

Exome-seq data analysis and visualization with **Chipster**

21.-22.11.2012

Eija Korpelainen, Jarno Tuimala
chipster@csc.fi



Program

- **Introduction to Chipster**
- **Quality control and preprocessing of reads (FastQC, PRINSEQ)**
- **Mapping (= aligning) reads to reference genome (BWA)**
- **Manipulation of alignment files (Samtools)**
- **Visualization of alignments in genome browser**
- **Variant calling (Samtools, GATK)**
- **Variant filtering (VCFtools)**
- **Variant annotation (Bioconductor)**
- **Matching sets of genomic regions (BEDtools)**
- **Saving automatic workflows**



Introduction to Chipster



Goal of Chipster is to enable biologists to

- **analyse and integrate high-throughput data**
- **visualize results efficiently**
- **save and share automatic workflows**



Chipster 2.1.0 (build 1252)

File Edit View Workflow Help

Datasets

- trimmed.fastq
- h1-hESC.bam
- junctions.bed
- h1-hESC.bam.bai
- GM12878.bam
- junctions.bed
- GM12878.bam.bai
- de-genes-cufflinks.tsv
- cufflinks-log.txt
- de-isoforms-cufflinks.tsv
- de-genes-cufflinks.bed
- de-isoforms-cufflinks.bed

Analysis tools

Microarrays NGS

- Quality control
- Filtering
- Utilities
- Matching genomic regions
- Alignment
- RNA-seq
- miRNA-seq
- ChIP-seq
- CNA-seq
- Methyl-seq

Map aligned reads to genes with HTSeq
 Map aligned reads to genes with HTSeq
 Differential expression analysis using DESeq2
 Differential expression analysis using edgeR
 Differential expression analysis using edgeR
 Utilities - Define NGS experiment.

Show parameters Run

More help Show tool sourcecode

Workflow

file → Qual → Util → Alig → RNA- → Util

Visualisation

Method: Genome browser

280k 290k 300k 310k 320k

PPAP2C-202
PPAP2C-203
MIER2-201

GM12878.bam

h1-hESC.bam

Settings Legend

- Reads
- Highlight SNPs
- Density graph
- Low complexity regions

Coverage type: total

Coverage scale: 50

Show all reads

Datasets

- GM12878.bam
- h1-hESC.bam

External links

View this region in [Ensembl](#) or [UCSC genome browser](#).

Notes for dataset

Alignment / TopHat for single end reads

Show

Connected to chipster.csc.fi

Ready 137M / 870M

Analysis functionality, overview

➤ 90 NGS tools for

- ChIP-seq
- RNA-seq
- miRNA-seq
- MeDIP-seq
- CNA-seq
- exome/genome-seq

➤ 140 microarray tools for

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

➤ Tools served in a biologist-friendly manner

- Vocabulary, parameter selection
- "Bigger" tools to avoid unnecessary steps



Chipster NGS functionality – part I

- **Quality control, filtering, trimming**
 - FastX, FastQC, PRINSEQ
- **Mapping (alignment)**
 - Bowtie, BWA, Tophat
- **Processing**
 - Picard, SAMtools
- **Visualization of reads and results in their genomic context**
- **Genomic region matching**
 - BEDTools, HTSeq, In-house tools
- **Exome-seq and genome-seq**
 - Variant calling with Samtools
 - Variant filtering with VCFtools
 - Variant annotation with Bioconductor



NGS data analysis functionality – part II

➤ **ChIP-seq**

- Peak detection and filtering (MACS)
- Motif detection and matching to JASPAR (MotIV, rGADEM)
- Retrieve nearby genes, pathway analysis (GO, ConsensusPathDB)

➤ **miRNA-seq**

- Differential expression (edgeR, DESeq)
- Retrieve target genes (PicTar, miRBase, TargetScan, miRanda,..)
- Pathway analysis (GO, KEGG)

➤ **RNA-seq**

- Differential expression (Cufflinks, edgeR, DESeq, DEXSeq)

➤ **MeDIP-seq**

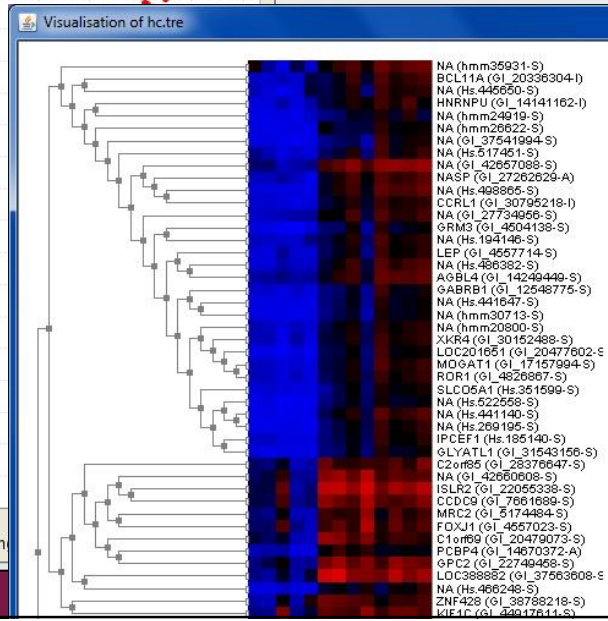
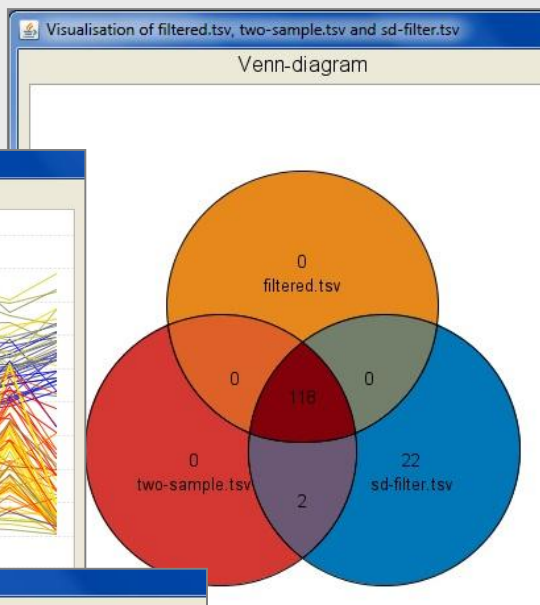
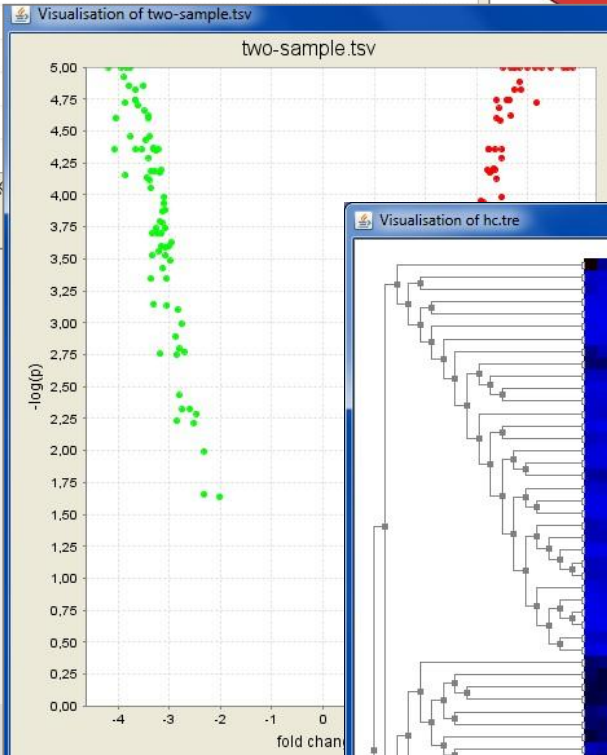
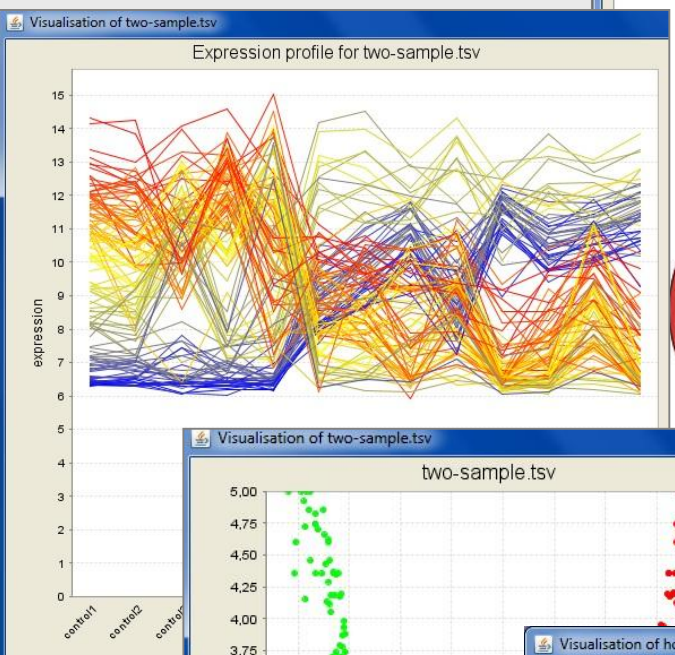
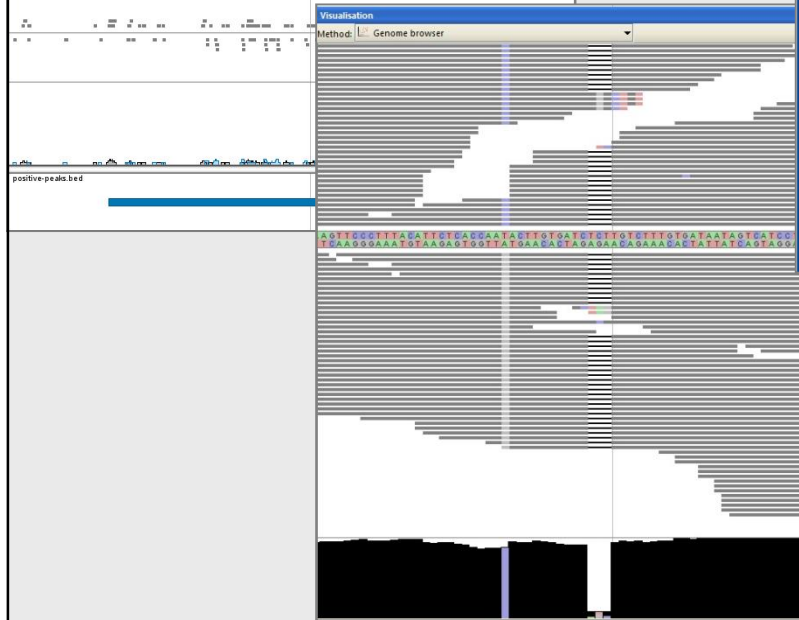
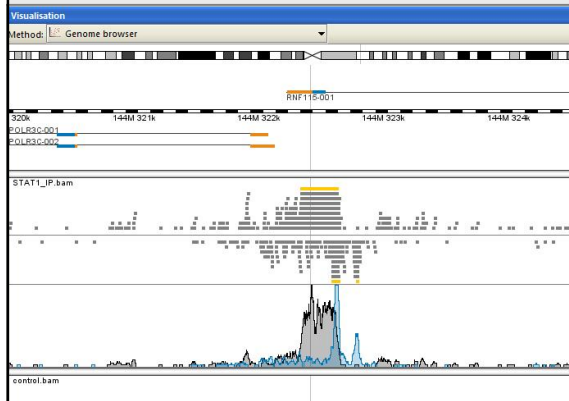
- Methylation analysis, comparison of two conditions (MEDIPS)

➤ **CNA-seq**

- Detection of copy number changes

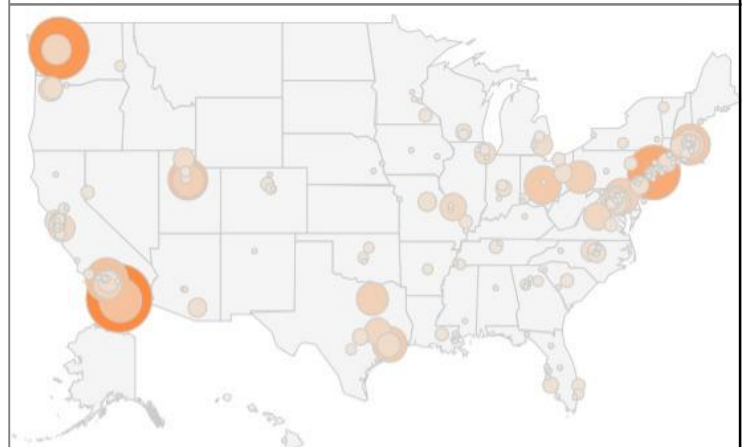
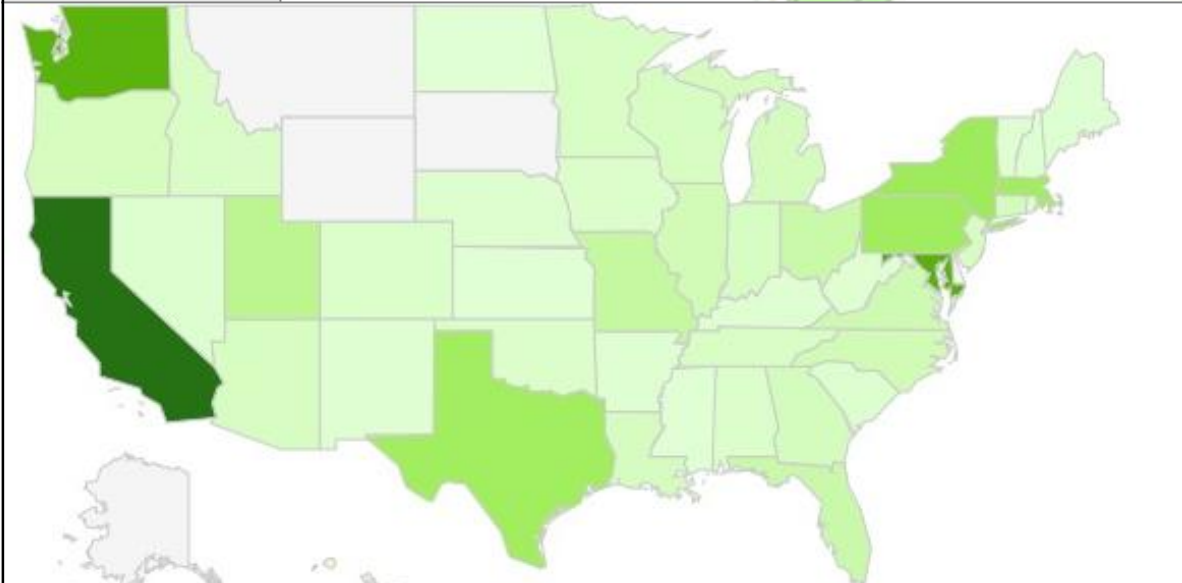


Interactive visualizations



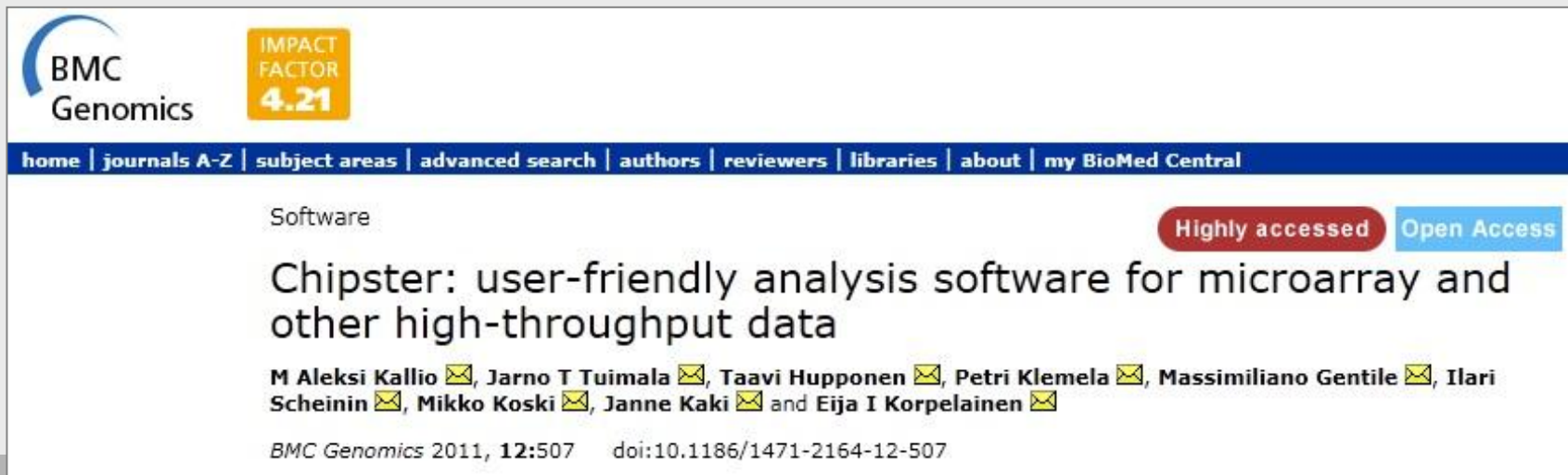
Users on CSC's Chipster server in Finland

- many other Chipster servers around the world



More info

- chipster@csc.fi
- <http://chipster.csc.fi>
- <http://chipster.sourceforge.net/>
- **BMC Genomics 2011, 12:507**



The screenshot shows the BMC Genomics website interface. At the top left is the BMC Genomics logo. To its right is an orange box with the text 'IMPACT FACTOR 4.21'. Below these is a blue navigation bar with links: 'home | journals A-Z | subject areas | advanced search | authors | reviewers | libraries | about | my BioMed Central'. The main content area features the word 'Software' on the left, and 'Highly accessed' (in a red rounded rectangle) and 'Open Access' (in a blue rounded rectangle) on the right. The title of the article is 'Chipster: user-friendly analysis software for microarray and other high-throughput data'. Below the title, the authors are listed: 'M Alekski Kallio', 'Jarno T Tuimala', 'Taavi Hupponen', 'Petri Klemela', 'Massimiliano Gentile', 'Ilari Scheinin', 'Mikko Koski', 'Janne Kaki', and 'Eija I Korpelainen', each followed by a small envelope icon. At the bottom, the publication information is given as 'BMC Genomics 2011, 12:507' and the DOI is 'doi:10.1186/1471-2164-12-507'.

BMC Genomics **IMPACT FACTOR 4.21**

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

Software **Highly accessed** **Open Access**

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

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

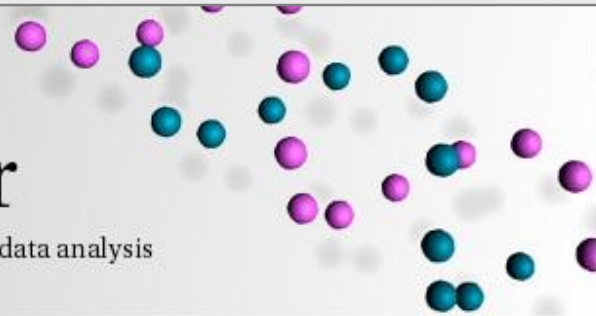
BMC Genomics 2011, **12**:507 doi:10.1186/1471-2164-12-507

Chipster start and info page: chipster.csc.fi



Chipster

Open source platform for data analysis



- Home
- Getting access
- Analysis tool content
- Supported chip types
- Tutorials
- Manual
- Cite
- FAQ
- Screenshots

- Open source project
- Contact

Welcome to Chipster

Chipster is a user-friendly analysis software for high-throughput data. It contains over 200 analysis tools for next generation sequencing (NGS), microarray and proteomics data. Users can save and share automatic analysis workflows, and visualize data interactively using a [built-in genome browser](#) and many other visualizations. Chipster's client software uses Java Web Start to install itself automatically, and it connects to computing servers for the actual analysis. Chipster is available for local server installations at the [Chipster open source site](#). If you would like to use Chipster running on CSC's server, you need a [user account](#). For basic sequence analysis tasks, please use [Embster](#).



Launch Chipster v2.2.0

27.9.2012: New version, read more



Launch Chipster Genome Browser

Visualization only, no user account required (manual).

Chipster mode of operation

- Select data
- Select tool category
- Select tool (set parameters if necessary) and click run
- View results

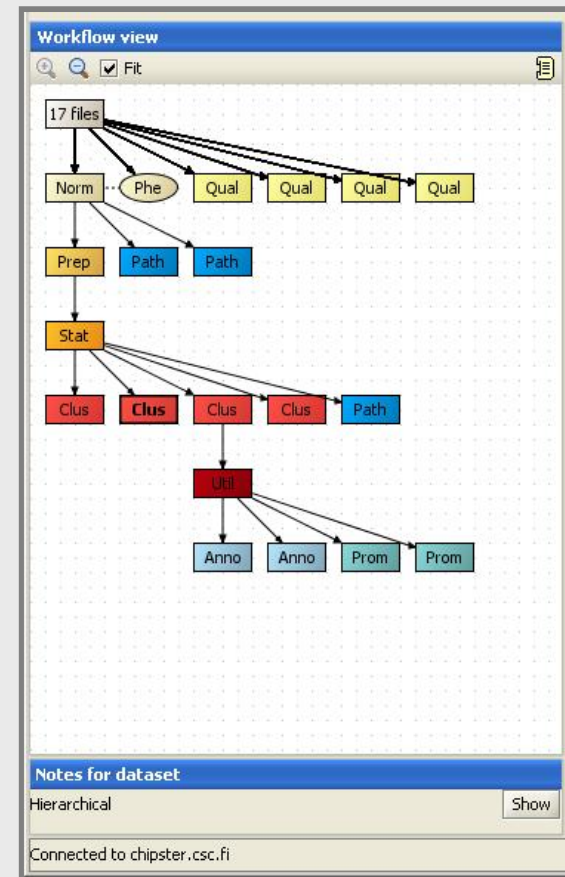
The screenshot displays the Chipster 2.0.0 (build 1135) interface. The top menu includes File, Edit, View, Workflow, and Help. The main workspace is divided into several panels:

- Datasets:** Lists files such as control.bam.bai, treatment.bam, and positive-peaks.tsv.
- Analysis tools:** Shows categories like Microarrays and Next Gen Sequencing. Under Next Gen Sequencing, 'Find peaks using MACS, treatment vs. control' is selected. A red arrow points from this tool to the 'Run' button.
- Workflow:** A drag-and-drop interface where tools are placed in a sequence. A red circle highlights an 'ChIP-seq' tool, with a red arrow pointing from it to the 'Find peaks using MACS, treatment vs. control' tool in the Analysis tools panel.
- Visualisation:** Shows a 'Genome browser' view for the RNF115-001 gene. It displays tracks for POLR3C-001, POLR3C-002, treatment.bam, and positive-peaks.tsv. A red arrow points from the 'ChIP-seq' tool in the workflow to the 'positive-peaks.tsv' track.
- Settings:** Includes 'Location' (Chromosome 1, 144323586) and 'Options' (Reads, Strand-specific coverage, etc.).
- Notes for dataset:** Provides details for the ChIP-seq analysis, including file format (BAM), genome (human), and various cutoff parameters.

At the bottom, the status bar shows 'Connected to chipster.csc.fi', 'Ready', and '64M / 870M'.

Workflow view

- Shows the relationships of the data sets
- Right clicking on the data file allows you to
 - Save an individual result file ("Export")
 - Delete
 - Link to another data file
 - Save workflow
- Zoom in/out or fit to the panel. You can also move the boxes around
- Select several datasets by keeping the Ctrl key down



Automatic tracking of analysis history

The screenshot displays the Chipster v1.0.2 (build 349) interface. The main window is divided into several panes:

- Datasets:** A tree view showing a folder 'My experiment' containing files like GSM11805.cel, GSM11814.cel, GSM11823.cel, GSM11830.cel, normalized.tsv, phenodata.tsv, sd-filter.tsv, two-sample.tsv, seqs.txt.wee, seqs.html, and annotations.html.
- Analysis tools:** A list of tools including Normalisation, Quality control, Preprocessing, Statistics, Clustering, Annotation, Pathways, Promoter Analysis, Visualisation, and Utilities.
- Workflow view:** A flowchart showing the analysis process. It starts with '4 files' leading to 'Norm', 'Phe', and 'Qual' steps. 'Norm' leads to 'Prep', which then leads to 'Stat', 'Path', and 'Prom' steps. 'Stat' leads to 'Prom' and 'Anno' steps. A red box highlights the 'Norm', 'Prep', and 'Stat' steps, and a red circle highlights the 'History' icon in the toolbar.
- History:** A dialog box showing the analysis history for two steps. Step 3 details the 'Two groups tests' operation on 'two-sample.tsv'. Step 4 details the 'Weeder' operation on 'seqs.txt.wee'.

The History dialog box contains the following information:

Show for Datasets:

- Step title
- Dataset Name
- Creation Date
- Applied Operation
- Parameters
- Operation Source Code
- User Notes

Step 3

Dataset name: two-sample.tsv
Created with operation: Two groups tests
Parameter column: group
Parameter test: t-test
Parameter p.value.adjustment.method: none
Parameter p.value.threshold: 0.05

Step 4

Dataset name: seqs.txt.wee
Created with operation: Weeder
Parameter species: human
Parameter promoter.size: short

Buttons: Save... Close

Analysis sessions

- In order to continue your work later on, you have to save the analysis session.
- Saving the session will save all the datasets and their relationships. The session is packed into a single .zip file and saved on your computer.
- Session files allow you to continue the work later, on another computer, or share it with a colleague.
- You can have multiple analysis sessions saved separately, and combine them later if needed.



Workflow – reusing and sharing your analysis pipeline

- **Chipster allows you to save your analysis workflow as a reusable automatic "macro", which can be applied to another dataset**
- **All the analysis steps and their parameters are saved as a script file, which you can share with other users**

Saving and using workflows

The screenshot shows the Chipster v1.4.2 (build 831) interface. The 'Workflow' menu is open, showing options: 'Run...', 'Run recent', 'Run from Chipster repository', and 'Save starting from selected...'. The 'Analysis tools' panel on the right lists: 'Normalisation', 'Gene expression analysis', 'Protein expression analysis', 'miRNA expression analysis', 'More information...', 'Promoter Analysis', 'Visualisation', 'Utilities', and 'aCGH tools (beta testing)'. The 'Workflow' panel at the bottom shows a workflow diagram with 13 files at the start, followed by 'Norm' and 'Phe' steps. A red box highlights a sequence of steps: 'Prep', 'Qual', 'Stat', 'Stat', 'Visu', 'Qual', 'Visu', 'Stat', 'Clus', 'Clus', 'Anno', 'Path', 'Clus', 'Clus', 'Util', 'Util', and 'Prom', 'Prom'.

- **Select the starting point for your workflow and click "Workflow/ Save starting from selected"**
- **You can save the workflow file anywhere on your computer and change its name, but the ending must be .bsh.**
- **To run a workflow select**
 - Workflow->Open and run
 - Workflow->Run recent (if you saved the workflow recently).

Importing data to Chipster



Different ways of importing data to Chipster

➤ Import a file

- **Files / Import files** (note that you can select several files by keeping the Ctrl key down)
- FASTQ, BAM, BED, VCF and GTF files are recognized automatically
- SAM/BAM and BED files can be preprocessed at the import stage (sort and index BAM, sort BED)

➤ Import a folder

- **Files / Import folder**

➤ If you want to continue an existing analysis session

- **Files / Open session**



Exercise 1: Start Chipster and import data

➤ Launch Chipster client program

- Go to <http://chipster.csc.fi>
- Log in
 - username: **oulu2012exome**
 - password: **training**

➤ Import data

- Click **Import files** and select file **h1-hESC_RNAseq.fastq**
- When the file appears in the workflow view, double-click on it to see what it looks like. Maximize the visualization panel.



Raw data: FASTQ file format (.fastq / .fq / .txt)

➤ Four lines per read:

- Line 1 begins with a '@' character and is followed by a sequence identifier.
- Line 2 is the sequence.
- Line 3 begins with a '+' character and can be followed by the sequence identifier.
- Line 4 encodes the quality values for the sequence, encoded with a single ASCII character for brevity.
- Example:

@SEQ_ID

GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT

+

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

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



Base qualities

- **If the quality of a base is 30, the probability that it is wrong is 0.001. So if you have 1000 bases of quality 30, one of them is wrong on average.**
 - 40 -> 1 in 10 000 is wrong
 - 20 -> 1 in 100 is wrong
 - Phred quality score $Q = -10 * \log_{10}$ (probability that the base is wrong)
- **Encoded as ASCII characters so that 33 is added to the Phred score**
 - This "Sanger" encoding is used by Illumina 1.8+, 454 and SOLiD
 - 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



Quality control



What and why?

➤ **Potential problems**

- low-quality sequences
- sequence artifacts
- sequence contamination
- Examples: low confidence bases, Ns, duplicate reads, location bias, adapters, another organism...

Knowing about potential problems in your data allows you to

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



Software packages for quality control

- **FastQC (available in Chipster)**
- **FastX (available in Chipster)**
- **PRINSEQ (available in Chipster, excellent for 454 data)**
 - Memory issues: use a subset of FASTQ file for this tool
- **HTSeq QC scripts**
- **SolexaQA**
- **TagDust**



Quality control measurements

➤ **Quality plots**

- Per base
- Per read

➤ **Composition plots**

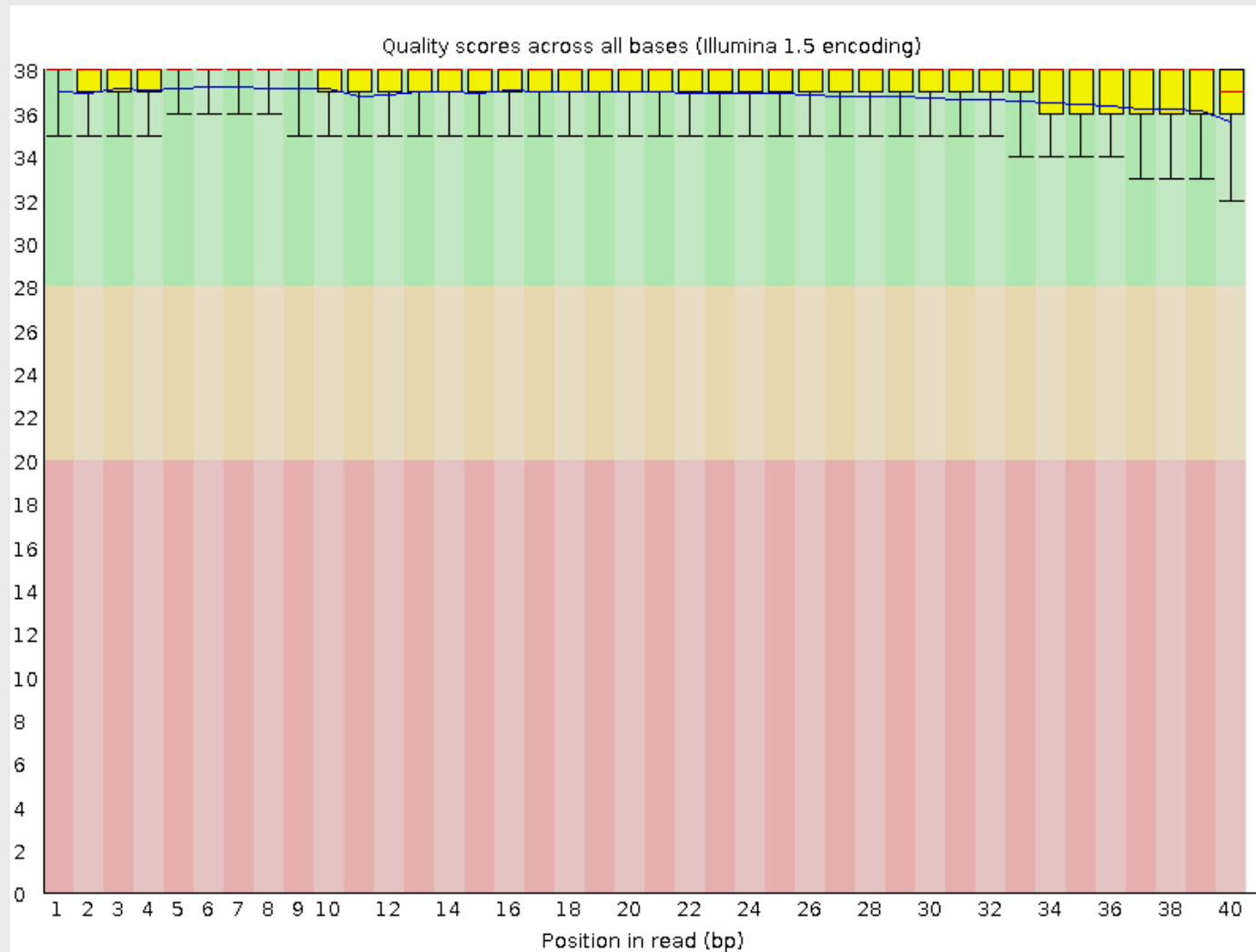
- Per base composition
- GC content and profile

➤ **Contaminant identification**

- Overrepresented sequences and k-mers
- Duplicate levels



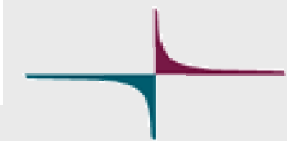
Per base sequence quality plot (FastQC)



good

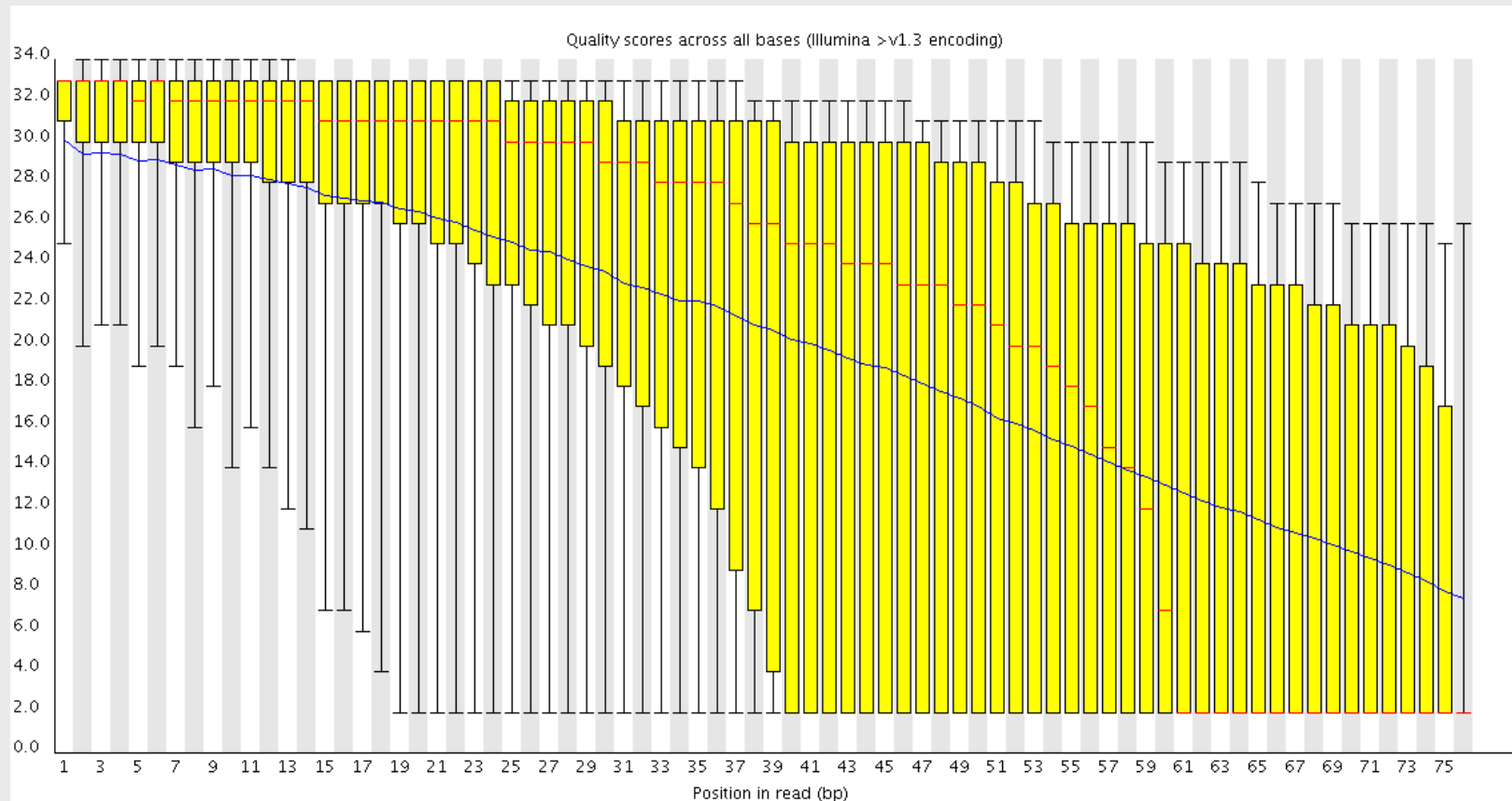
so and so

bad



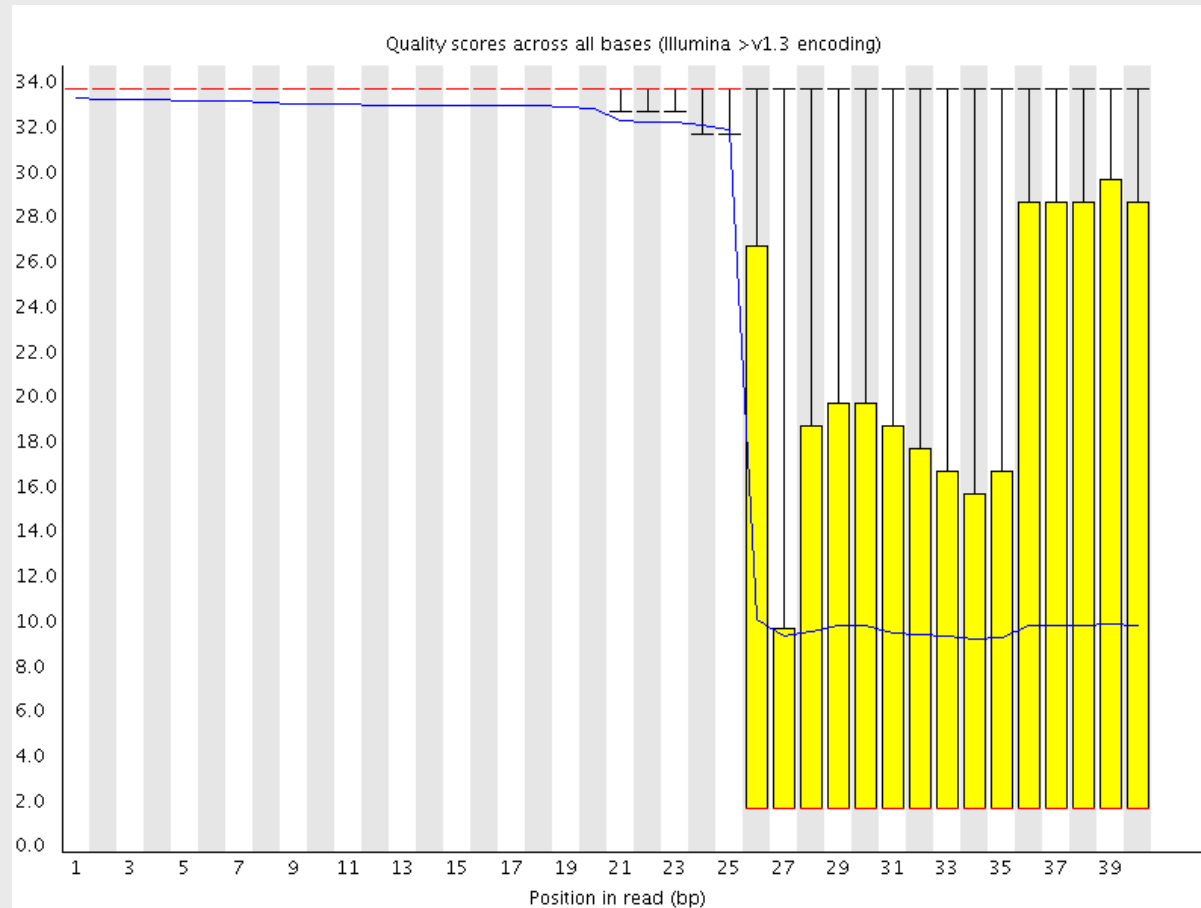
CSC

Quality drops gradually



➤ **Typical for longer runs -> filter / trim the low quality ends.**

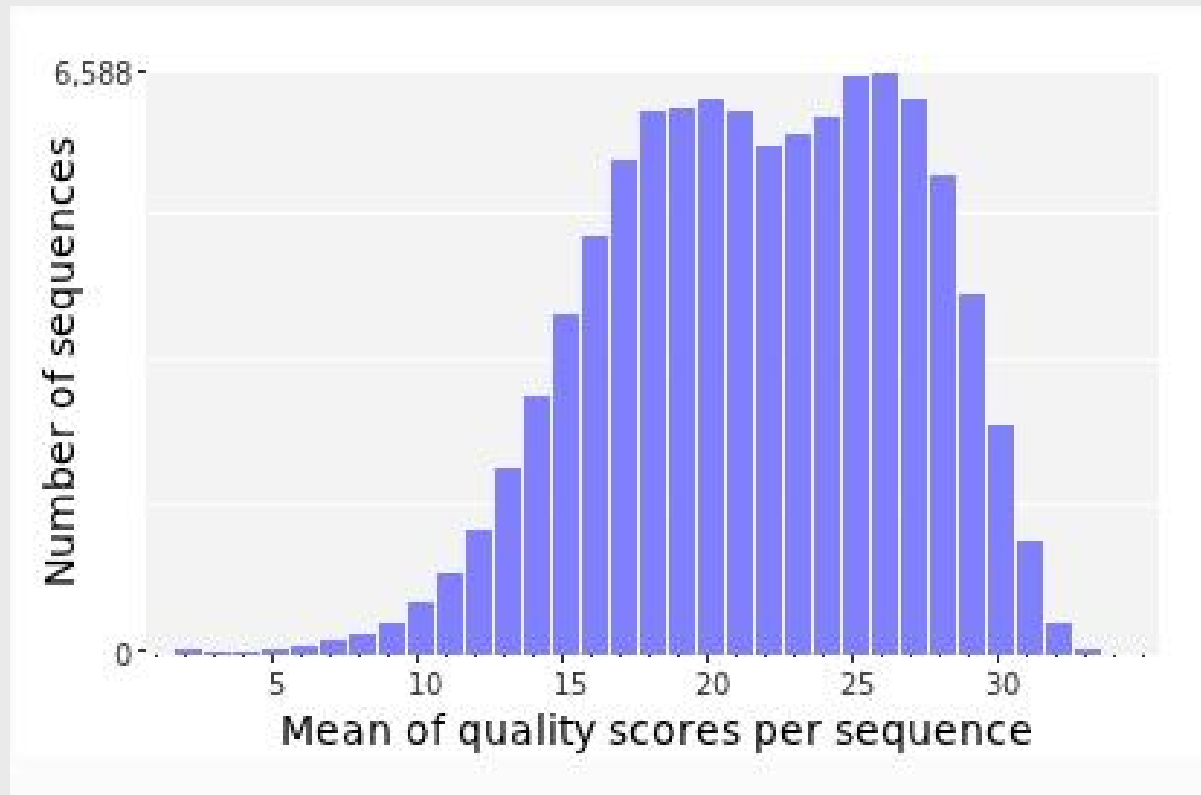
Quality drops suddenly



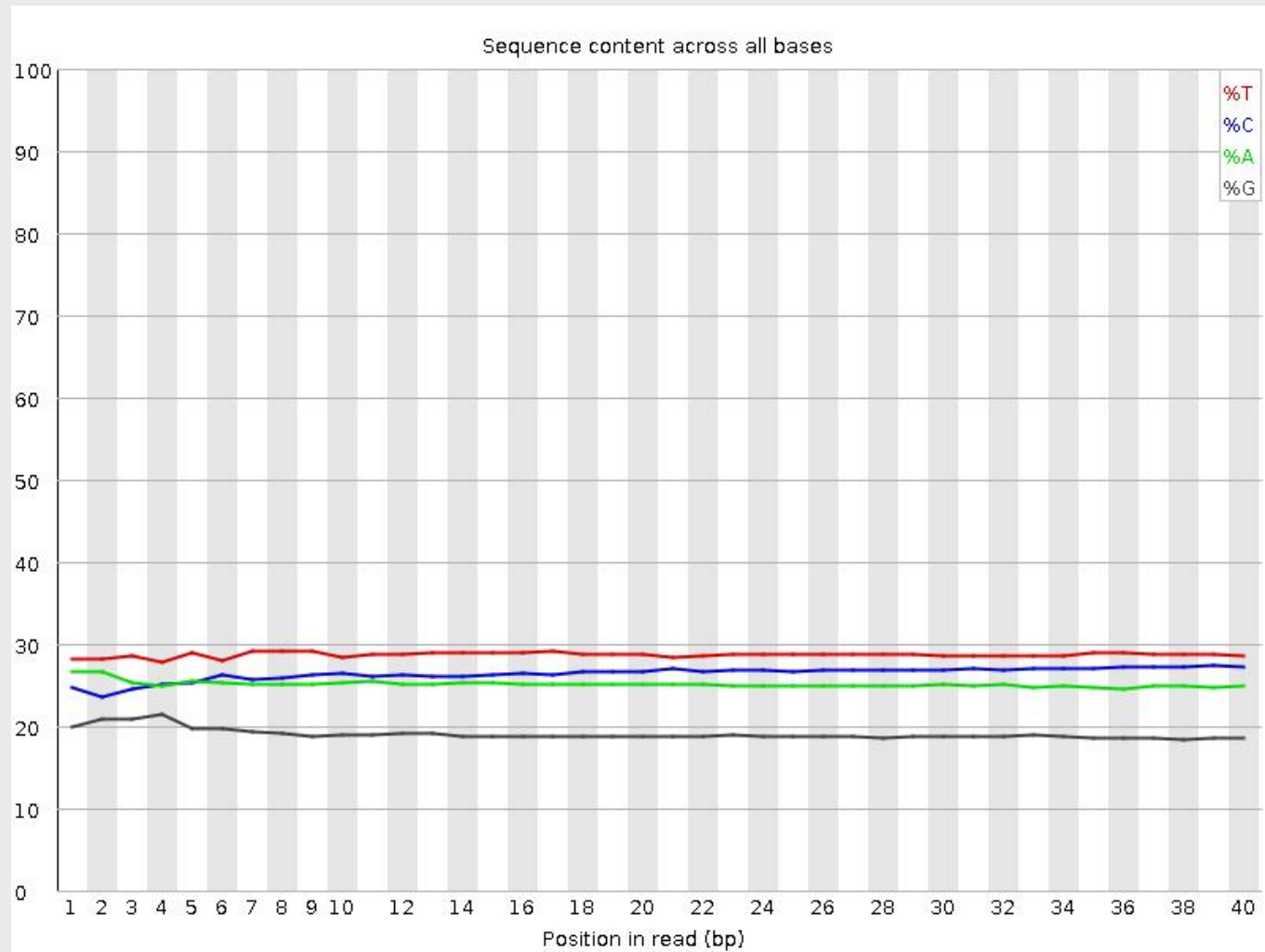
➤ **Problem in the flow cell -> trim the sequences**



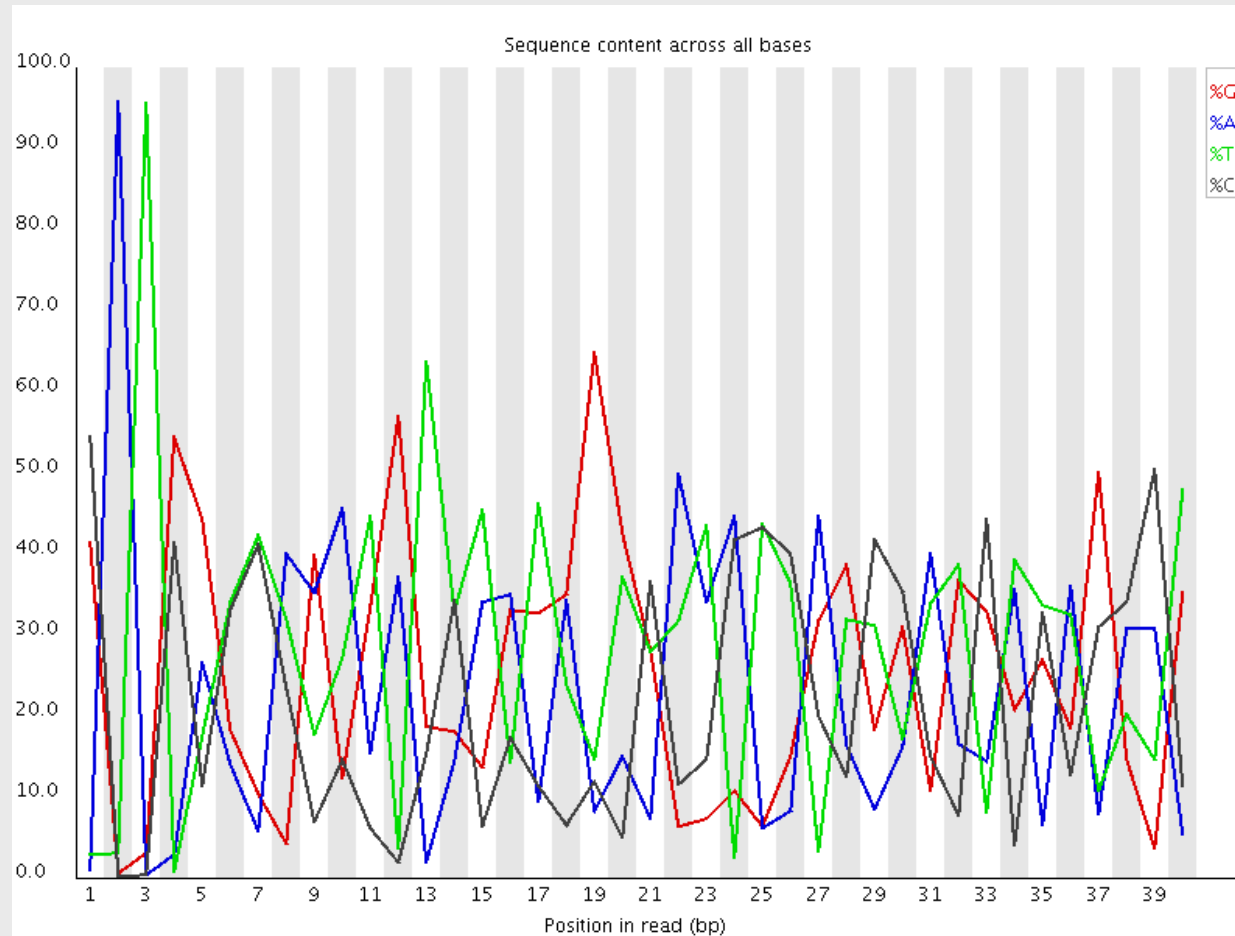
Read mean quality scores (PRINSEQ)



Per base sequence content (FastQC)



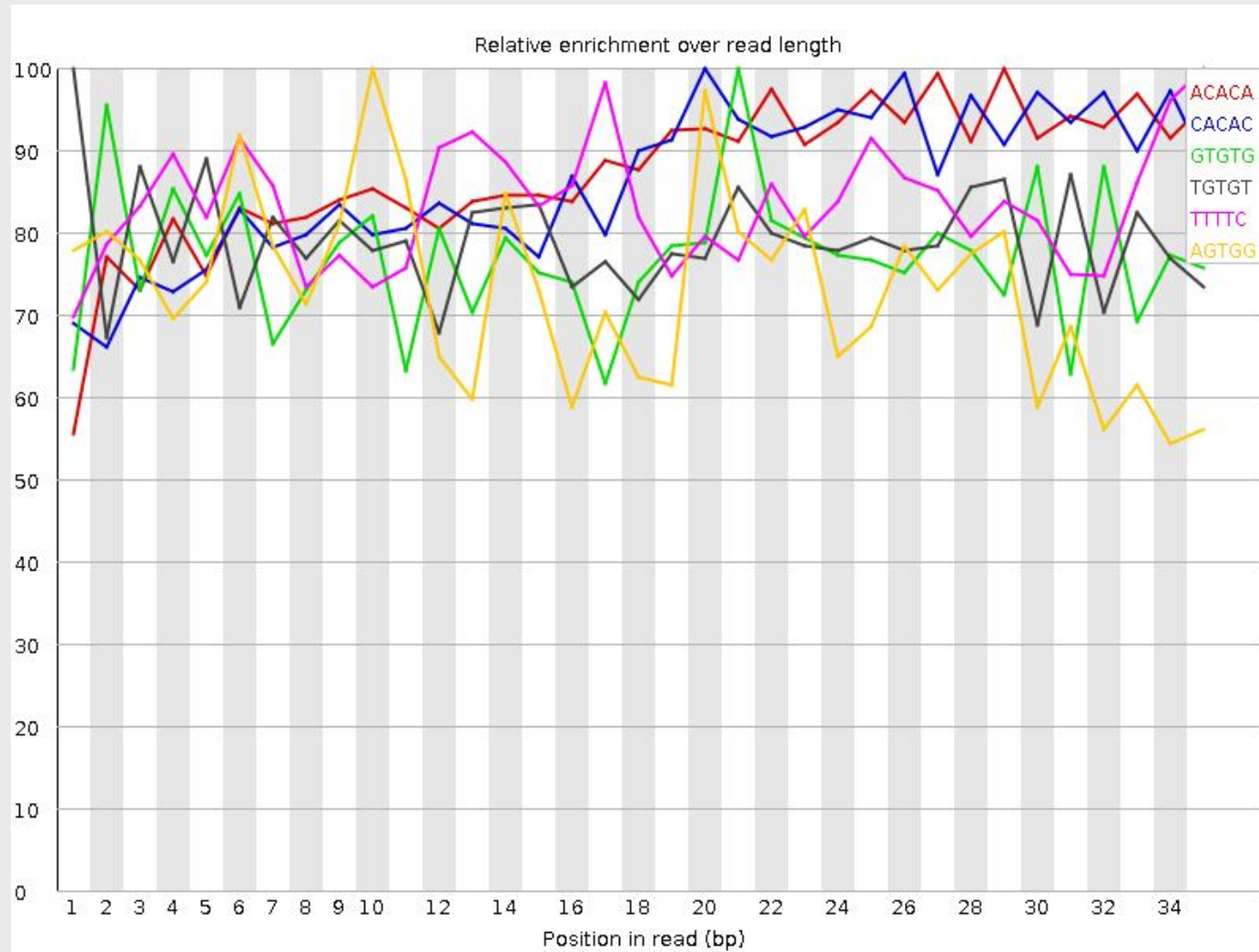
Biased sequence



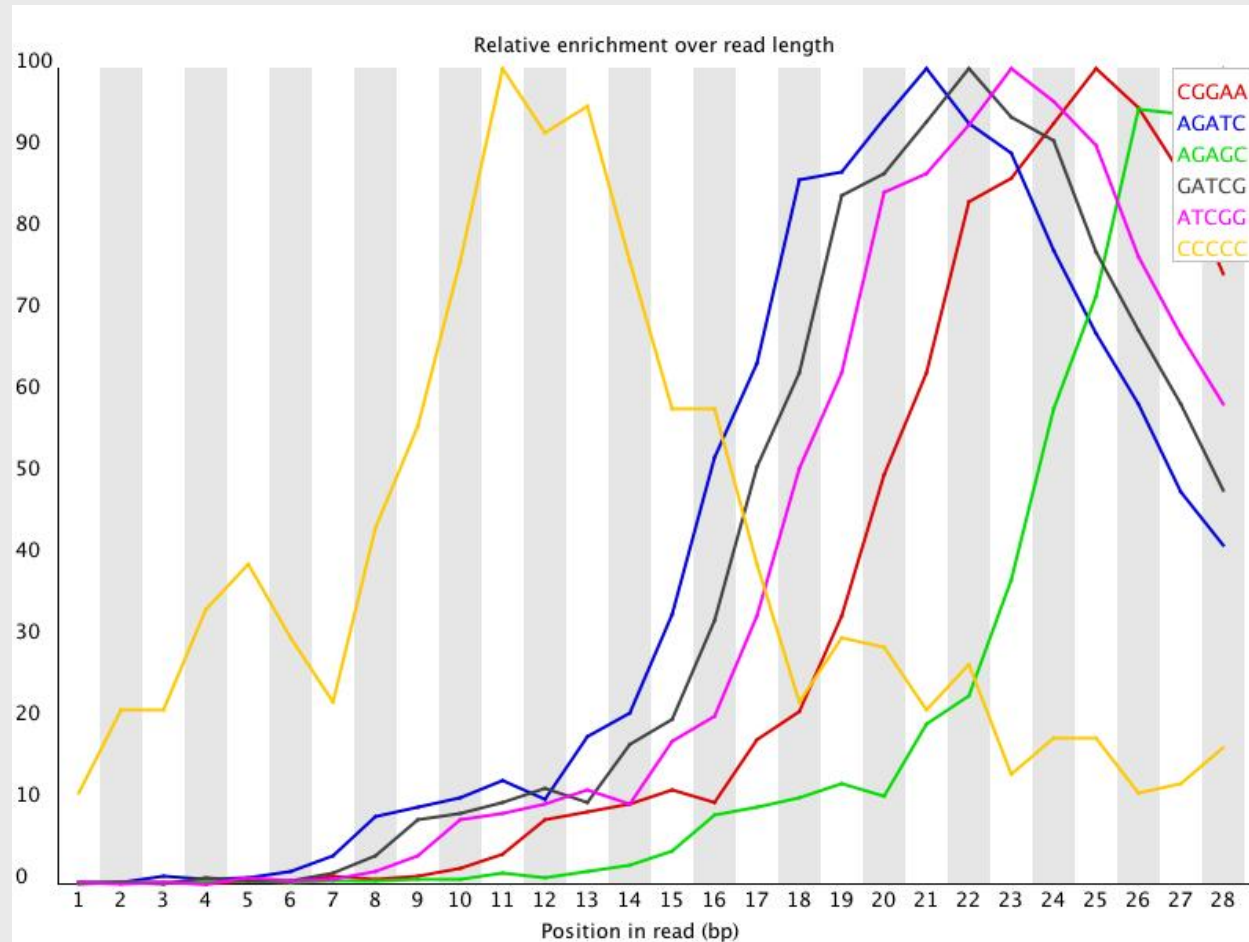
- **Library has a restriction site at the front**
- **A single sequence makes up of 20% of the library**



K-mer profile plot (FastQC)



K-mer enrichment rises towards the end



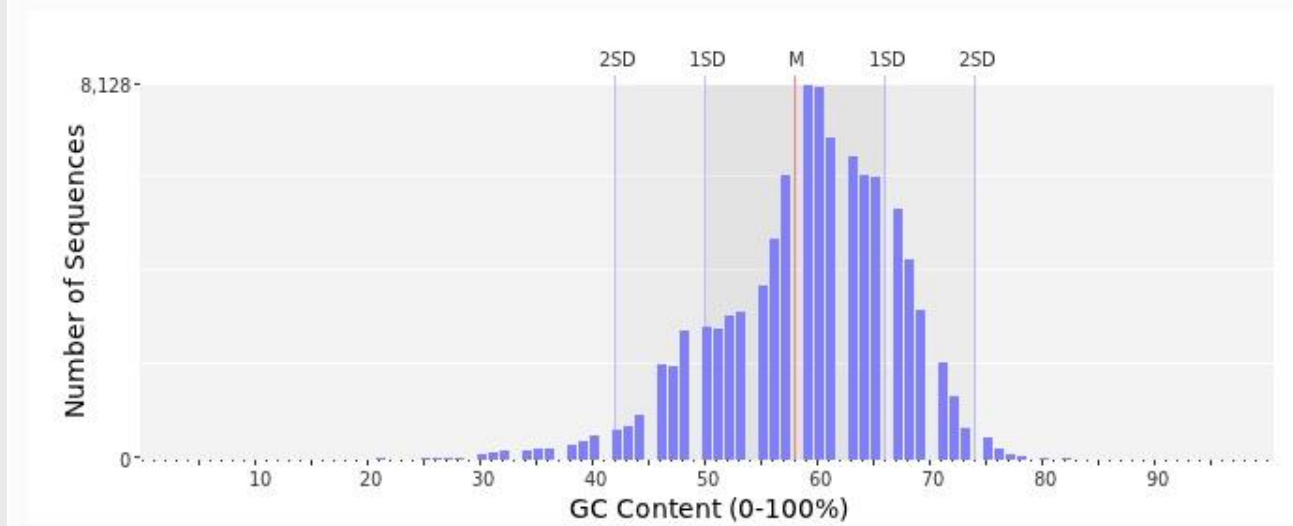
➤ Reads contain partial Illumina adapter sequences -> trim



GC content (PRINSEQ)

GC CONTENT DISTRIBUTION

	Value
Min	10
Max	90
Range (Max - Min)	81
Mean (Average)	58.88
Standard deviation	8.06
Mode (x-axis value)	59
Mode value (y-axis value)	8,128



Percentage of Ns per read (PRINSEQ)

OCCURENCE OF N

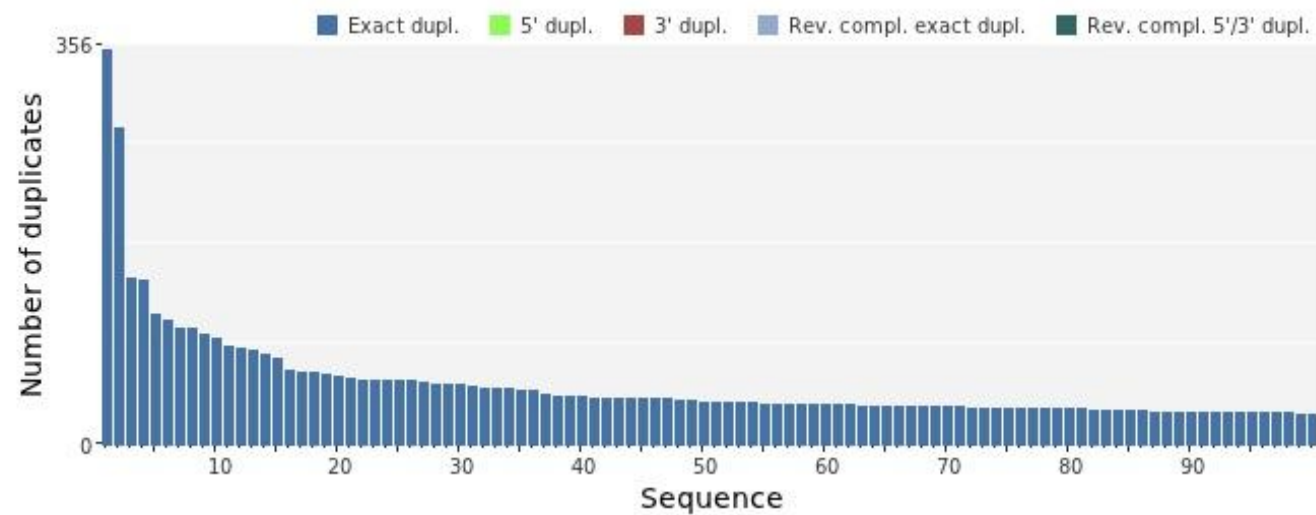
	Value
# Sequences with Ns	5895 (5.95%)
Max percentage of Ns per sequence	5



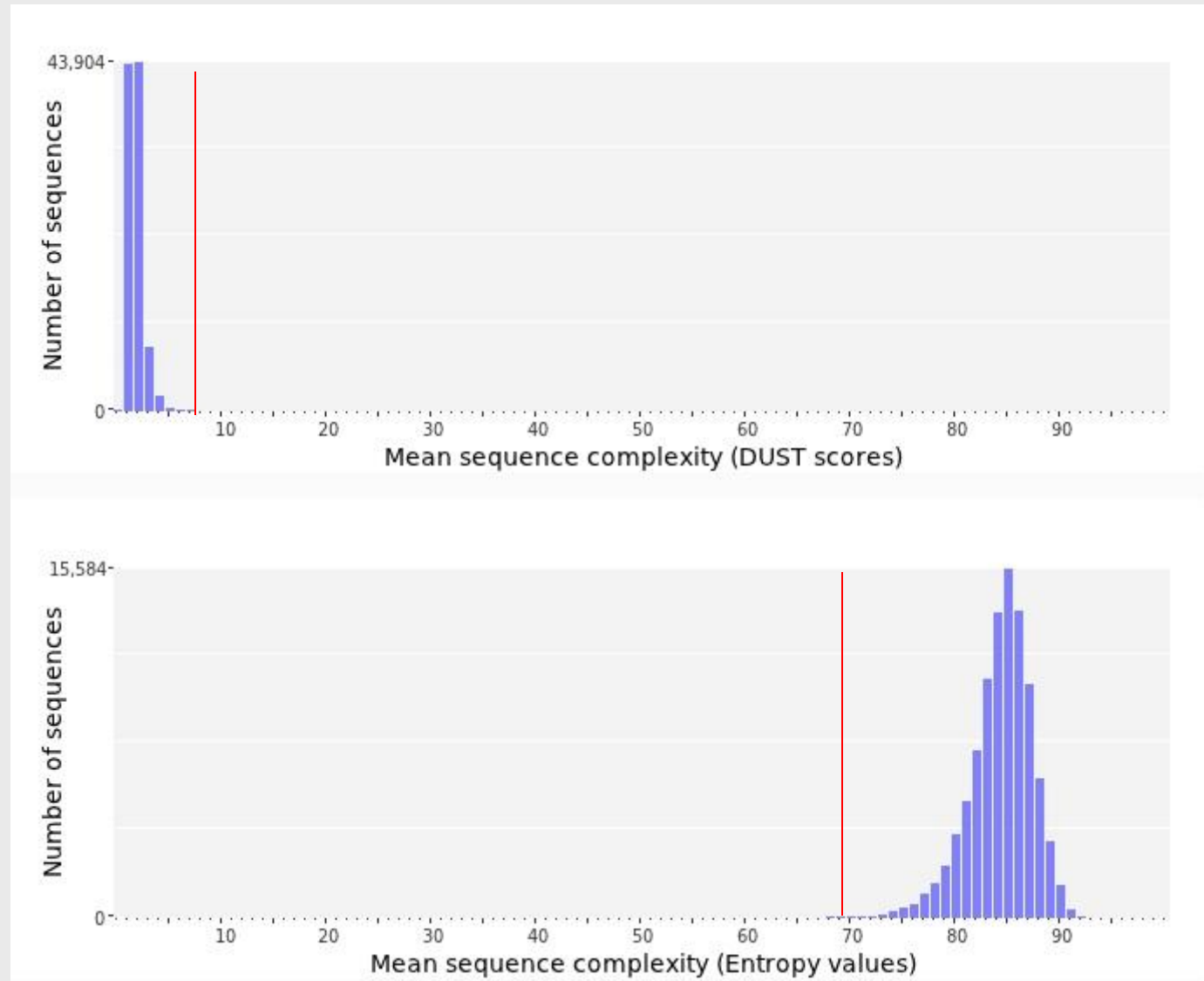
Number of duplicates per read (PRINSEQ)

SEQUENCE DUPLICATION

	# Sequences	Max duplicates
Exact duplicates	29478 (29.74%)	353
5' duplicates	0	0
3' duplicates	0	0
Exact duplicates with reverse complements	2 (0.00%)	1
5'/3' duplicates with reverse complements	0	0
Total	29480 (29.74%)	-



Low complexity plots (PRINSEQ)



Exercise 2: Quality control plots

- Select the file **h1-hESC_RNAseq.fastq**
- Select the tool category **Quality control**
- Select the tool **Read quality with FastQC** and click run.
 - How many reads are there and how long are they (fastqc_data.txt) ?
 - Up to what length is the quality acceptable (per_base_quality.png)?
What quality encoding is used?
- Analyze the same fastq file with **Read quality statistics with PRINSEQ**
 - Select the file **read-stats.html** and the visualization method **Open in external web browser**.
 - Inspect the report. What is the percentage of reads containing Ns and duplicates? Do you need to filter for low complexity reads?



Filtering and trimming low quality reads



Filter and trim low quality reads: FastX

➤ **Filter sequences based on quality**

- Decide what is the minimum quality value allowed (e.g. 20)
- Decide what percentage of bases in a read should have at least this quality

➤ **Trim a certain number of bases from all reads**

- Decide how many bases should be trimmed and from which end

➤ **Filter reads for adapters, ambiguous nucleotides (N) and length**

- Give the adapter sequence and decide what is the minimum allowed sequence length after clipping



Filter low quality reads: PRINSEQ

- **Filter sequences based on quality scores**
 - Min/ max quality score per base, mean of quality scores
- **Filter for low complexity**
 - DUST (score 1-100, > 7 means low complexity)
 - Entropy (score 1-100, < 70 means low complexity)
- **Filter for Ns**
 - Maximum count/ percentage of Ns that a read is allowed to have
- **Filter for length**
 - Min/ max length of a read
- **Filter for duplicates**
 - Exact, reverse complement, or 5'/3' duplicates
- **Filter for several criteria**
 - All above, and possibility to get filtered pairs for paired end data

Trim low quality reads: PRINSEQ

- **Trim based on quality scores**
 - Min, mean
 - In a sliding window
 - From 3' or 5' end
- **Trim polyA/T tails**
 - Minimum number of A/Ts
 - From left or right
- **Trim based on several criteria**
 - All above
 - Trim x bases from left/ right
 - Trim to length x



Exercise 3: Filter and trim reads (FastX)

➤ Filter based on base quality

- Select the file **h1-hESC_RNAseq.fastq** and the tool **Filtering / Filter reads for quality**, and set the quality cut-off value to **20**
- How many reads were filtered out?
- Check if the base quality now looks acceptable using the tool **Read quality with FastQC** on the filtered data (file **quality_filtered.fq**)

➤ Trim off the ends of all reads

- Select the file **h1-hESC_RNAseq.fastq** and the tool **Utilities / Trim reads**, and set the last base to keep to **50**
- Run the tool **Read quality with FastQC** on the trimmed data

Which approach would you use to get rid of low quality sequence: trimming all reads or filtering based on qualities? Why?



Exercise 4: Filter and trim reads (PRINSEQ)

➤ Trim based on quality

- Select the file **h1-hESC_RNAseq.fastq** and the tool **Utilities / Trim reads by quality**, and set the parameter **Trim 3' end by quality** to **20**
- How many reads were filtered out? How does the fastq file look like?

➤ Remove duplicates, Ns and too short reads

- Select the file **trimmed.fastq** and run the tool **Filter reads by several criteria** using the following parameters:
 - **Minimum length = 50**
 - **Maximum count of Ns = 0**
 - **Types of duplicates to filter = exact duplicates**
 - **Number of allowed duplicates = 2**
- How many reads were kept?



Map (=align) reads to reference genome



Why?

- **Most NGS applications (apart from de novo assembly) require location information = mapping the reads to a reference**
 - RNA-seq
 - Re-sequencing, variant detection
 - ChIP-seq
 - Assembly by mapping
 - Methyl-seq
 - CNA-seq
- **Mapping can seriously affect the analysis results**



Alignment (= mapping) programs

➤ **Bowtie**

- Bowtie1 is fast but cannot handle indels
- Bowtie2 can handle indels

➤ **BWA**

- Can handle indels, for variant detection

➤ **TopHat**

- For RNA-seq data, can handle spliced reads



BWA

- **Performs gapped alignments = able to detect indels**
- **Two alignment algorithms available:**
 - **BWA** for short (< 200 bp) good quality reads (error rate <3%)
 - **BWA-SW** for longer reads (for single-end reads only)
- **Chipster has separate tools for single-end and paired end-data**
- **Reference genome indexes currently for human, mouse and rat**
 - Let us know if other reference genomes need to be added
- **You can use your own reference too, but indexing may take some hours**
 - Supply reference genome in **FASTA** format



BWA cycles and parameters

- **First alignment cycle: match seed regions of the reads**
 - seed length (starts from the beginning of the read)
 - max number of seed region differences
- **Second alignment cycle: gapped alignment**
 - mismatch, gap opening, gap extension, total number of gaps
 - quality based trimming
 - disallowed regions for gaps
- **Parameters controlling which alignments are reported**
 - maximum edit distance
 - maximum number of gaps and gap extensions
 - mismatch penalty threshold
 - max. number of alignments for a read



File format for mapped reads: BAM/SAM

- SAM (Sequence Alignment/Map) is a tab-delimited text file containing read alignment data. BAM is a binary form of SAM.
- Optional header section
- Alignment section has one line with 11 mandatory fields for each read:
 - read name, flag, reference name, position, mapping quality, CIGAR, mate name, mate position, fragment length, sequence, base qualities
- CIGAR reports match (M), insertion (I), deletion (D), intron (N), etc
- 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
```



BAM file format (.bam) and index (.bai)

- **BAM files can be indexed by genomic position to efficiently retrieve reads for a given region. The index file must have the same name.**
- **Genome browser requires both BAM and the index file.**
- **Chipster allows you to view BAM files with BAM viewer.**
- **The alignment tools of Chipster automatically produce sorted and indexed BAMs.**
- **When you import BAM files, Chipster asks if you would like to preprocess them (convert SAM to BAM, sort and index BAM).**



Exercise 5: Map reads to genome

- Select file **accepted.fastq** and **Alignment / BWA for single end reads**. Set **genome** to **human (hg19)**
 - What does the BAM file look like?
- Select the BAM file and run **Utilities / Count alignments in BAM**
 - How many alignments are there?
 - How many alignments have a mapping quality higher than 20?
- What happens to the alignment number when you run BWA and
 - set the **maximum number of differences in the seed region** to **0**



Visualization



Why?

Nothing beats the human eye in detecting potentially interesting patterns in the data



Software packages for visualization

- **Chipster genome browser**
- **IGV**
- **UCSC genome browser**
- **Tablet**
- **....**

- **Differences in memory consumption, interactivity, ability to edit, annotations, contig view,...**



Chipster Genome Browser

- **Integrated with Chipster analysis environment**
- **Automatic sorting and indexing of BAM and BED**
- **Automatic coverage calculation**
- **Zoom in to nucleotide level**
- **Highlight SNPs**
- **View spliced reads**
- **Jump to locations using BED and VCF files**
- **Several views (reads, coverage profile, density graph)**
- **Memory-efficient**



Exercise 6: View reads in genomic context

- Select **bwa.bam** and **bwa.bam.bai**
- In the visualization panel, select **genome browser** and click on the **maximize** button
 - Select genome **hg19**, set the coverage scale to **250**, type gene **CNN2** in the location field and click **go**.
 - Zoom in to the nucleotide level. Can you see any SNPs?



Manipulating BAM files



Manipulating BAM files (SAMtools, Picard)

- **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.
- **Index BAM**
- **Statistics for BAM**
 - How many reads align to the different chromosomes.
- **Count alignments in BAM**
 - How many alignments does the BAM contain.
 - Includes an optional mapping quality filter.
- **Retrieve alignments for a given chromosome/region**
 - Makes a subset of BAM, e.g. chr1:100-1000, inc quality filter.
- **Create consensus sequence from BAM**



Exercise 7: Retrieve reads which map to chr19

- **How many reads map to chromosome 19?**
 - Select **bwa.bam** and **bwa.bam.bai** and the tool **Statistics for BAM**, and check that the files are correctly assigned.
- **Retrieve alignments to chromosome 19**
 - Select **bwa.bam** and **bwa.bam.bai** and the tool **Make a subset of BAM**



Matching sets of genomic regions



Why?

Useful for many questions, for example:

- **Do my mapped reads match to positions of known genes?**
- **Does my SNP list contain known SNPs?**
- **Give me only the reads which do not match known genes / SNPs**
- **What genes are closest upstream to my ChIP-seq peaks?**
- **Do the peaks overlap with transcription start sites?**
- **....**



Software packages for region matching

➤ **BEDTools**

- Supports BED, GTF, VCF, BAM
- Rich functionality

➤ **HTSeq**

- Supports GTF
- Good gene models for mapping reads to genes

➤ **Chipster's own region matching tools**

- Support BED
- Tolerant for chromosome naming (chr1 vs 1)



Region file formats: BED

- **5 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
chr19	325637	326573	JUNC00000014	55



Region file formats: GFF/GTF

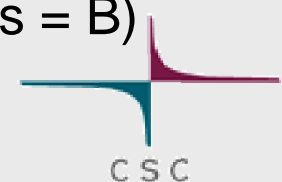
- **9 obligatory columns: chr, source, name, start, end, score, strand, frame, attribute**
- **1-based, like VCF**

```
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";
```



Intersect BED (BEDTools)

- **Looks for overlapping regions between two BED/GFF/VCF files**
 - One of the files can also be BAM
 - Option for strand-awareness
- **Reporting options**
 - Only the overlapping region
 - Original region in file A or B
 - Region in A so that the overlapping part is removed
 - Remove the portion of a region that is overlapped by another region
- **The B file is loaded to memory**
 - So the smaller one should be B (e.g. BAM = A, exons = B)



Closest BED (BEDTools)

- Looks for overlapping regions between two BED/GFF/VCF files and if no overlap is found, the closest region is reported.
- Reports a region in A followed by its closest region in B.
- Option for strand-awareness
- E.g. What is the nearest gene to this SNP?



Window BED (BEDTools)

- **Looks for overlapping regions between two BED/GFF/VCF files after adding a given number of bases upstream and downstream of regions in A.**
 - One of the files can be BAM
- **Reports the regions in A which overlap with regions in B.**
- **Option for strand-awareness**

