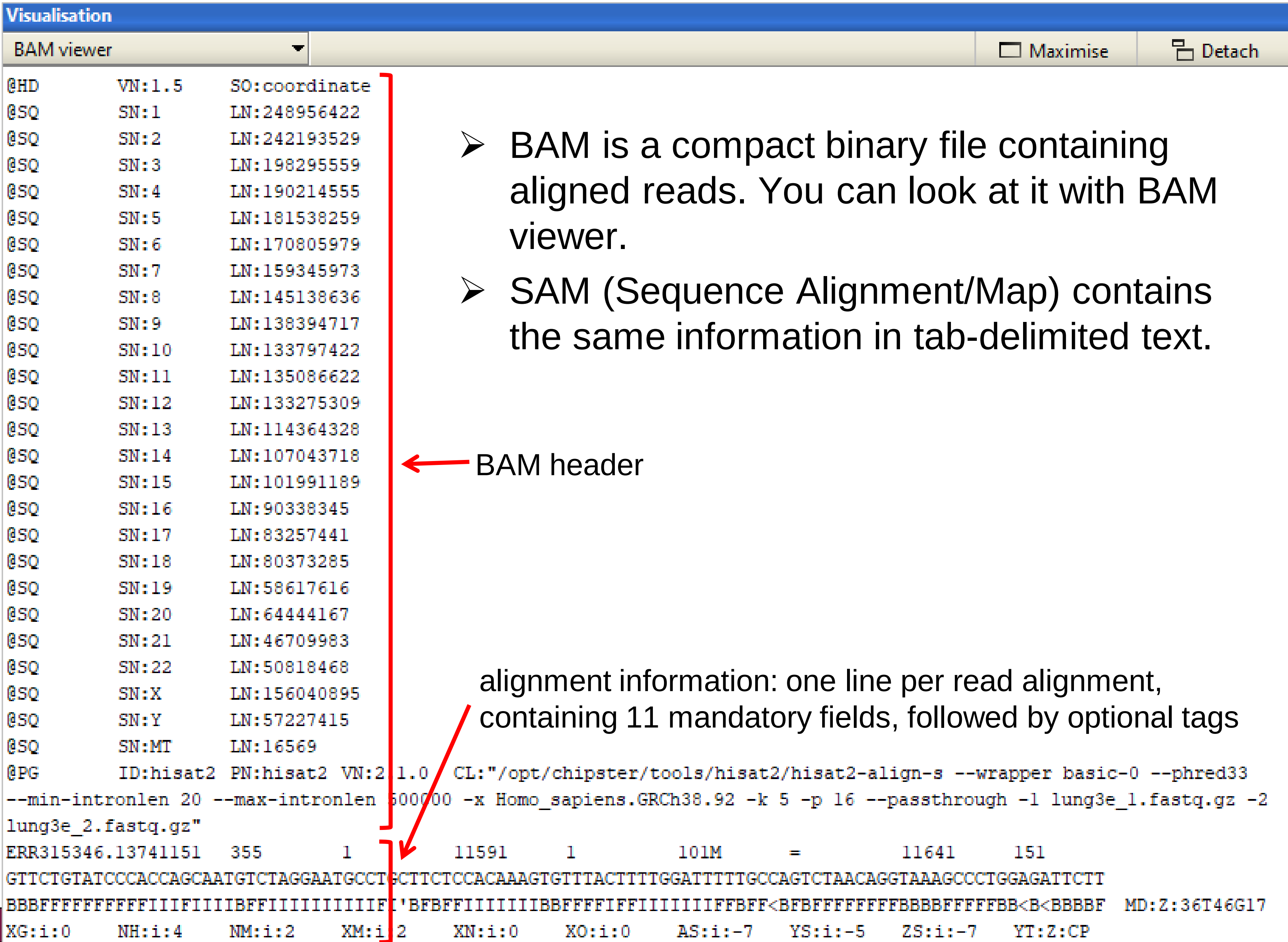
List of read 1 files: R1files.txt

List of read 2 files: R2files.txt

Optional GTF file:



File format for mapped reads: BAM/SAM



```
@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 -1 lung3e_1.fastq.gz -2
lung3e_2.fastq.gz"
ERR315346.13741151   355        1           11591       1           101M        =           11641       151
GTTCTGTATCCCACCAGCAATGTCTAGGAATGCCTGCTTCTCCACAAAGTGTTTACTTTTGGATTTTTGCCAGTCTAACAGGTAAAGCCCTGGAGATTCTT
BBBFFFFFFFFFFIIIFIIIIBFFIIIIIIIIIIIFI'BFBFFIIIIIIIIBBFFFFIFFIIIIIIIIFFBFF<BFBFFFFFFFFBBBBFFFFFBB<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
```

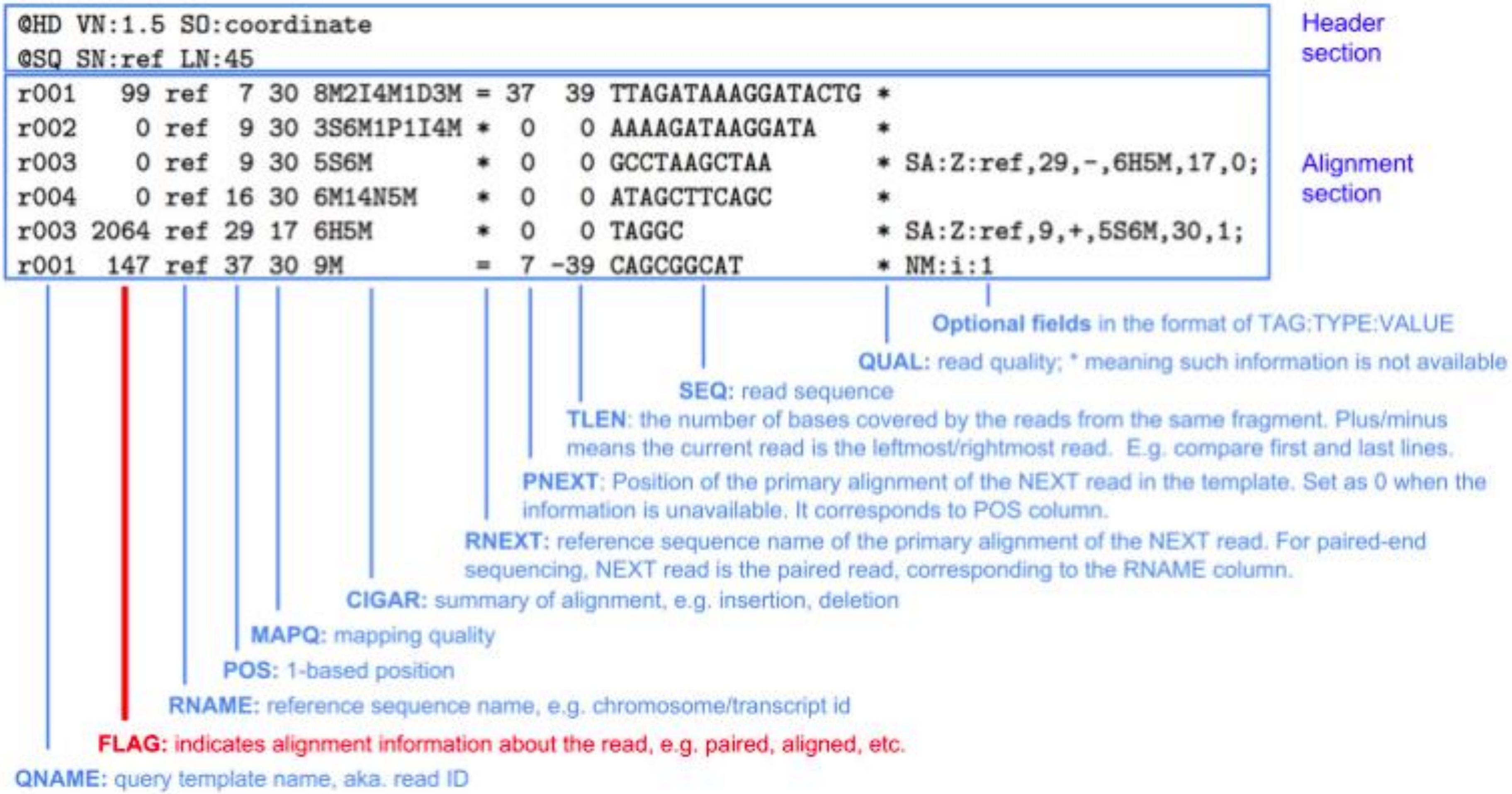
- ➢ BAM is a compact binary file containing aligned reads. You can look at it with BAM viewer.
- ➢ SAM (Sequence Alignment/Map) contains the same information in tab-delimited text.

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	AGGGCCGATCTTGGTGCCATCCAGGGGGCCTCTACAAGGATAATCTGACCTGCTGAAGATGTCTCCAGAGACCTT
base qualities	ECC@EEF@EB:EECFEECCCBEEEE;>5;2FBB@FBFEEFCF@FFFFCEFFFFFEE>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
```

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 * $\log_{10}$ (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:

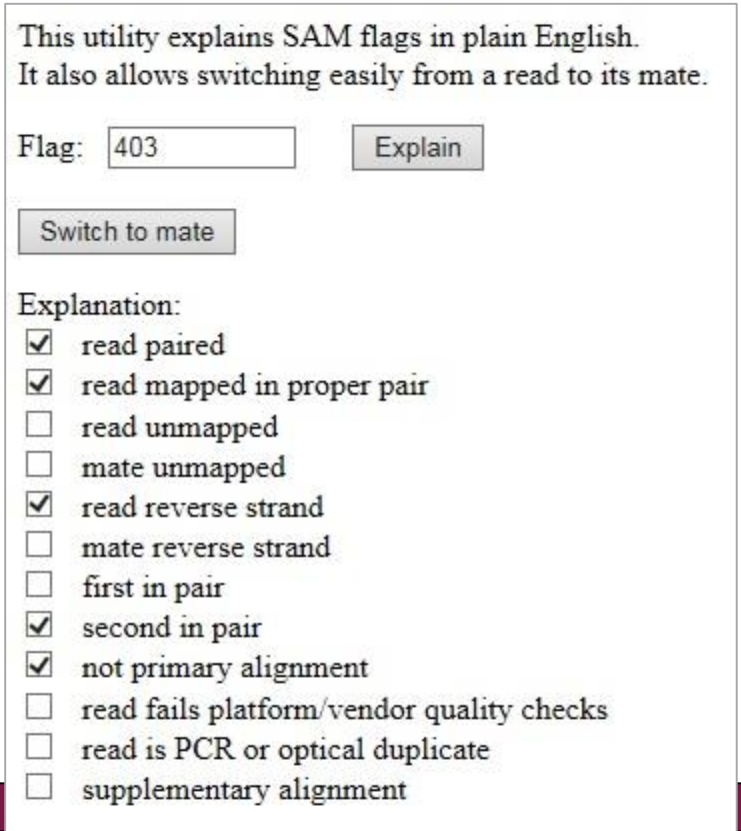
```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
r001  163  ref  7  30  8M2I4M1D3M   =   37   39   TTAGATAAAGGATACTG   *
```

  - The corresponding alignment

```
Ref  AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
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.

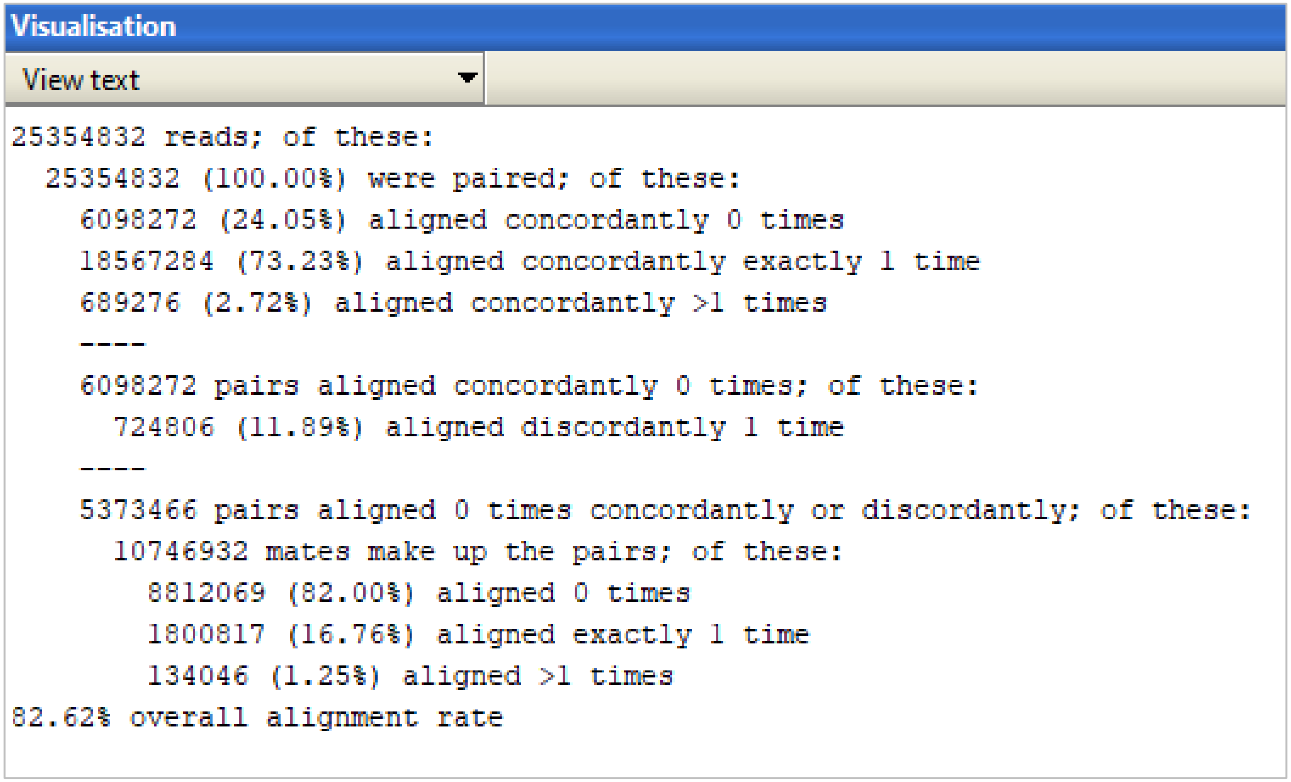
Flag: 403

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

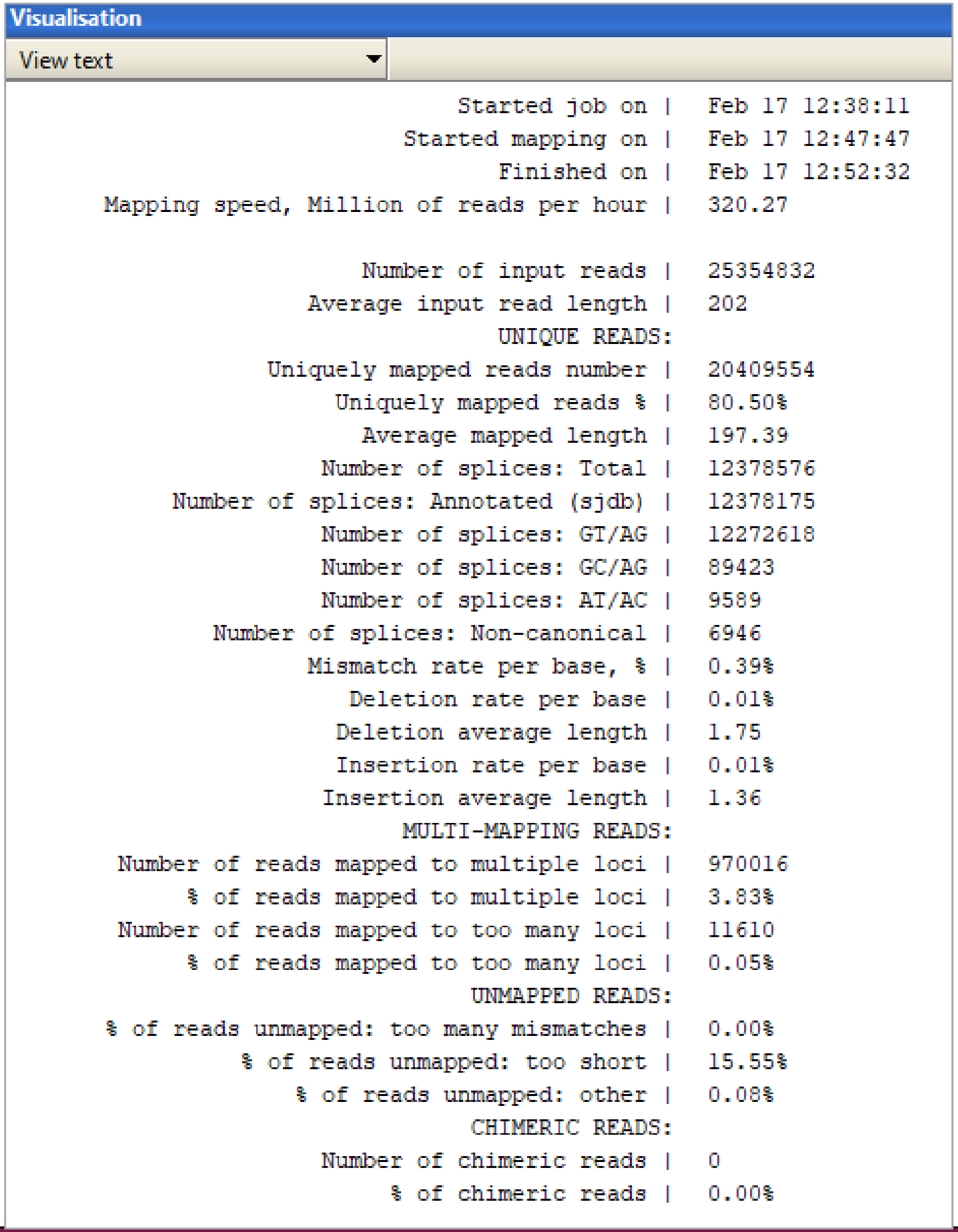
How did the alignment go? Check the log file

- **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?**



```
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
```

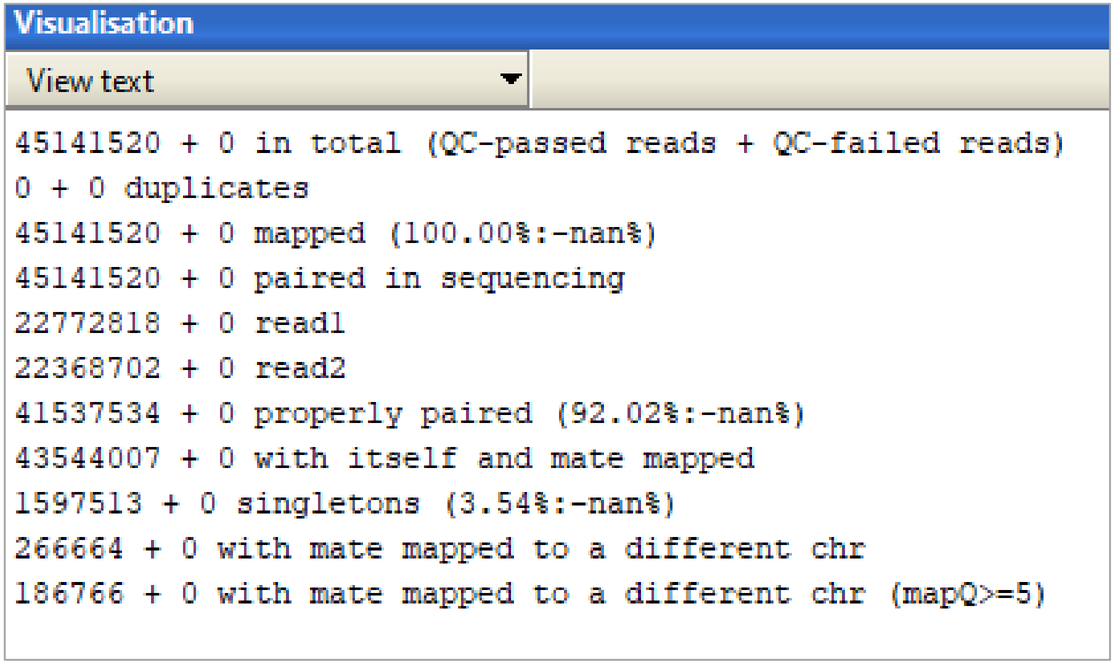
Log file by STAR



```
                          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**



```
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)
```

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

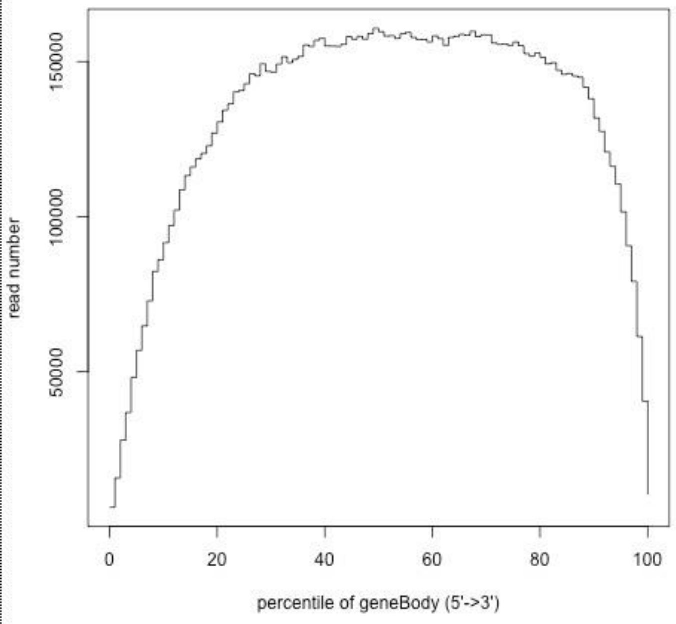
- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- Alignment (=mapping) to reference genome
- **Alignment level quality control**
- Quantitation
- Experiment level quality control
- Differential expression analysis
- Visualization of reads and results in genomic context

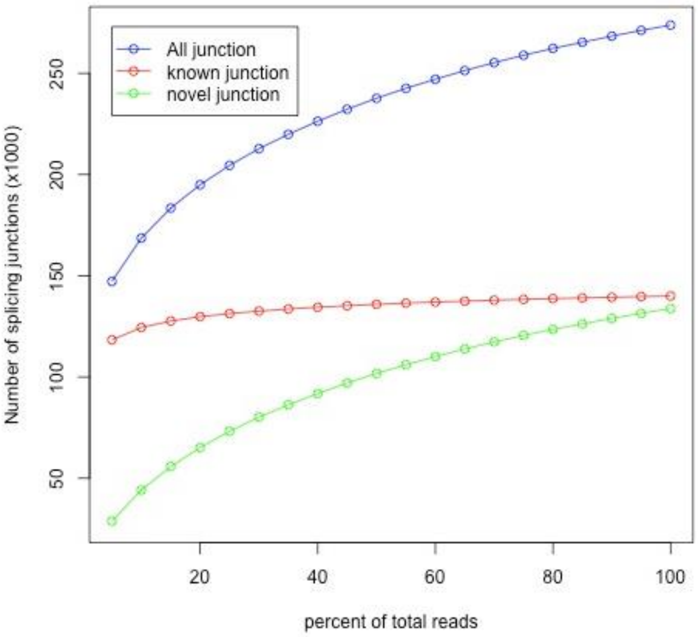
Annotation-based quality metrics

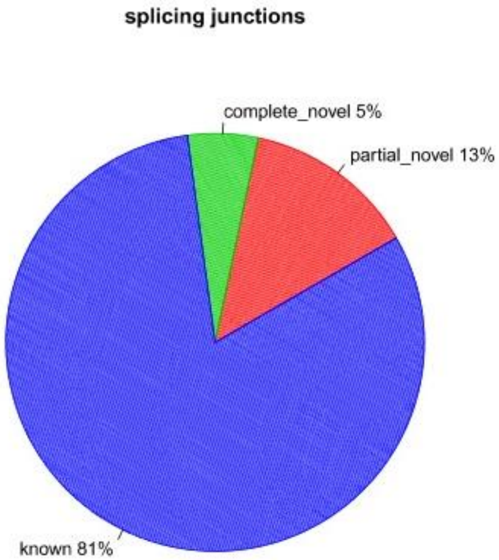
- **Saturation of sequencing depth**
  - Would more sequencing detect more genes and splice junctions?
- **Read distribution between different genomic features**
  - Exonic, intronic, intergenic regions
  - Coding, 3’ and 5’ UTR exons
  - Protein coding genes, pseudogenes, rRNA, miRNA, etc
- **Is read 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

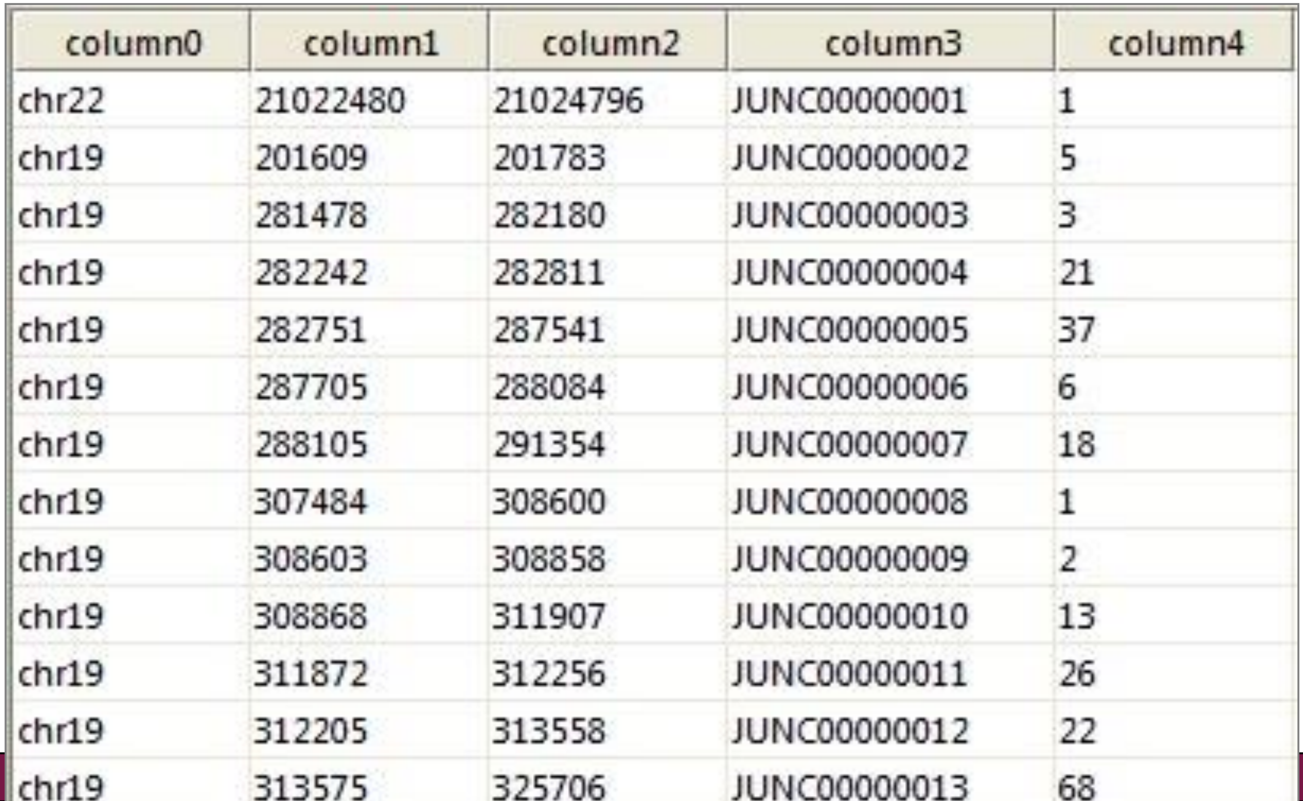






BED file format

- **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**

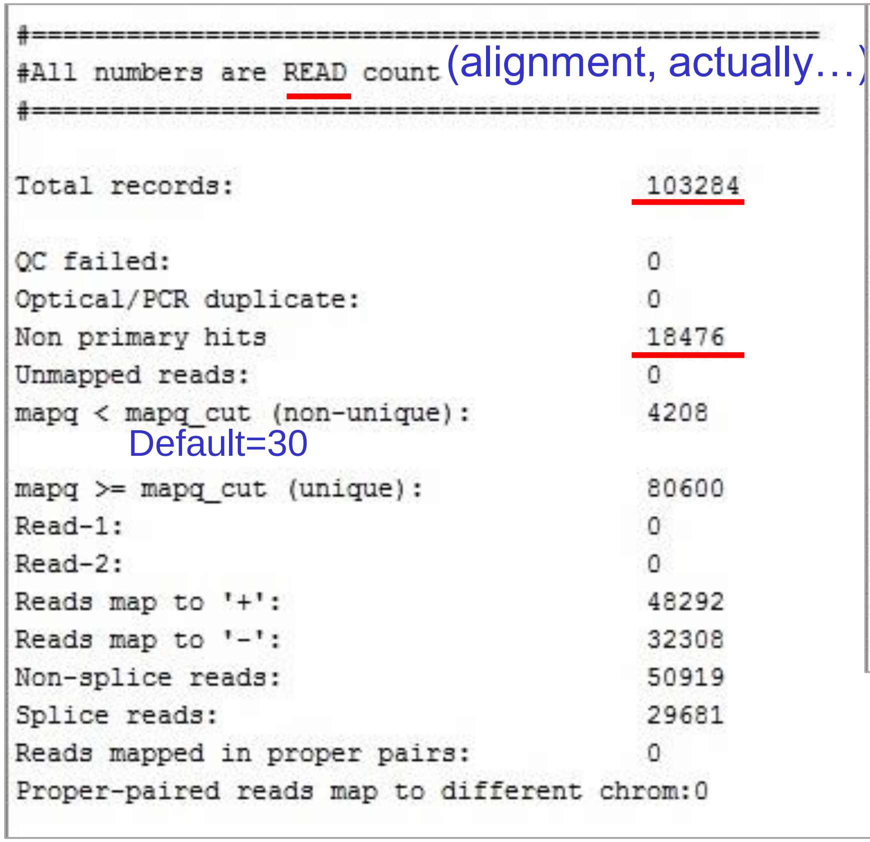


column0	column1	column2	column3	column4
chr22	21022480	21024796	JUNC00000001	1
chr19	201609	201783	JUNC00000002	5
chr19	281478	282180	JUNC00000003	3
chr19	282242	282811	JUNC00000004	21
chr19	282751	287541	JUNC00000005	37
chr19	287705	288084	JUNC00000006	6
chr19	288105	291354	JUNC00000007	18
chr19	307484	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

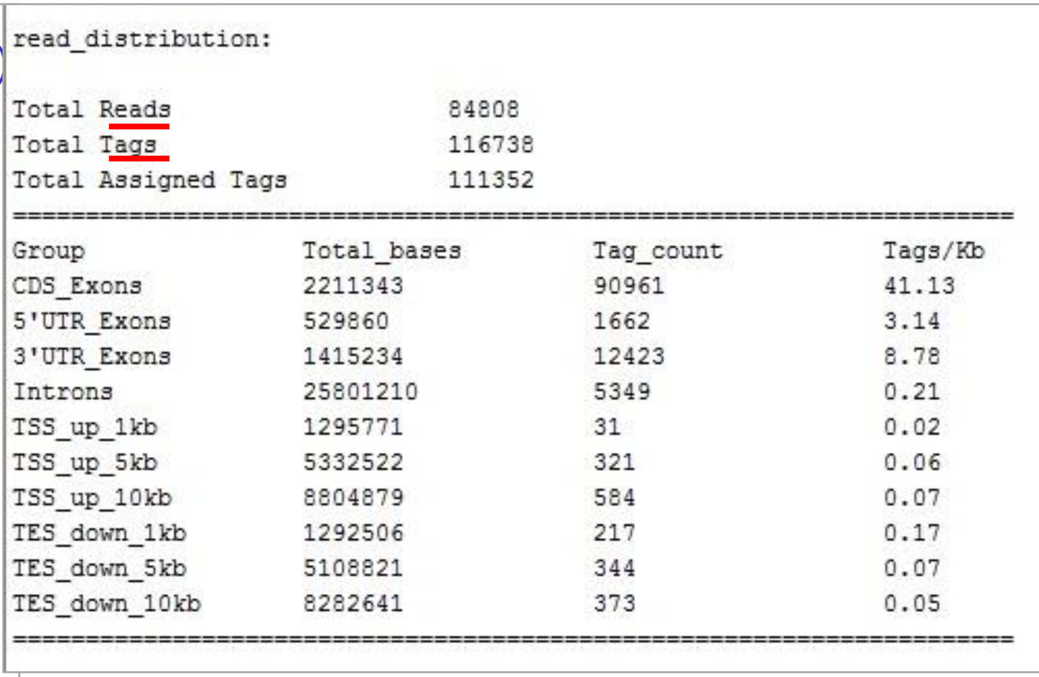
QC tables by RseQC



```
#==================================================
#All numbers are READ count
#==================================================

Total records:                          103284

QC failed:                              0
Optical/PCR duplicate:                  0
Non primary hits                        18476
Unmapped reads:                         0
mapq < mapq_cut (non-unique):           4208

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
```

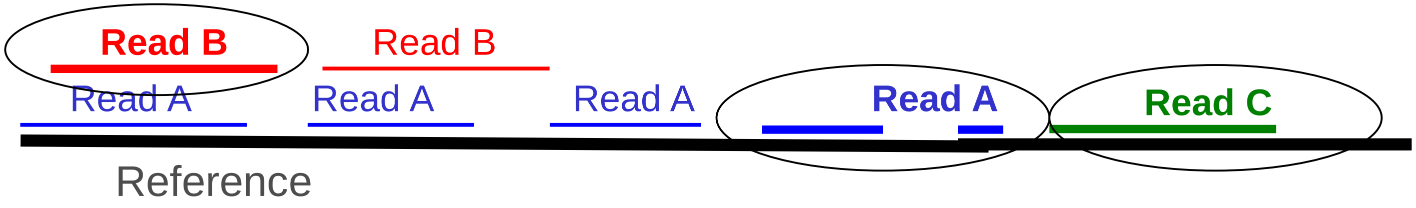
(alignment, actually…)

Default=30

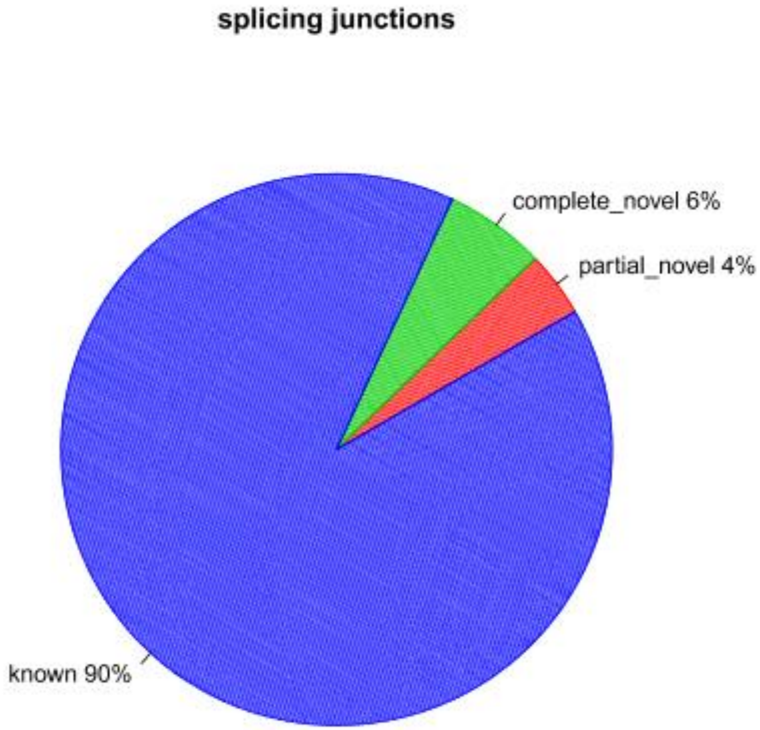


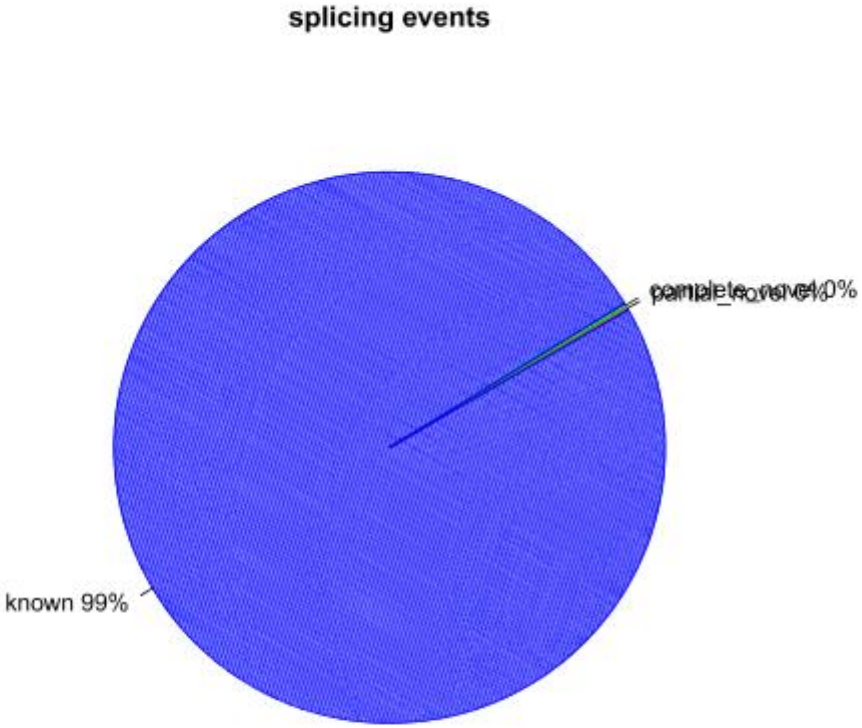
```
read_distribution:

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**
- **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**

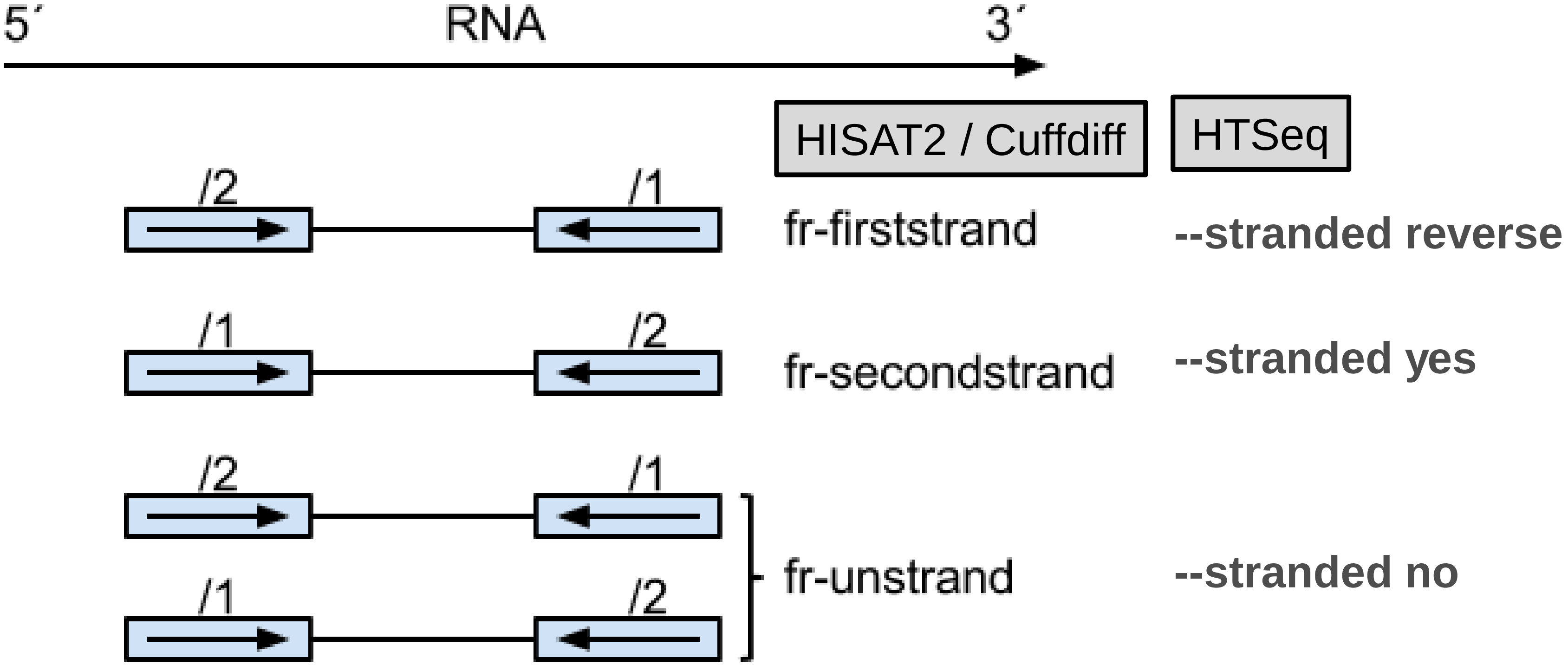
- **Kallisto**
- **Salmon**

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!

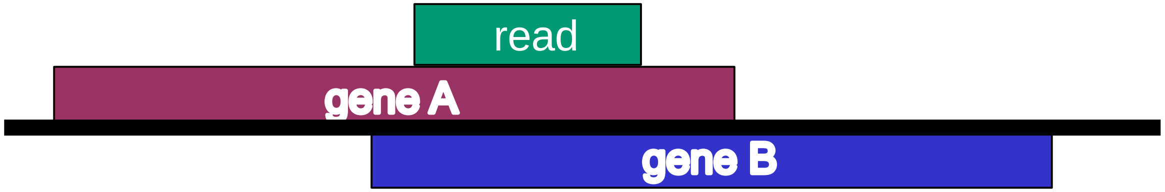
Stranded / directional RNA-seq data

- **Several protocols available**
  - TruSeq stranded, NEB Ultra Directional, Agilent SureSelect Strand-Specific…
- **Make sure that you set the strandedness parameter correctly**

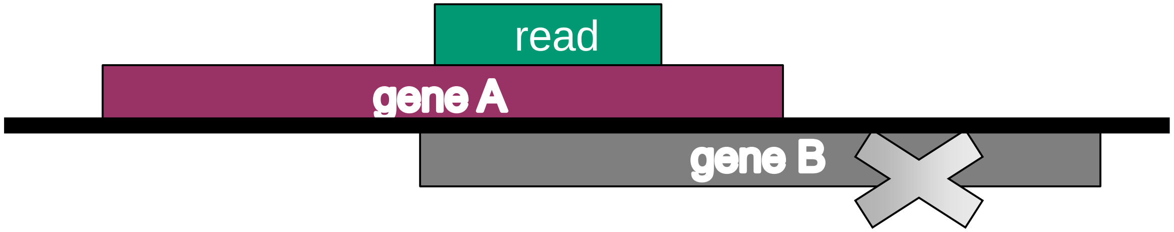


HISAT2 / Cuffdiff	HTSeq
fr-firststrand	**--stranded reverse**
fr-secondstrand	**--stranded yes**
fr-unstrand	**--stranded no**

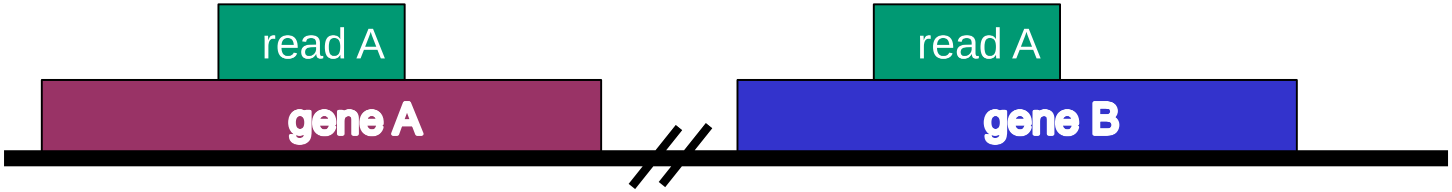
Not unique or ambiguous?



Ambiguous

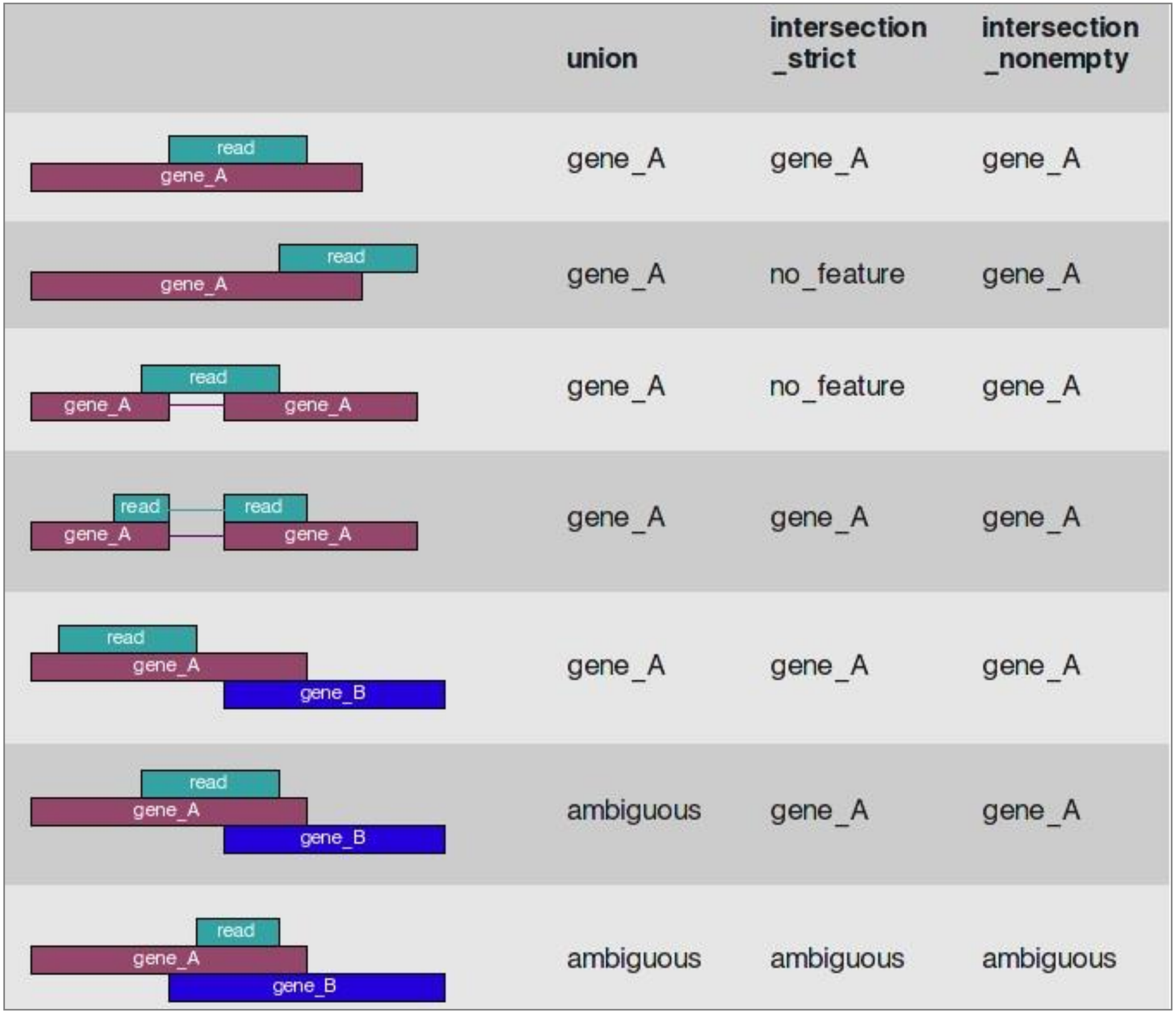


Stranded data
→ Not ambiguous



Multimapping
(not unique)

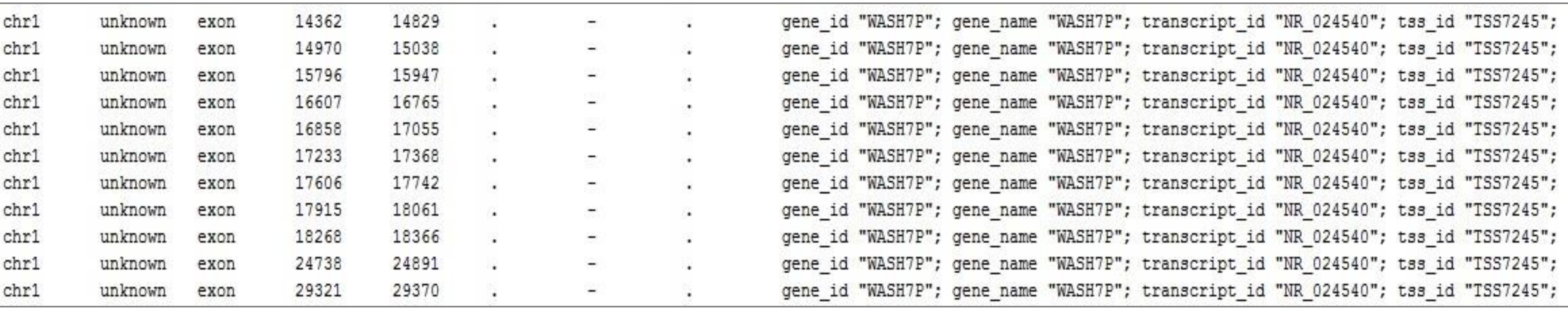
HTSeq count modes



	union	intersection _strict	intersection _nonempty
read gene_A	gene_A	gene_A	gene_A
read gene_A	gene_A	no_feature	gene_A
read gene_A gene_A	gene_A	no_feature	gene_A
read read gene_A gene_A	gene_A	gene_A	gene_A
read gene_A gene_B	gene_A	gene_A	gene_A
read gene_A gene_B	ambiguous	gene_A	gene_A
read gene_A gene_B	ambiguous	ambiguous	ambiguous

GTF file format

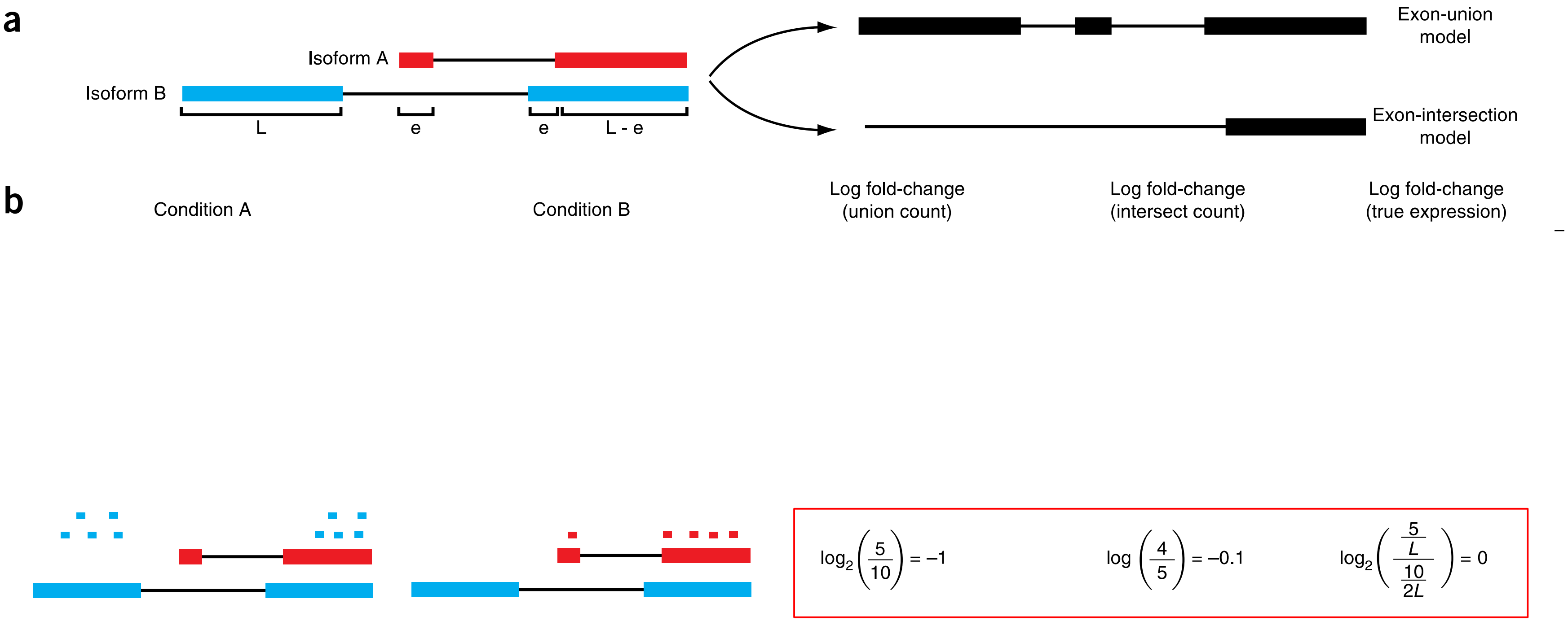
- **9 obligatory columns: chr, source, name, start, end, score, strand, frame, attribute**
- **1-based**
- **For HTSeq to work, all exons of a gene must have the same gene_id**
  - Use GTFs from Ensembl, avoid UCSC



```
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

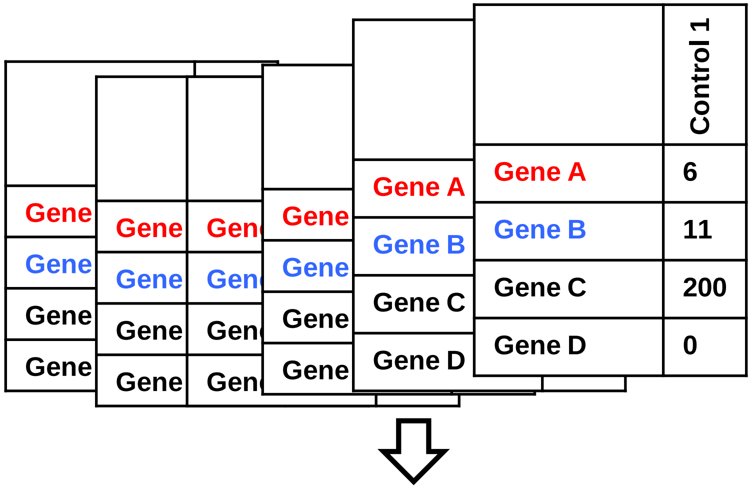
- the isoform switching problem


a
Isoform A
Isoform B
L
e
e
L - e
Exon-union model
Exon-intersection model
b
Condition A
Condition B
Log fold-change (union count)
Log fold-change (intersect count)
Log fold-change (true expression)
$\log_2\left(\frac{5}{10}\right) = -1$
$\log\left(\frac{4}{5}\right) = -0.1$
$\log_2\left(\frac{\frac{5}{L}}{\frac{10}{2L}}\right) = 0$

Trapnell et al. Nature Biotechnology 2013

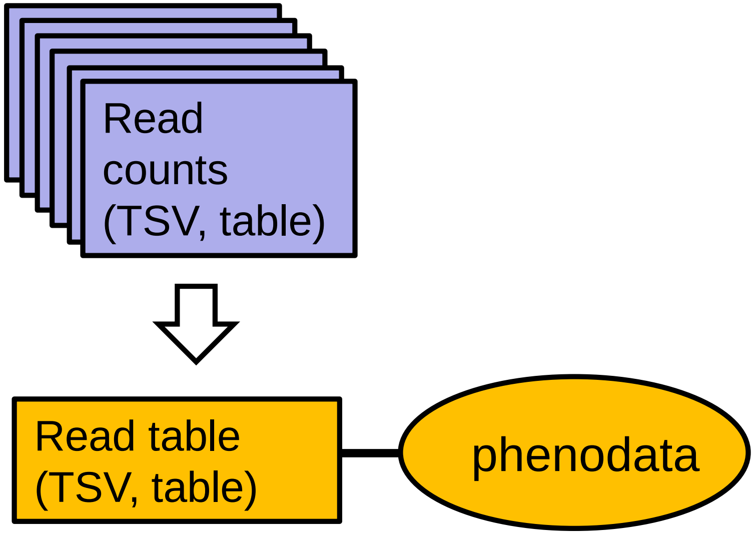
Combine individual count files into a count table

- **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**



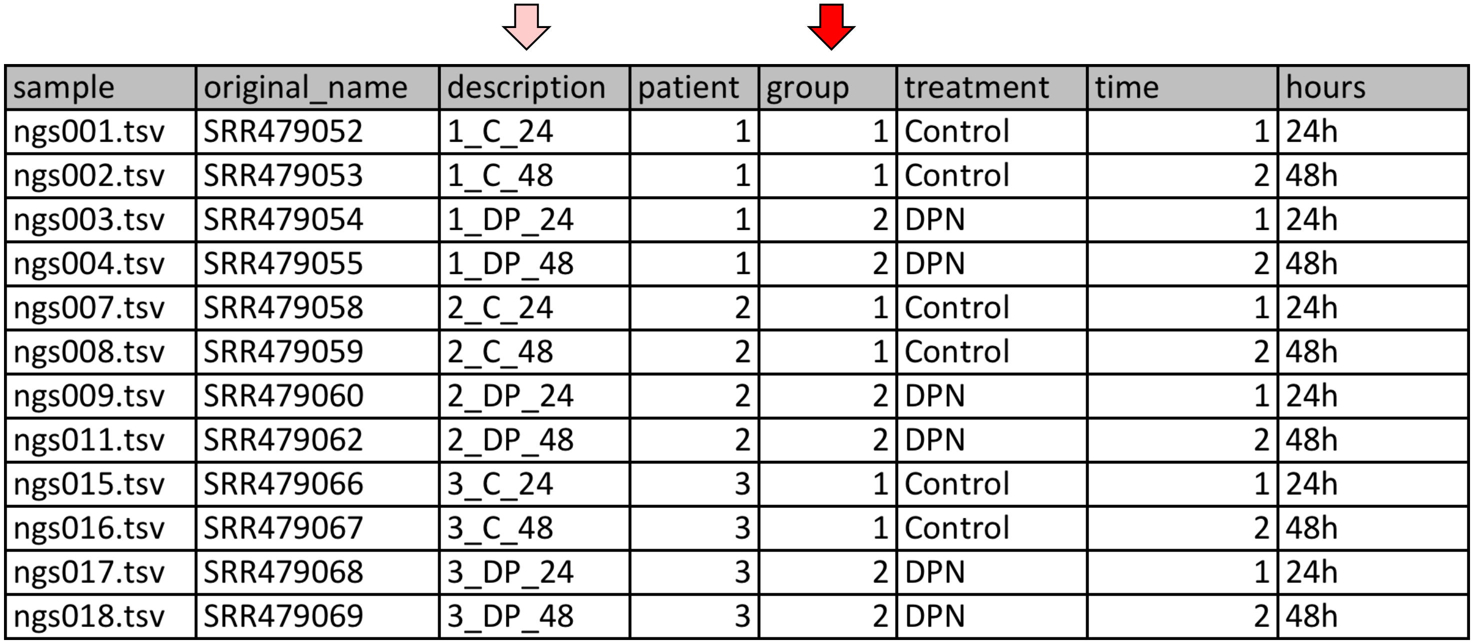
	Control 1
Gene A	6
Gene B	11
Gene C	200
Gene D	0

	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



Phenodata file: describe the experiment

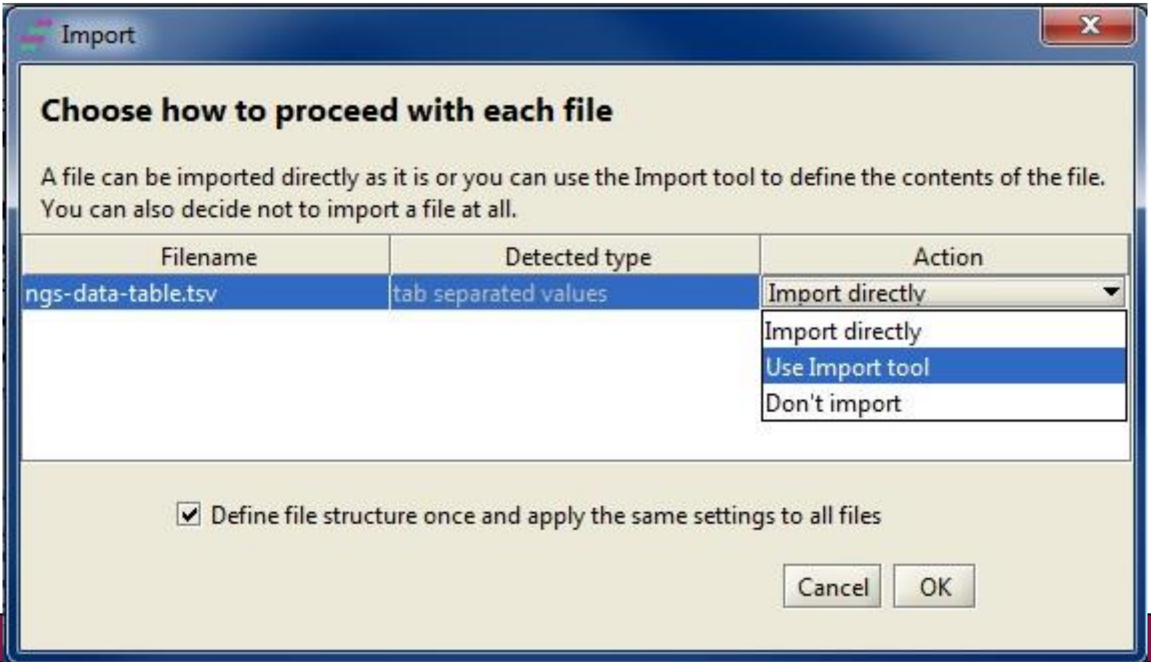
- **Describe experimental groups, time, pairing etc <u>with numbers</u>**
  - e.g. 1 = control, 2 = cancer
- **Define sample names for visualizations in the Description column**



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



Choose how to proceed with each file

A file can be imported directly as it is or you can use the Import tool to define the contents of the file. You can also decide not to import a file at all.

Filename	Detected type	Action
ngs-data-table.tsv	tab separated values	Import directly

☑ Define file structure once and apply the same settings to all files

RNA-seq data analysis workflow

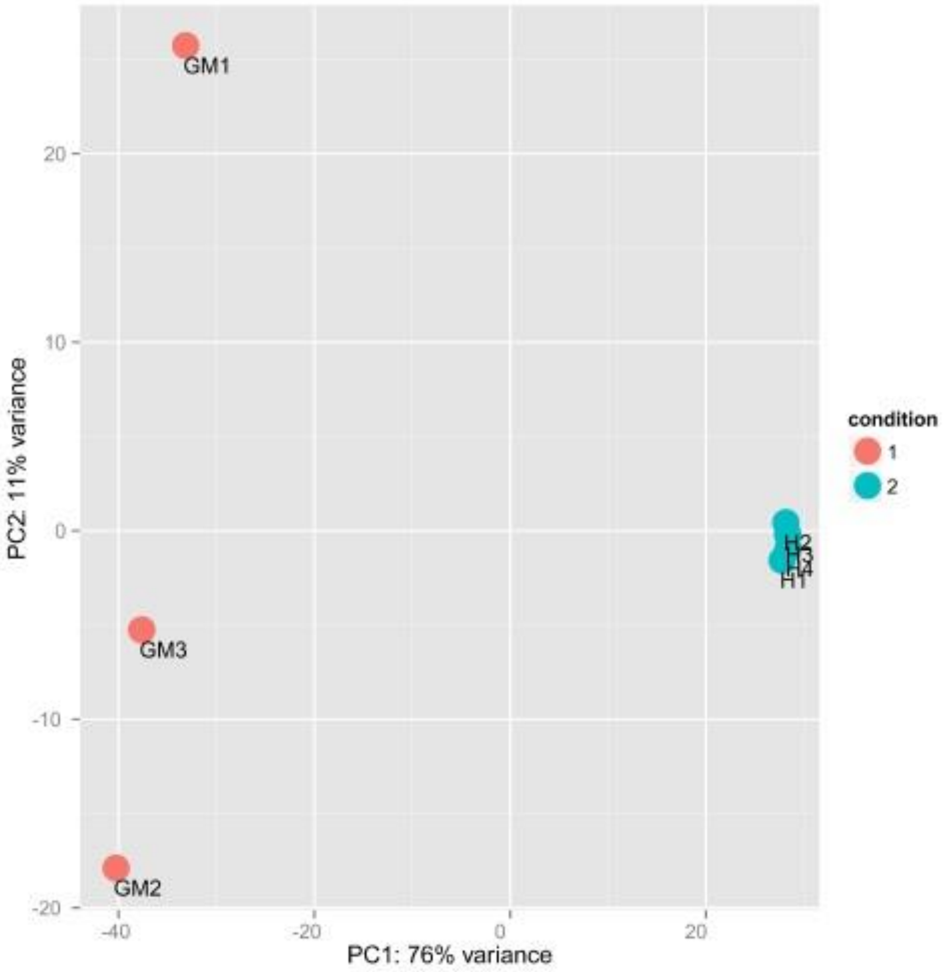
- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- Alignment (=mapping) to reference genome
- Alignment level quality control
- Quantitation
- **Experiment level quality control**
- Differential expression analysis
- Visualization of reads and results in genomic context

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



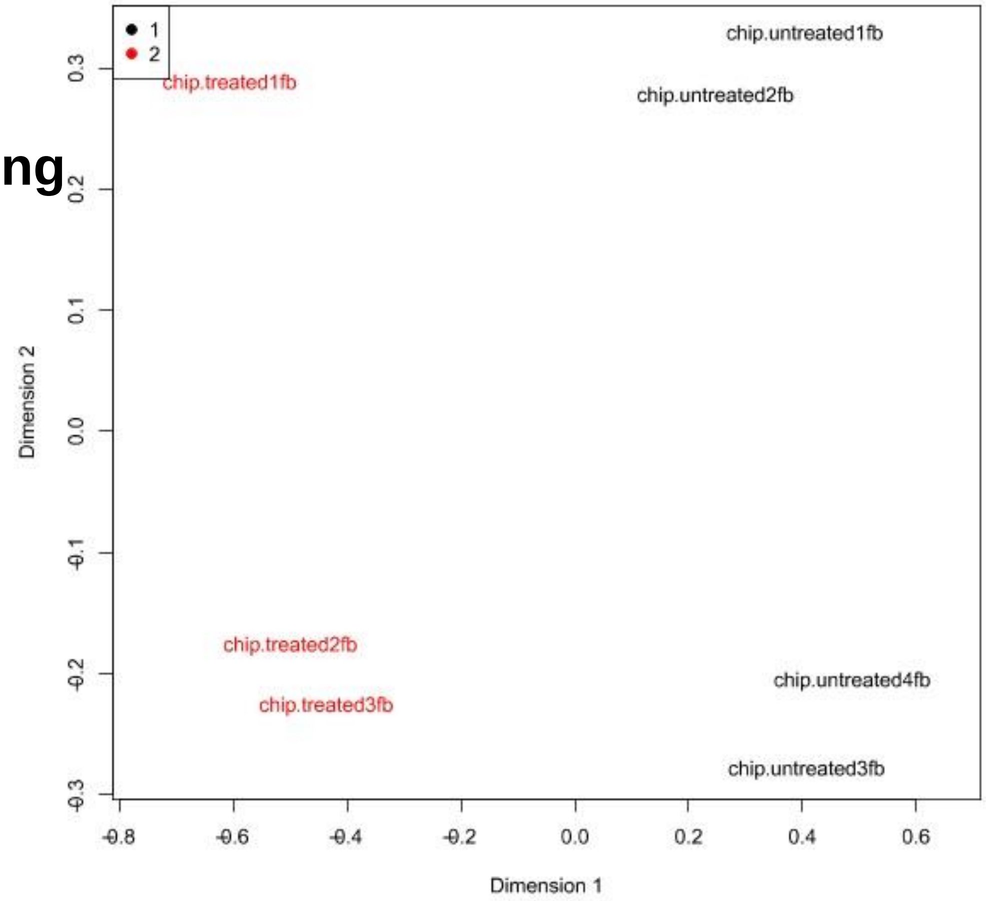
- **The first two principal components, calculated after variance stabilizing transformation**

- **Indicates the proportion of variance explained by each component**
  - If PC2 explains only a small percentage of variance, it can be ignored

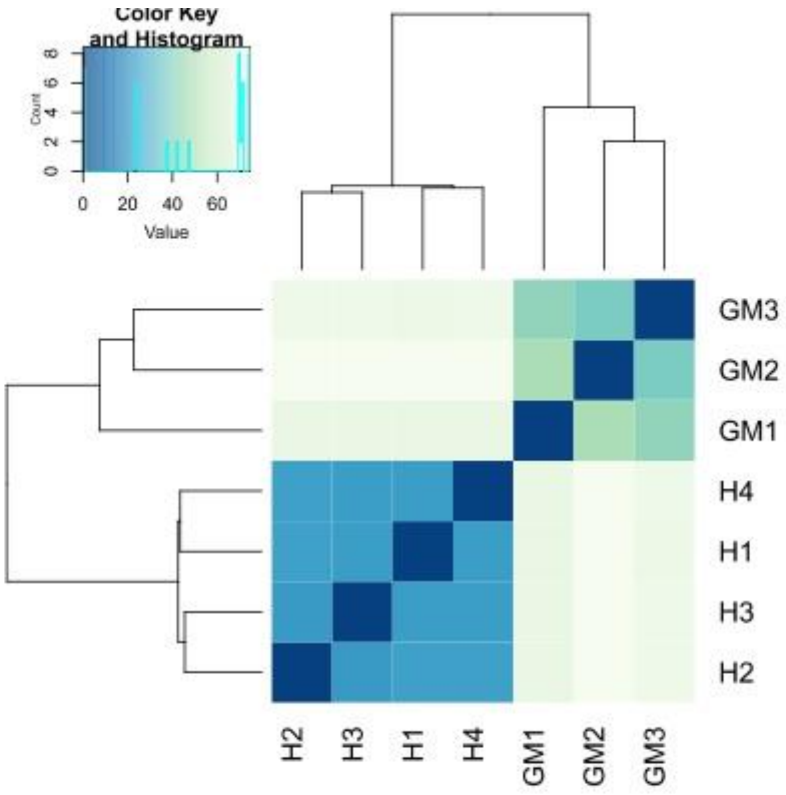
MDS plot by edgeR

- Distances correspond to the logFC or biological coefficient of variation (BCV) between each pair of samples

- Calculated using 500 most heterogenous genes (= 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
- Manipulation of alignment files
- Alignment level quality control
- Quantitation
- Experiment level quality control
- **Differential expression analysis**
- Visualization of reads and results in genomic context

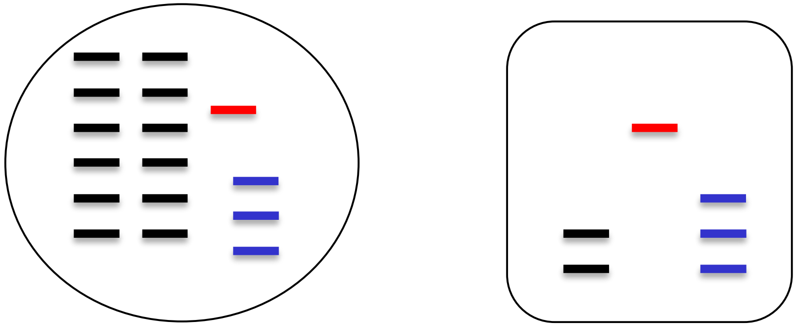
Differential gene expression analysis

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

Differential expression analysis: Normalization

Normalization

- **For comparing gene expression <u>between (groups of) samples</u>, 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

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?

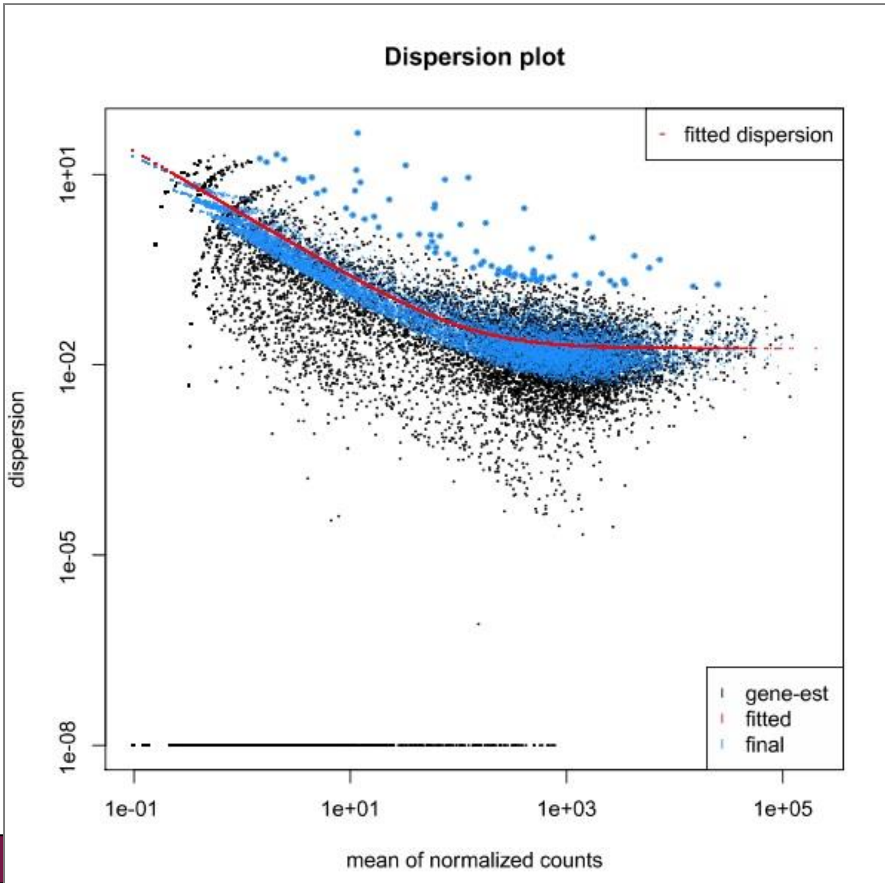
- **RNA-seq experiments typically have only few replicates**
  → it is difficult to estimate within-group variability

- **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 · group) + (c · time) + (d · 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

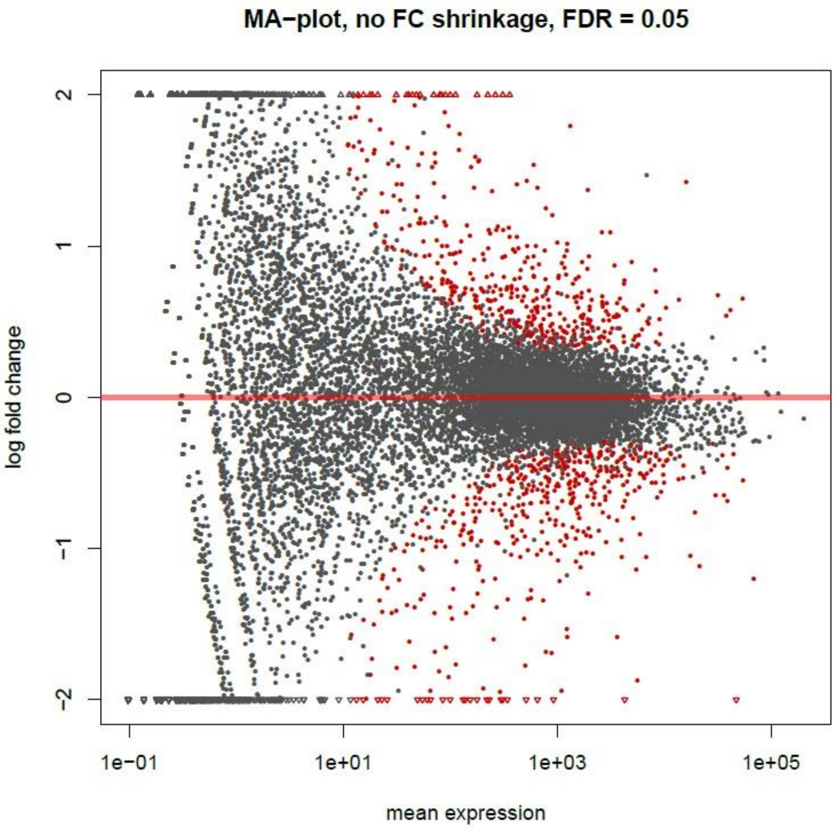
- **Generalized linear model (GLM) allows the expression value distribution to be different from normal distribution**

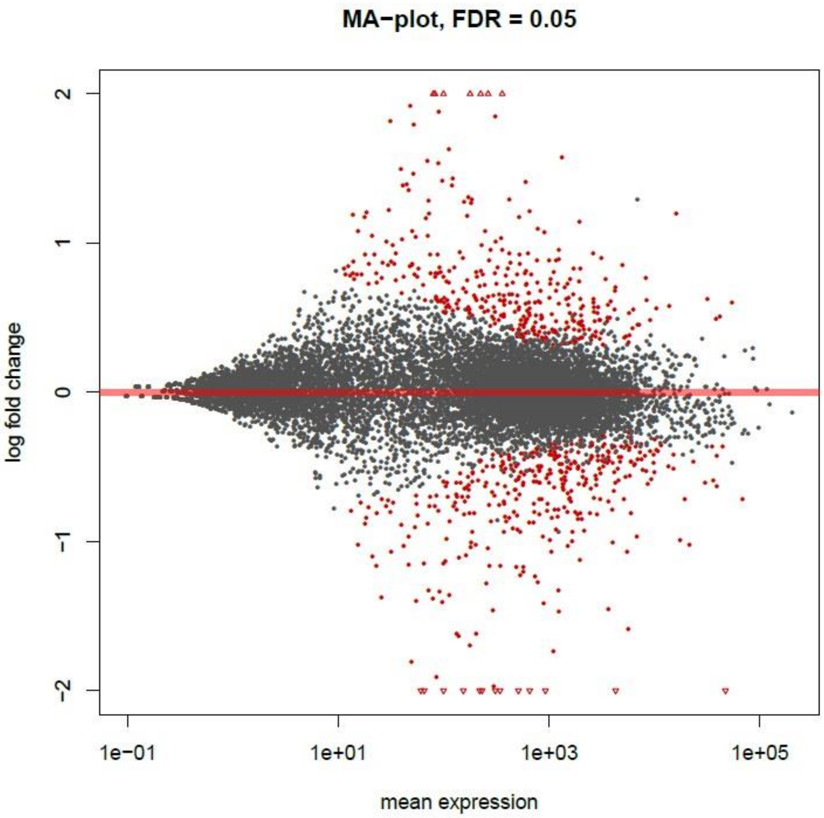
  - Negative binomial distribution used for count data

Statistical testing

- **edgeR**
  - Two group comparisons
    - Exact test for negative binomial distribution.
  - Multifactor experiments
    - Generalized linear model, likelyhood 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
    - Shrunken 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





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)**

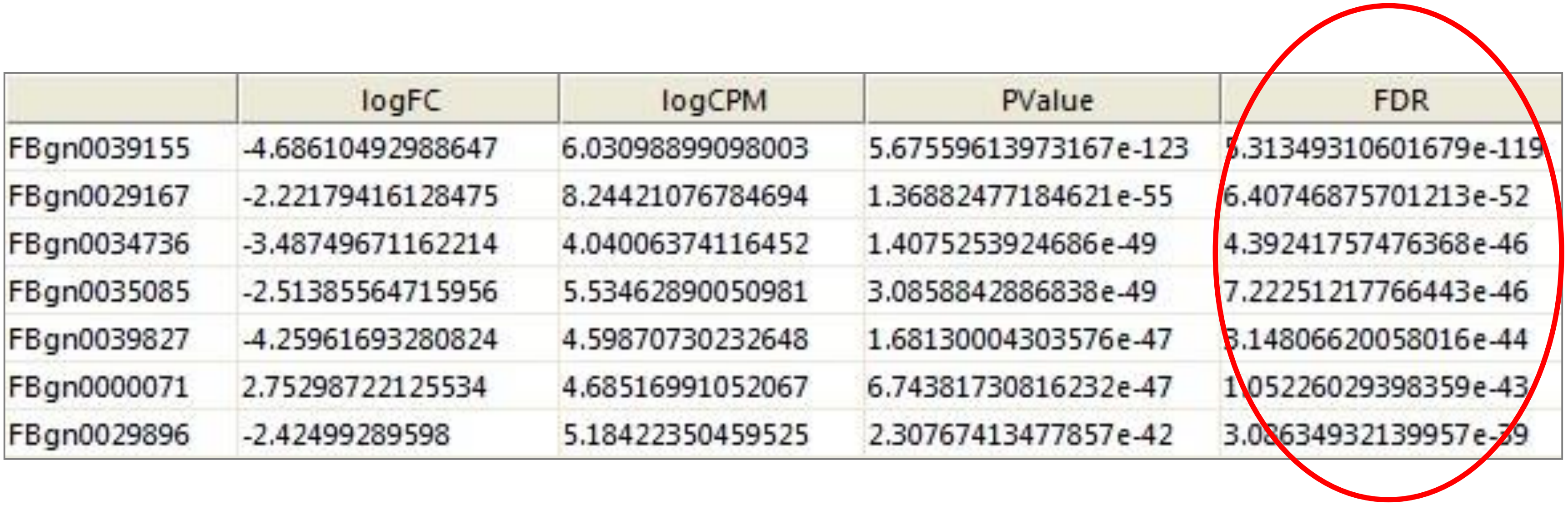
- **Filter out genes which have little chance of showing evidence for significant differential expression**
  - genes which are not expressed
  - genes which are expressed at very low level (low counts are unreliable)

- **Should be independent**
  - do not use information on what group the sample belongs to

- **DESeq2 selects filtering threshold automatically**

edgeR result table

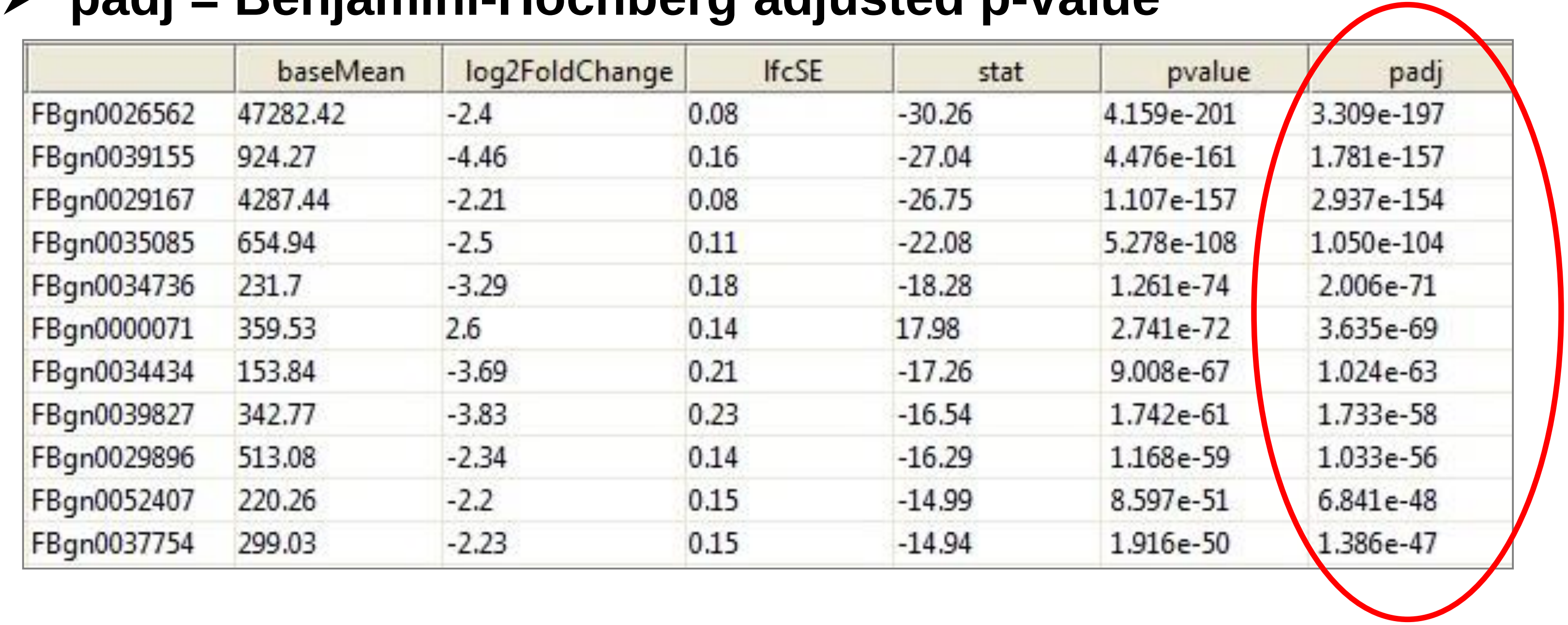
- **logFC = log2 fold change**
- **logCPM = average log2 counts per million**
- **Pvalue = raw p-value**
- **FDR = false discovery rate (Benjamini-Hochberg adjusted p-value)**



	logFC	logCPM	PValue	FDR
FBgn0039155	-4.68610492988647	6.03098899098003	5.67559613973167e-123	5.31349310601679e-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

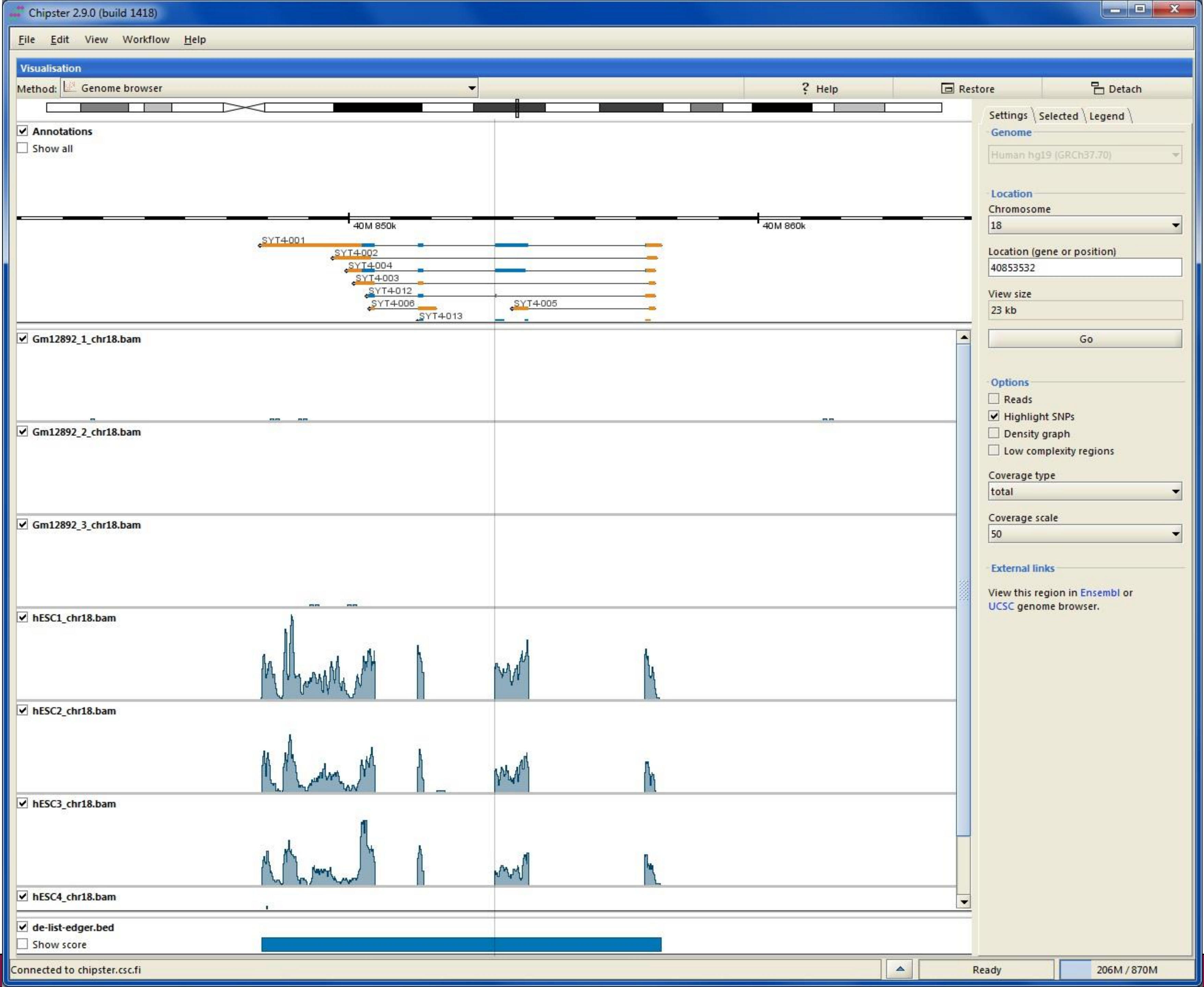
- **Quality control / Read quality with FastQC** → **html report**
- (Preprocessing / Trim reads with Trimmomatic → FASTQ)
- (Utilities / Make a list of file names → txt)
- **Alignment / HISAT2 for paired end reads** → **BAM**
- **Quality control / RNA-seq quality metrics with RseQC** → **pdf**
- **RNA-seq / 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**

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**

Chipster Genome Browser

- **Integrated with Chipster analysis environment**
- **Automatic sorting and indexing of BAM, BED and GTF files**
- **Automatic coverage calculation (total and strand-specific)**
- **Zoom in to nucleotide level**
- **Highlight variants**
- **Jump to locations using BED, GTF, VCF and tsv files**
- **View details of selected BED, GTF and VCF features**
- **Several views (reads, coverage profile, density graph)**



Method: Genome browser

☑ Annotations

□ Show all

☑ Gm12892_1_chr18.bam

☑ Gm12892_2_chr18.bam

☑ Gm12892_3_chr18.bam

☑ hESC1_chr18.bam

☑ hESC2_chr18.bam

☑ hESC3_chr18.bam

☑ hESC4_chr18.bam

☑ de-list-edger.bed

□ Show score

Genome

Human hg19 (GRCh37.70)

Location

Chromosome: 18

Location (gene or position): 40853532

View size: 23 kb

Options

□ Reads

☑ Highlight SNPs

□ Density graph

□ Low complexity regions

Coverage type: total

Coverage scale: 50

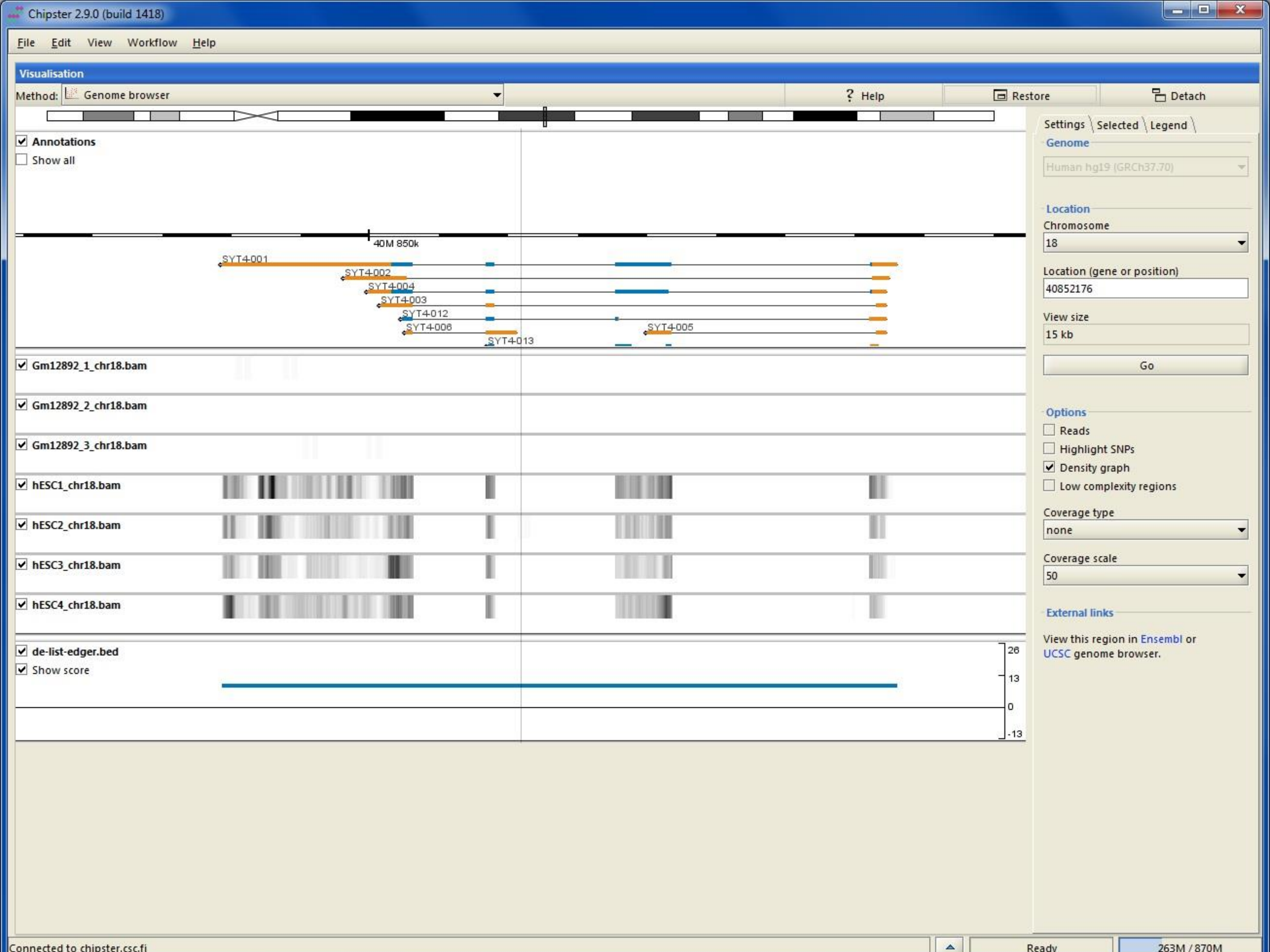
External links

View this region in Ensembl or UCSC genome browser.

Connected to chipster.csc.fi

Ready

206M / 870M



Genome

Human hg19 (GRCh37.70)

Location

Chromosome: 18

Location (gene or position): 40852176

View size: 15 kb

Options

☐ Reads

☐ Highlight SNPs

☑ Density graph

☐ Low complexity regions

Coverage type: none

Coverage scale: 50

External links

View this region in Ensembl or UCSC genome browser.

Connected to chipster.csc.fi

Ready

263M / 870M