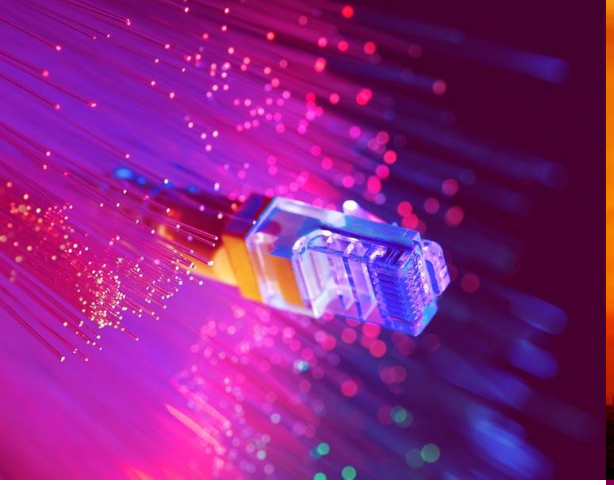


Single-cell RNA-seq data analysis using Chipster

9.2.2018

chipster@csc.fi

CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

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 → less money wasted
- Allows you to discuss more easily with bioinformaticians

What will I learn?

- How to operate the Chipster software
- Analysis of single cell RNA-seq data
  - Central concepts
  - Analysis steps
  - File formats
- Exercises
  - DropSeq data preprocessing (from raw reads to expression values)
  - Clustering analysis of 10X Genomics data with Seurat tools

Introduction to Chipster

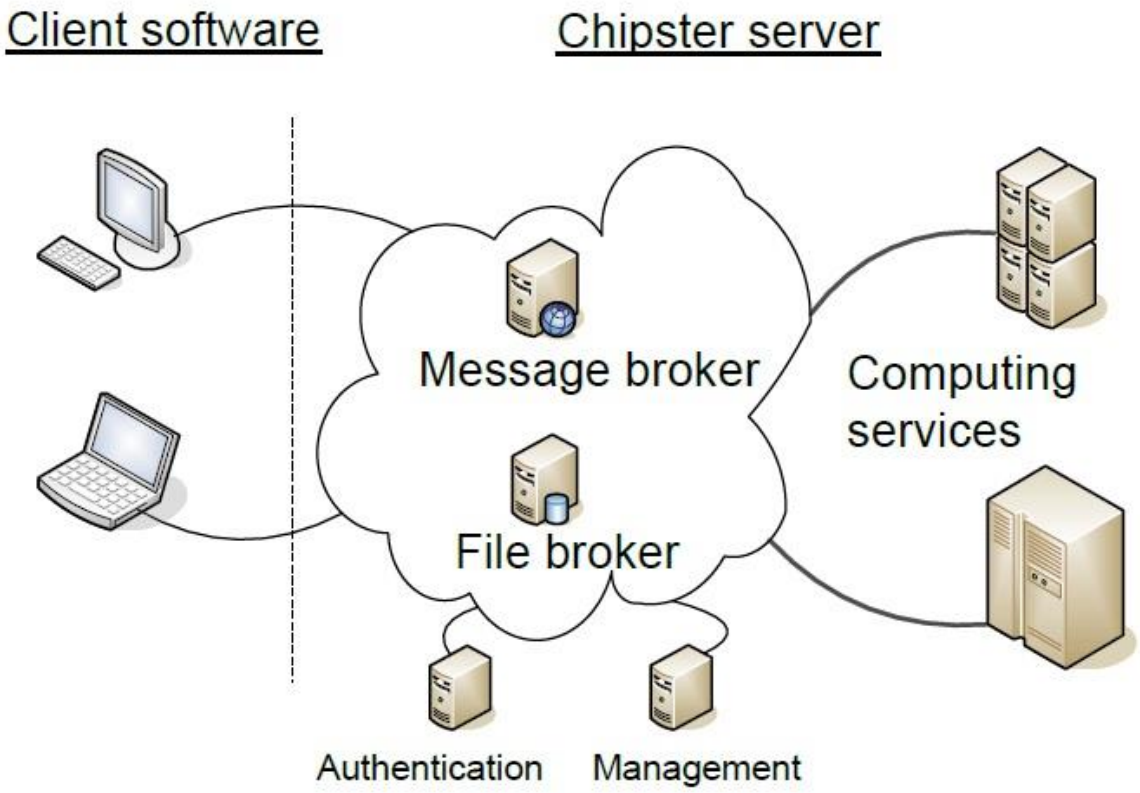
Chipster

- Provides an easy access to over 360 analysis tools
  - Command line tools
  - R/Bioconductor packages
- Free, open source software

- What can I do with Chipster?
  - analyze and integrate high-throughput data
  - visualize data efficiently
  - share analysis sessions
  - save and share automatic workflows

Technical aspects

- **Client-server system**
  - Enough CPU and memory for large analysis jobs
  - Centralized maintenance
- **Easy to install**
  - Client uses Java Web Start
  - Server available as a virtual machine



Chipster

Open source platform for data analysis

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

- For developers:
  - Open source project
  - Tool editor

Welcome to Chipster

Chipster is a user-friendly software for analyzing high-throughput data such as NGS and microarrays. It contains over 360 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 v3.12

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

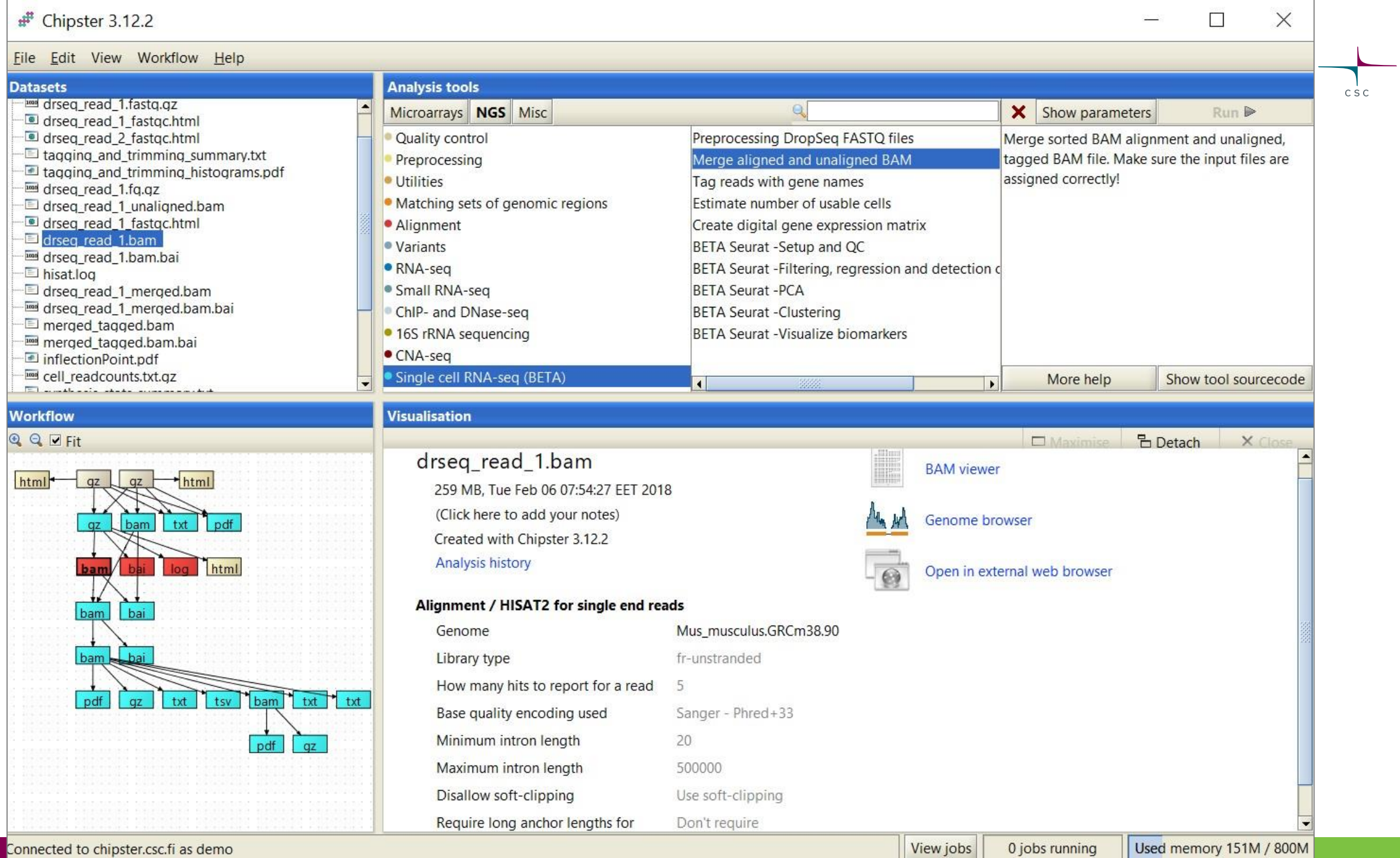
If you have trouble launching Chipster, read this

News and resources:

- 25.10.2017 Version 3.12 released
- 11.10.2016 Webinar: Beginner's introduction to RNA-seq data analysis using Chipster
- 10.10.2016 Instructions for stranded RNA-seq data
- 19.8.2014 RNA-seq data analysis guidebook with Chipster instructions
- News archive

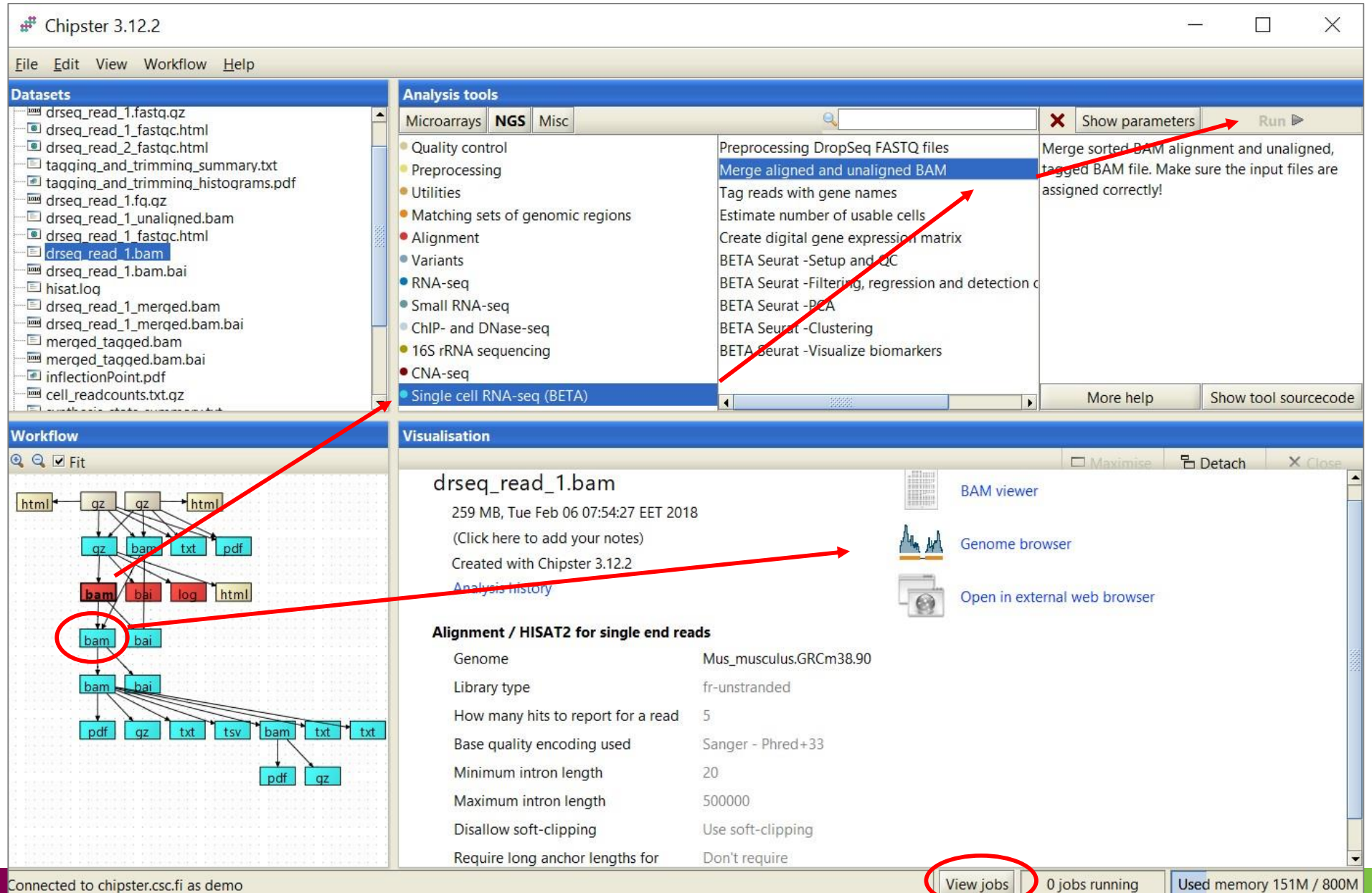
Training:

- 9.2.2018 Single cell RNA-seq data analysis using Chipster, CSC
- 16.1.2018 Webinar: VirusDetect pipeline


Chipster 3.12.2
File Edit View Workflow Help
Datasets
drseq_read_1.fastq.qz
drseq_read_1_fastqc.html
drseq_read_2_fastqc.html
tagging_and_trimming_summary.txt
tagging_and_trimming_histograms.pdf
drseq_read_1.fq.qz
drseq_read_1_unaligned.bam
drseq_read_1_fastqc.html
drseq_read_1.bam
drseq_read_1.bam.bai
hisat.log
drseq_read_1_merged.bam
drseq_read_1_merged.bam.bai
merged_tagged.bam
merged_tagged.bam.bai
inflectionPoint.pdf
cell_readcounts.txt.qz
Analysis tools
Microarrays NGS Misc
Show parameters Run
Quality control
Preprocessing
Utilities
Matching sets of genomic regions
Alignment
Variants
RNA-seq
Small RNA-seq
ChIP- and DNase-seq
16S rRNA sequencing
CNA-seq
Single cell RNA-seq (BETA)
Preprocessing DropSeq FASTQ files
Merge aligned and unaligned BAM
Tag reads with gene names
Estimate number of usable cells
Create digital gene expression matrix
BETA Seurat -Setup and QC
BETA Seurat -Filtering, regression and detection c
BETA Seurat -PCA
BETA Seurat -Clustering
BETA Seurat -Visualize biomarkers
Merge sorted BAM alignment and unaligned, tagged BAM file. Make sure the input files are assigned correctly!
More help
Show tool sourcecode
Workflow
Fit
html
gz
gz
html
gz
bam
txt
pdf
bam
bai
log
html
bam
bai
bam
bai
pdf
gz
txt
tsv
bam
txt
txt
pdf
gz
Visualisation
Maximise
Detach
Close
drseq_read_1.bam
259 MB, Tue Feb 06 07:54:27 EET 2018
(Click here to add your notes)
Created with Chipster 3.12.2
Analysis history
BAM viewer
Genome browser
Open in external web browser
Alignment / HISAT2 for single end reads
Genome Mus_musculus.GRCm38.90
Library type fr-unstranded
How many hits to report for a read 5
Base quality encoding used Sanger - Phred+33
Minimum intron length 20
Maximum intron length 500000
Disallow soft-clipping Use soft-clipping
Require long anchor lengths for Don't require
Connected to chipster.csc.fi as demo
View jobs
0 jobs running
Used memory 151M / 800M

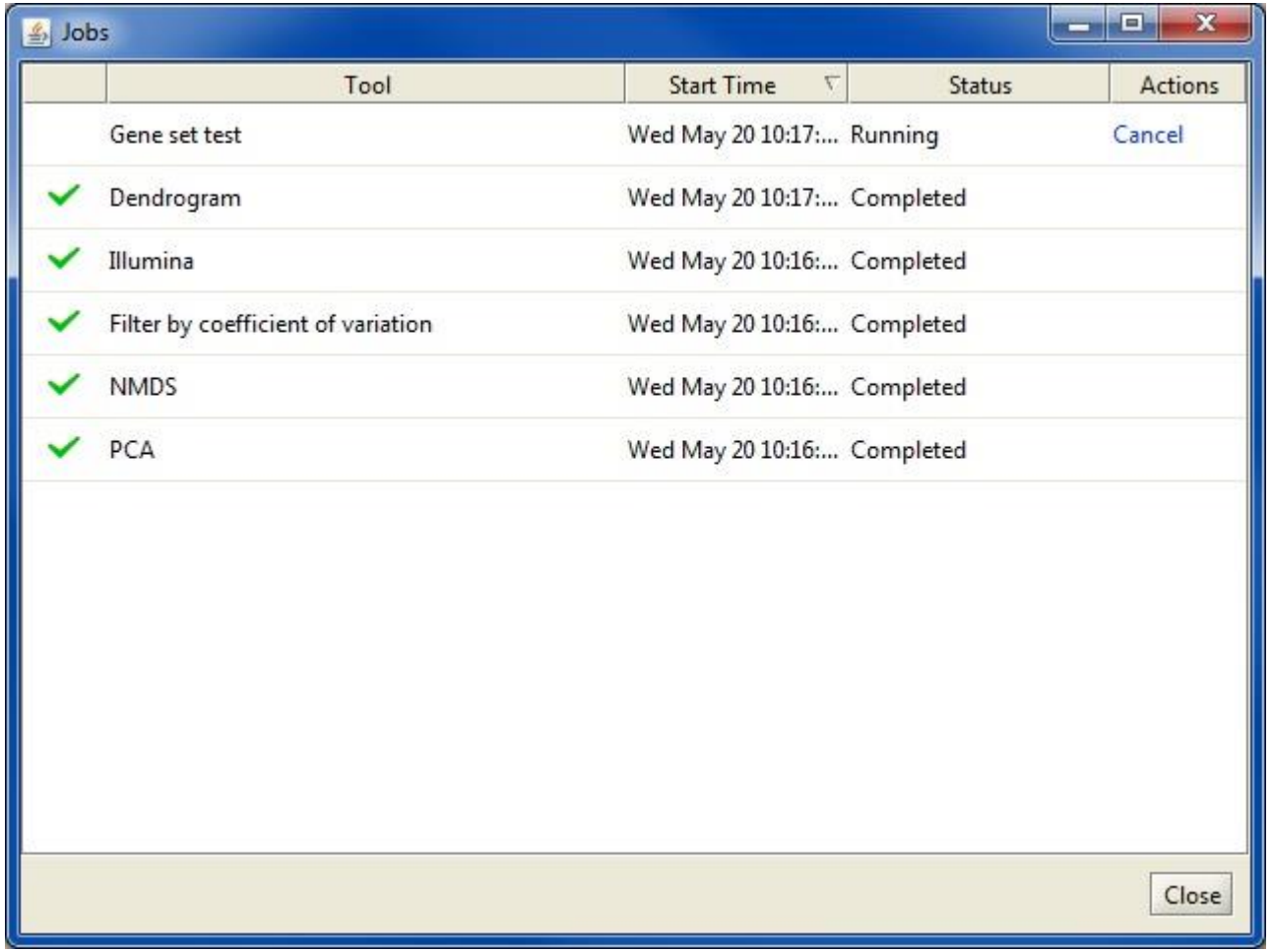
Mode of operation

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



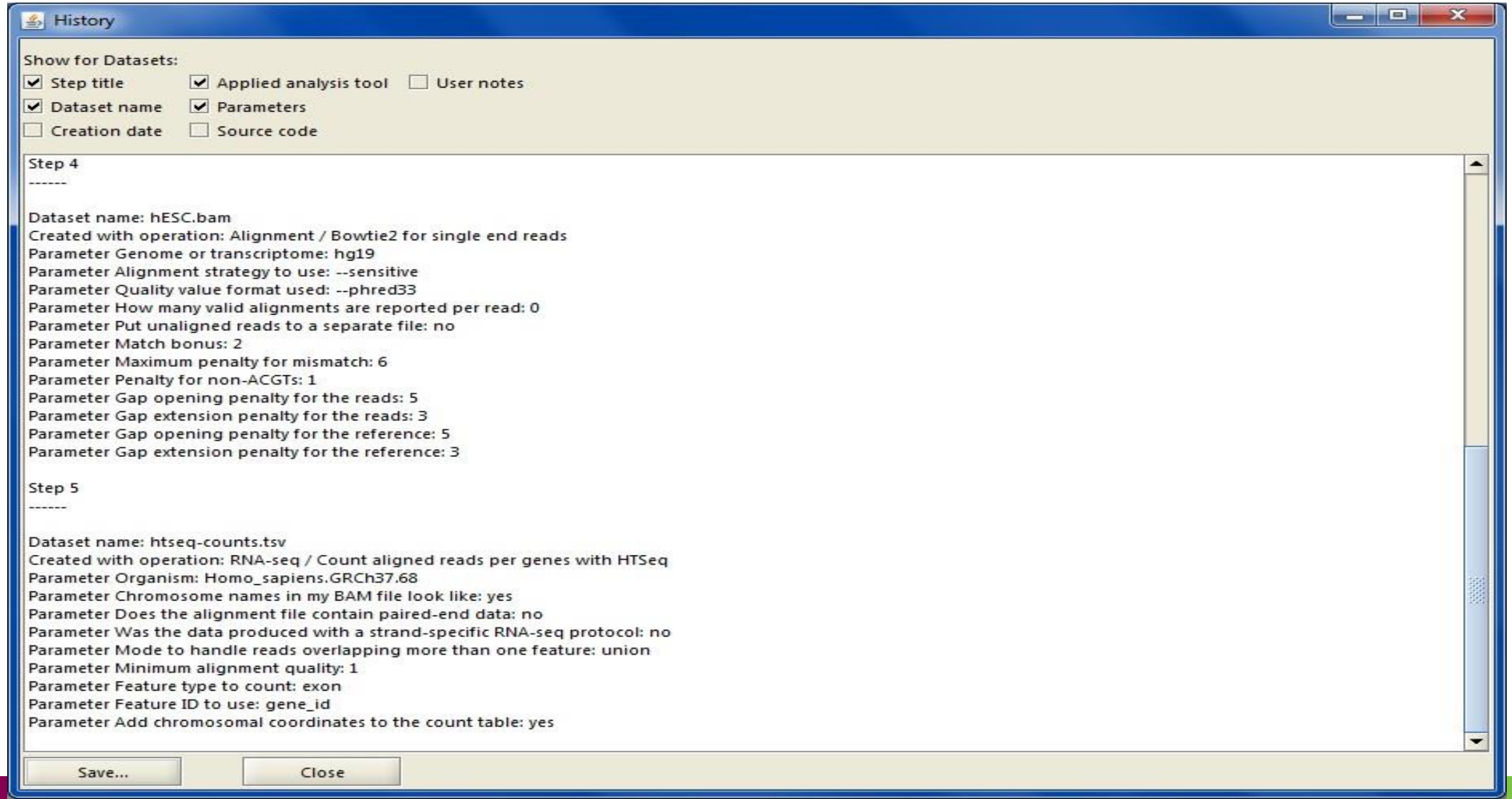
Job manager

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



Analysis history is saved automatically

-you can add tool source code to reports if needed



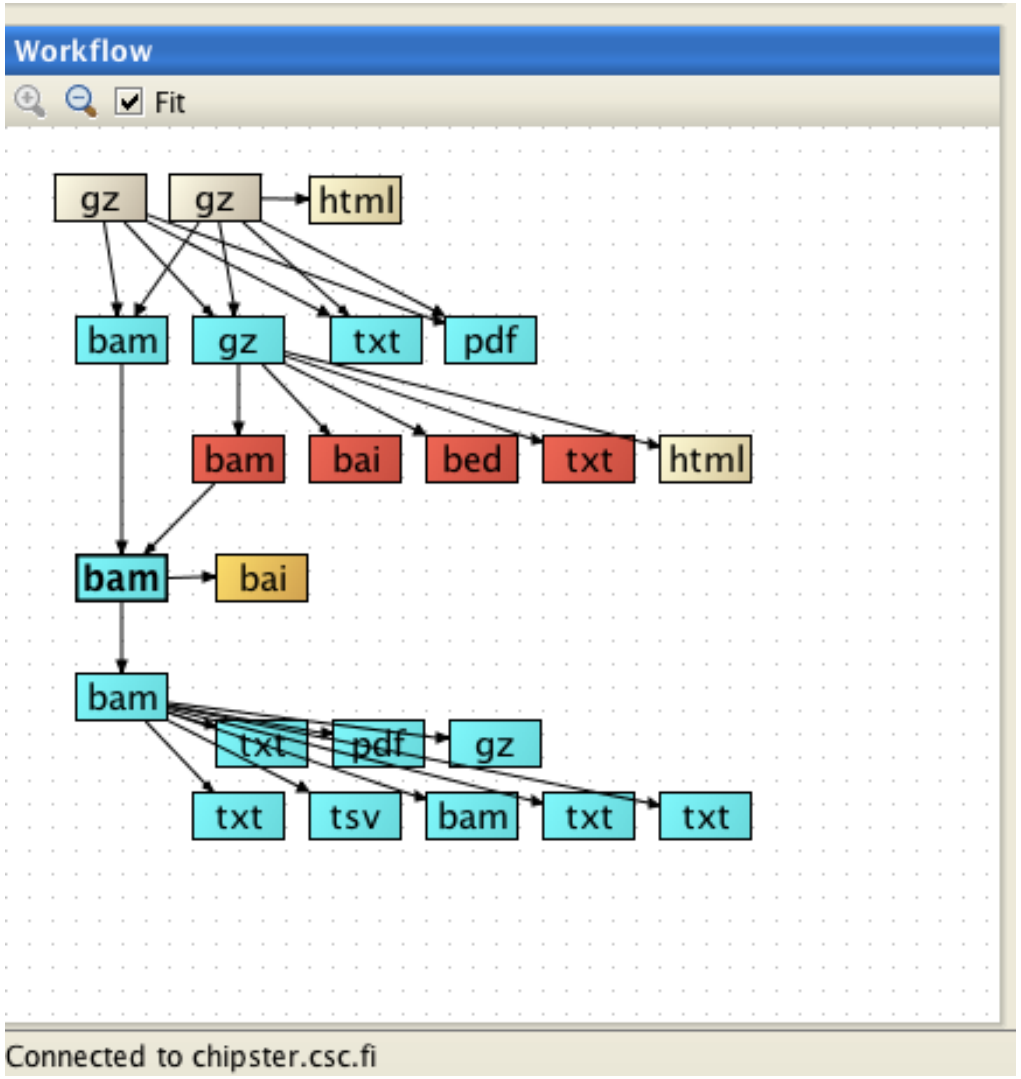
Analysis sessions

- **Remember to save the analysis session.**
  - 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.

- **You can save a session locally (on your computer)**
- **and in the cloud**
  - but note that the cloud sessions are not stored forever!

Workflow panel

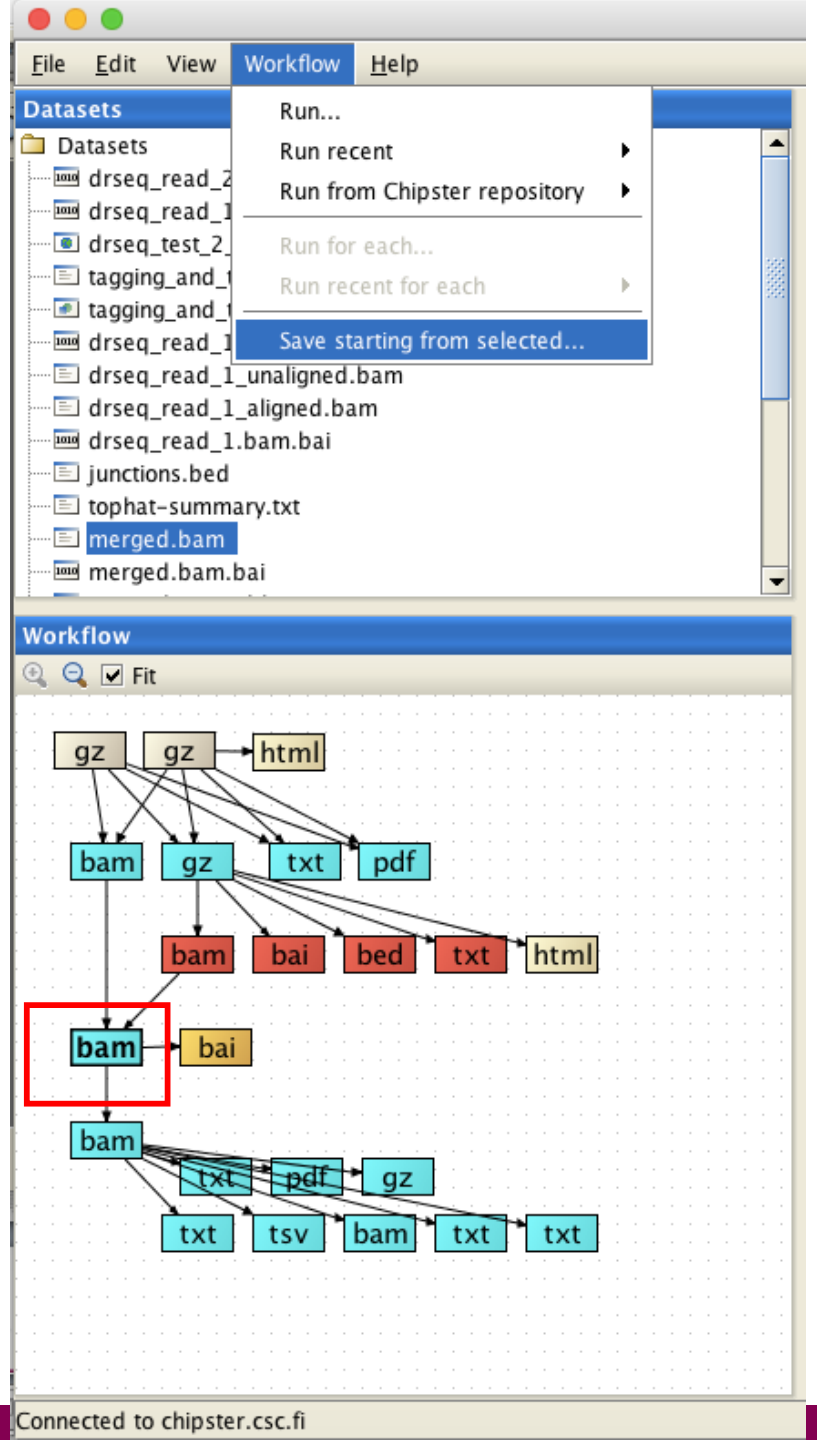
- Shows the relationships of the files
- You can move the boxes around, and zoom in and out.
- Several files can be selected by keeping the Ctrl key down
- Right clicking on the data file allows you to
  - Save an individual result file (”Export”)
  - Delete
  - Link to another data file
  - Save workflow



Workflow – reusing and sharing your analysis pipeline

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

Saving and using workflows



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

Analysis tool overview

- **160 NGS tools for**
  - RNA-seq
  - single cell RNA-seq
  - small RNA-seq
  - exome/genome-seq
  - ChIP-seq
  - FAIRE/DNase-seq
  - CNA-seq
  - Metagenomics (16S rRNA)

- **140 microarray tools for**
  - gene expression
  - miRNA expression
  - protein expression
  - aCGH
  - SNP
  - integration of different data

- **60 tools for sequence analysis**
  - BLAST, EMBOSS, MAFFT
  - Phylip

Visualizing the data

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

- **Two types of visualizations**

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

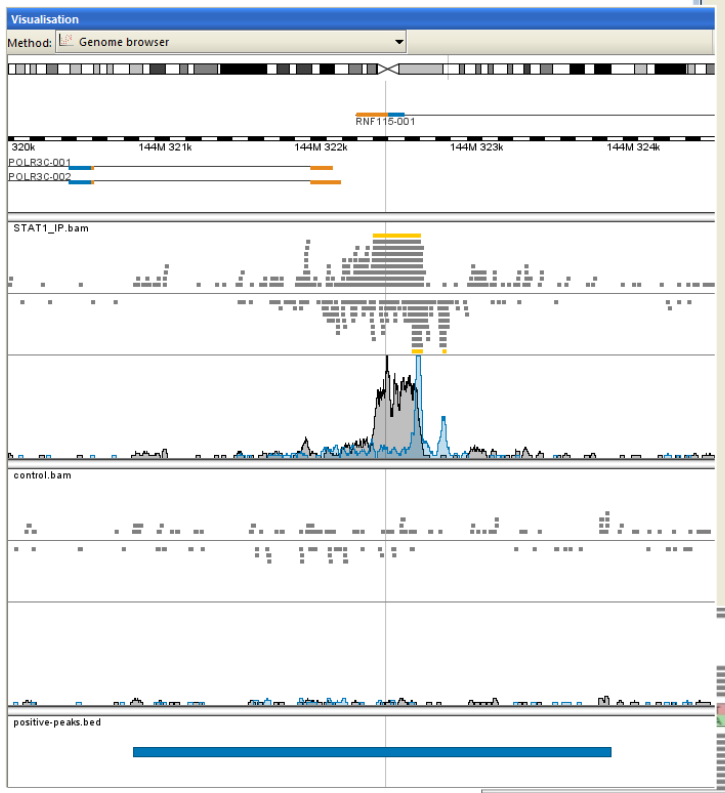
  2. **Static images produced by analysis tools**
     - Select from Analysis tools / Visualisation
     - View by double clicking on the image file
     - Save by right clicking on the file name and choosing "Export"

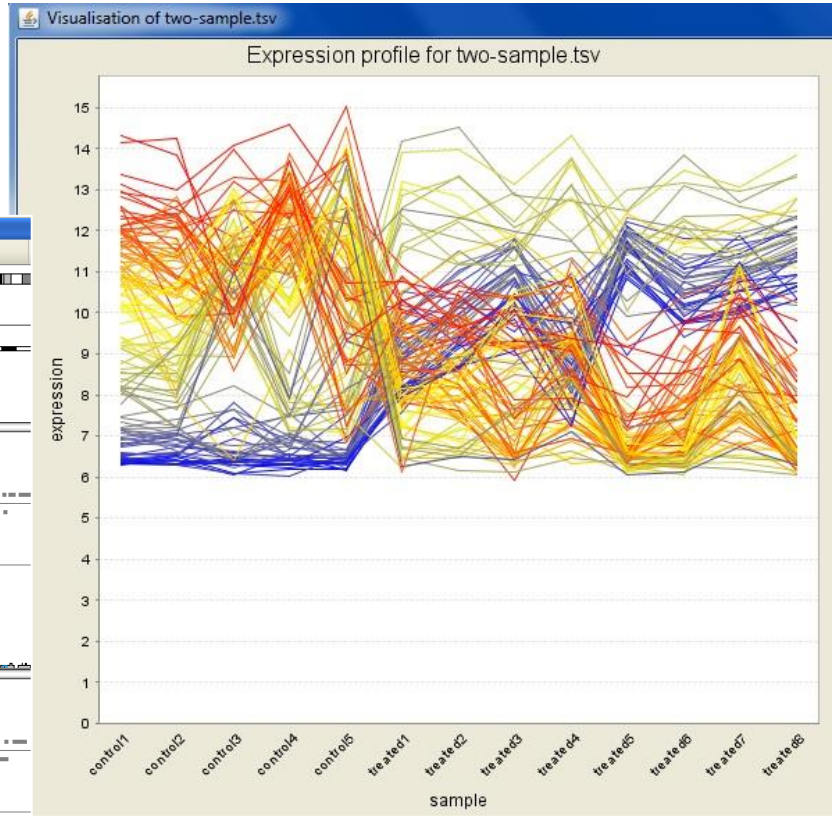
Interactive visualizations by the client

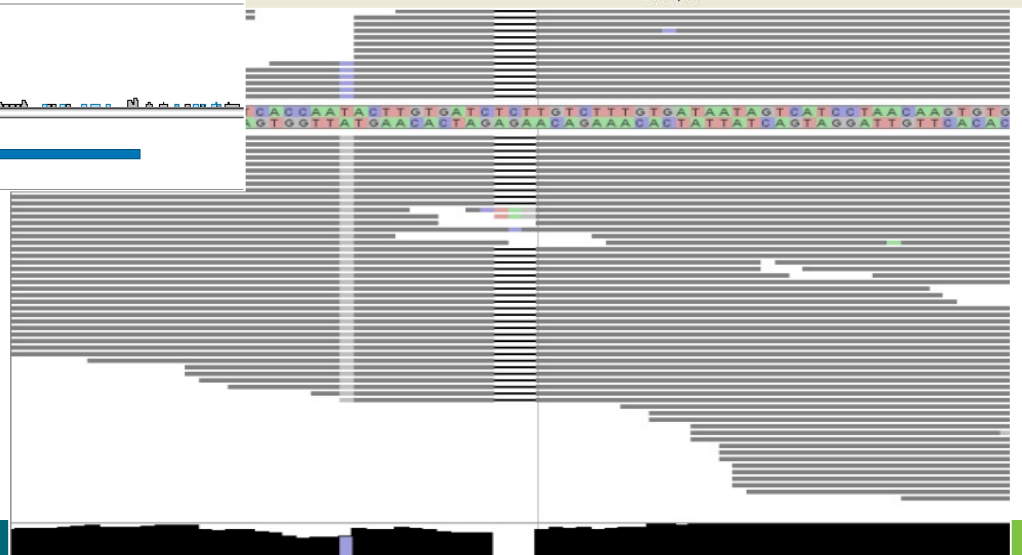
- Genome browser
- Spreadsheet
- Histogram
- Venn diagram
- Scatterplot
- 3D scatterplot
- Volcano plot
- Expression profiles
- Clustered profiles
- Hierarchical clustering
- SOM clustering

Available actions:

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

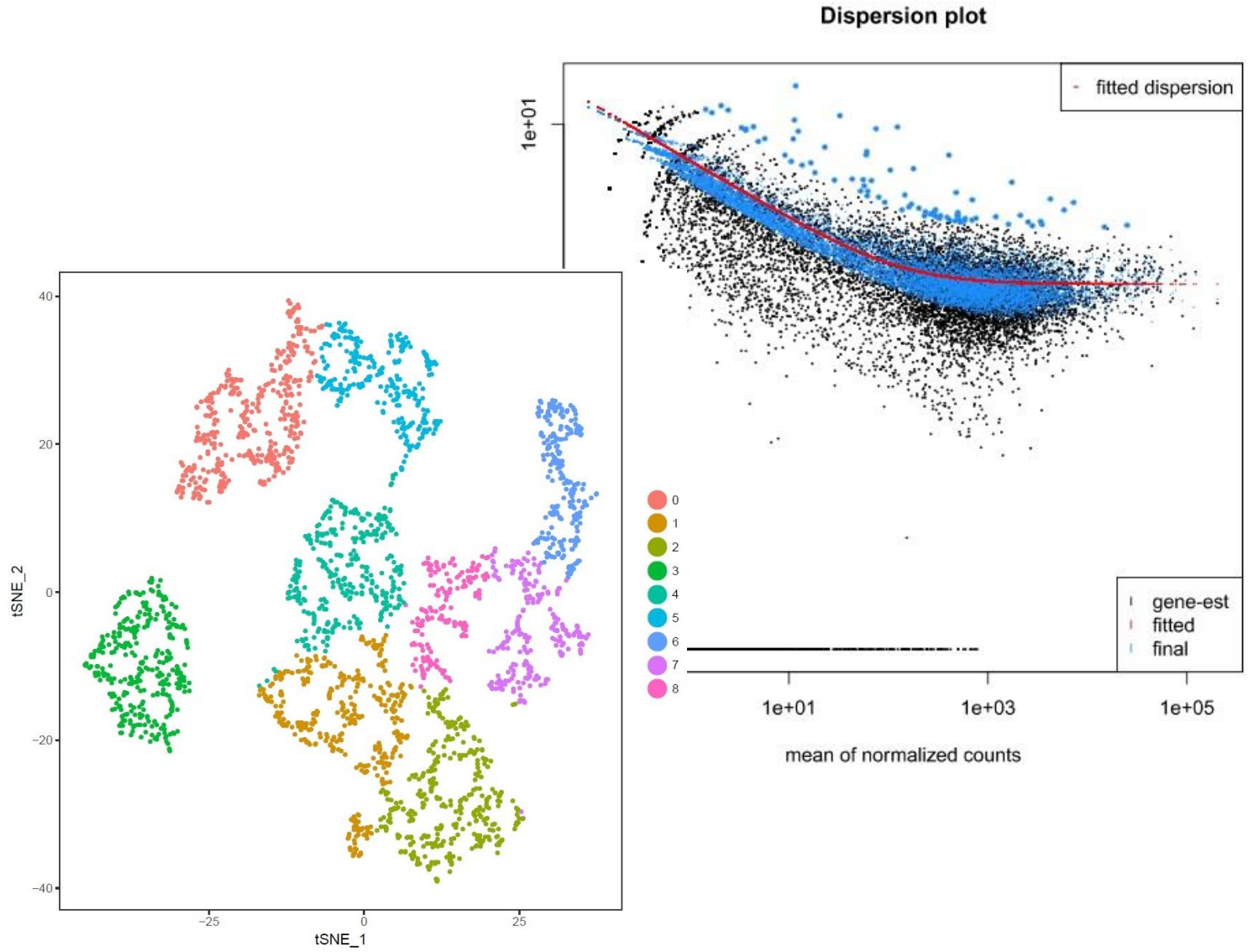






Static images produced by R/Bioconductor

- Dispersion plot
- MA plot
- MDS plot
- Box plot
- Histogram
- Heatmap
- tSNE plot
- Violin plot
- PCA plot
- Dendrogram
- K-means clustering
- SOM-clustering
- Etc…



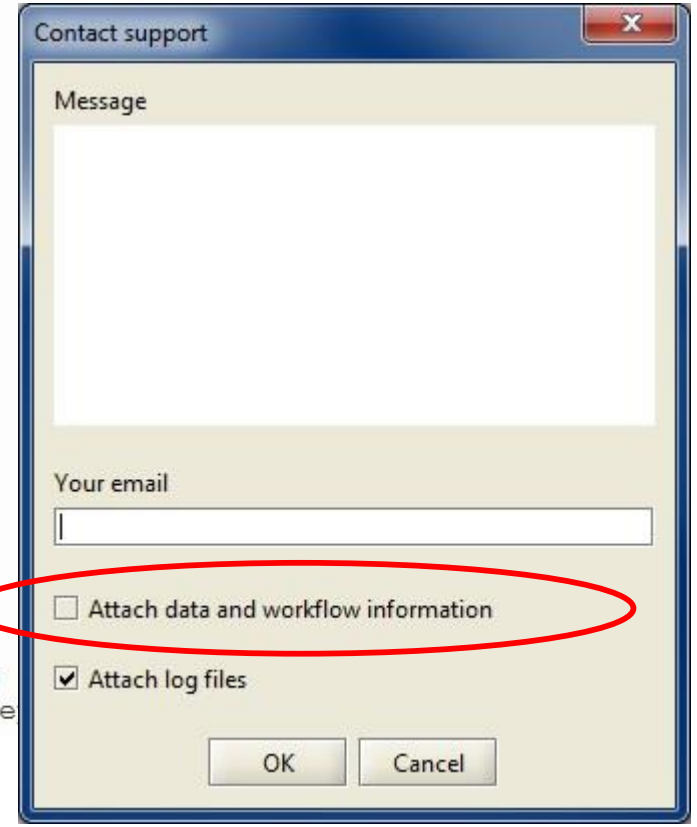
Options for importing data to Chipster

- **Import files/ Import folder**
- **Import from URL**
  - Utilities / Download file from URL directly to server
- **Open an analysis session**
  - Files / Open session
- **Import from 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 RNA-seq data files can I use in Chipster?**
  - Compressed files (.gz) and tar packages (.tar) are ok
  - FASTQ, BAM, read count files (.tsv), GTF

Problems? Send us a support request

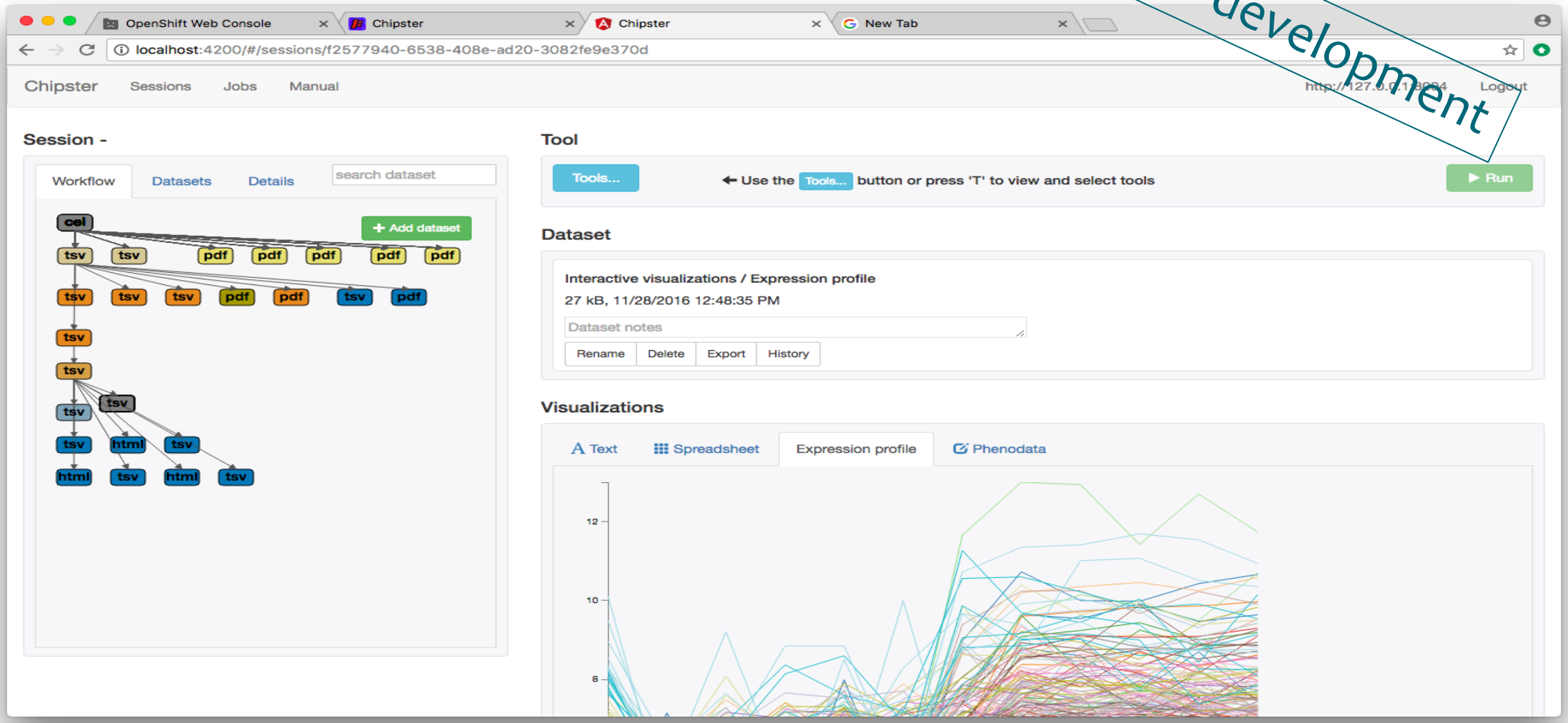
-request includes the error message and link to analysis session (optional)

```
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-2.12'
> 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" (Illumina normalization using
BeadSummaryData files, and using lumi methodology. If you have a BeadSummaryData that re
```



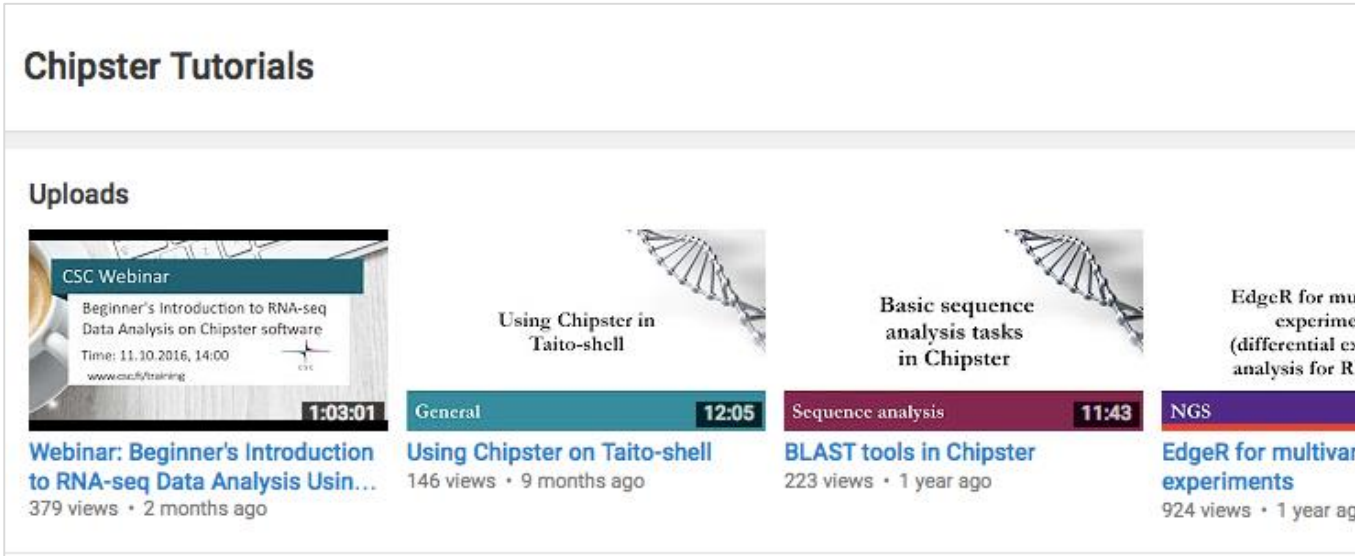
Chipster Web version

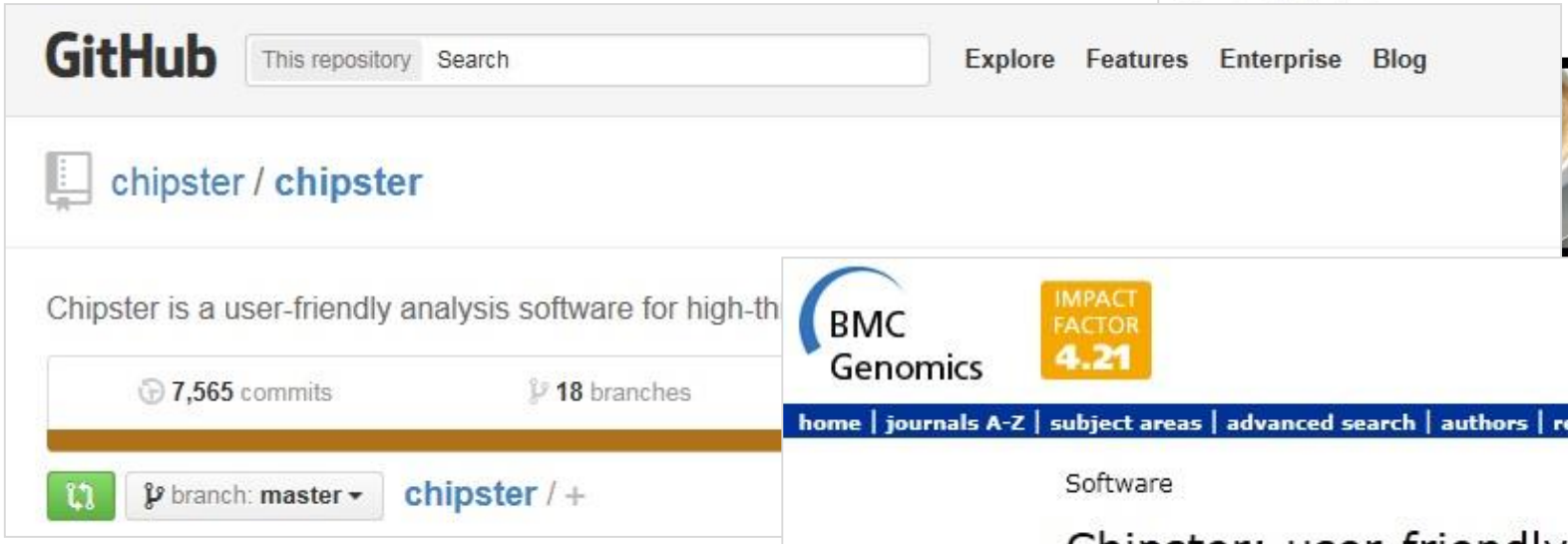
Under development

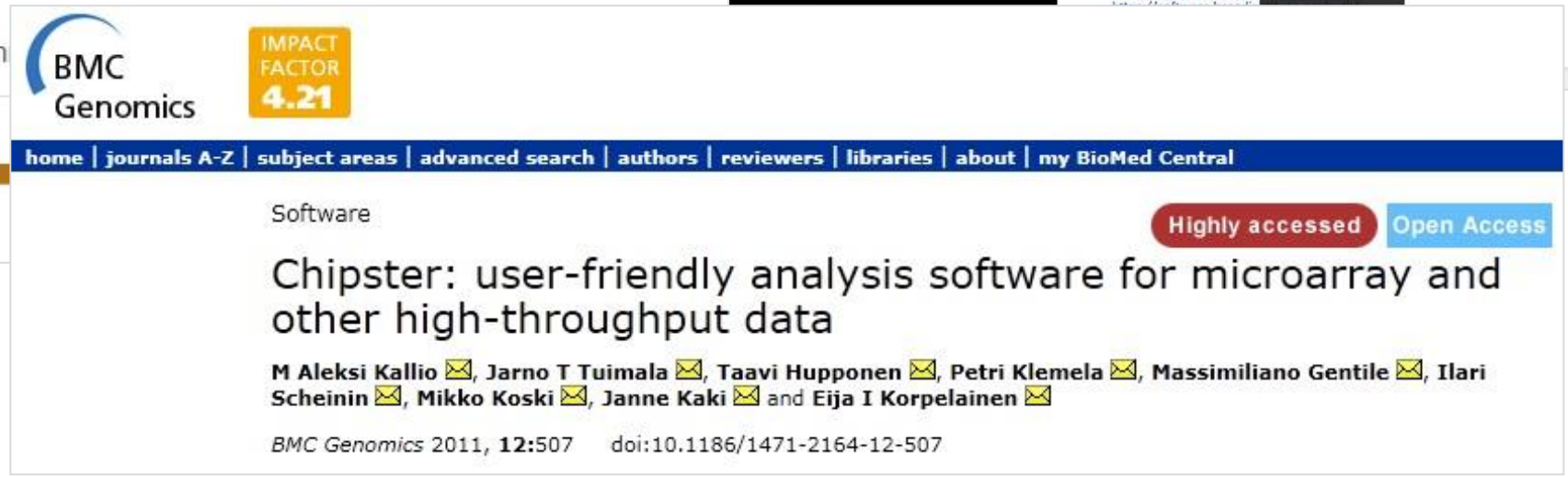


More info

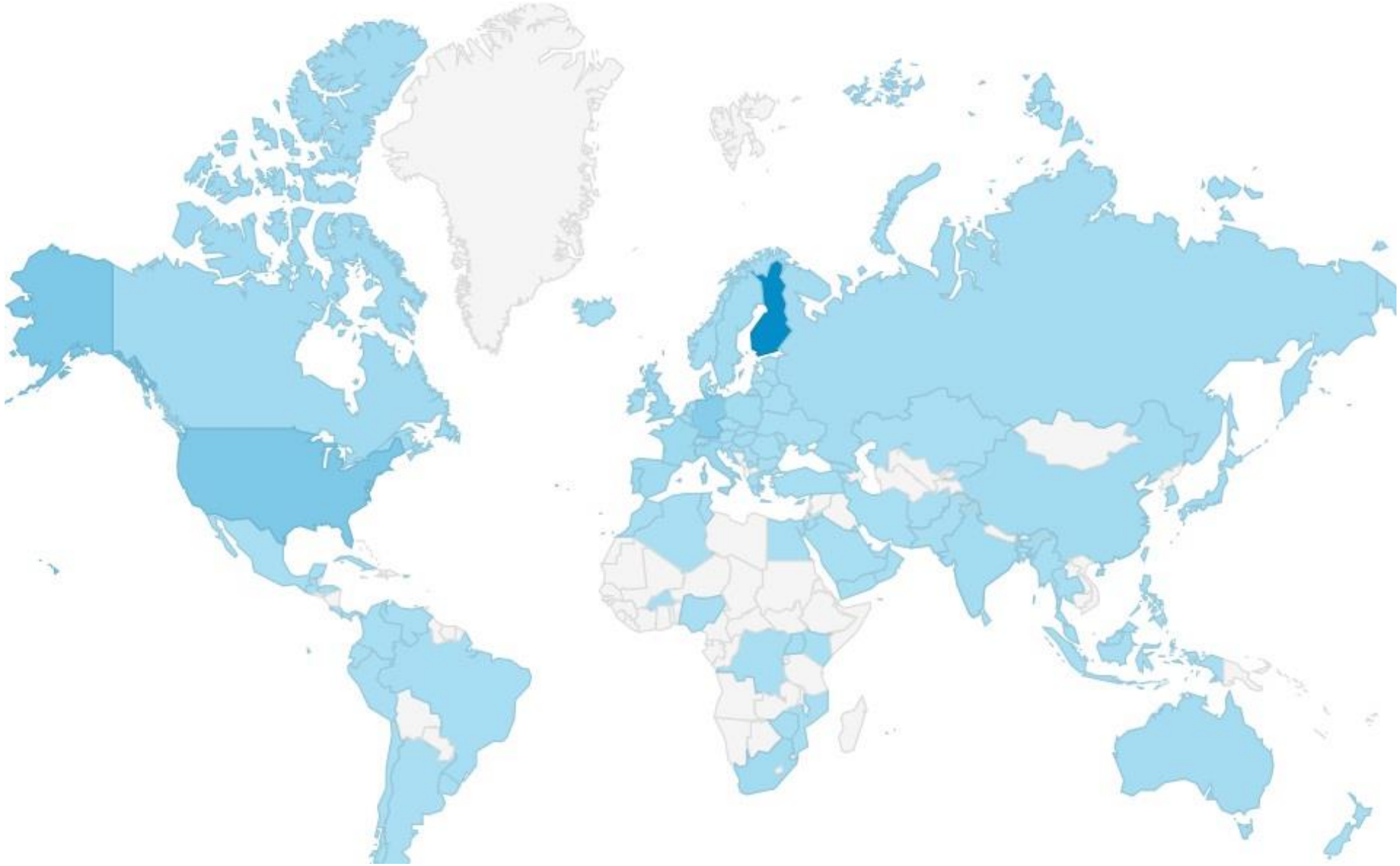
- chipster@csc.fi
- http://chipster.csc.fi
- Chipster tutorials in YouTube







Acknowledgements to Chipster users and contibutors



Introduction to single cell RNA-seq

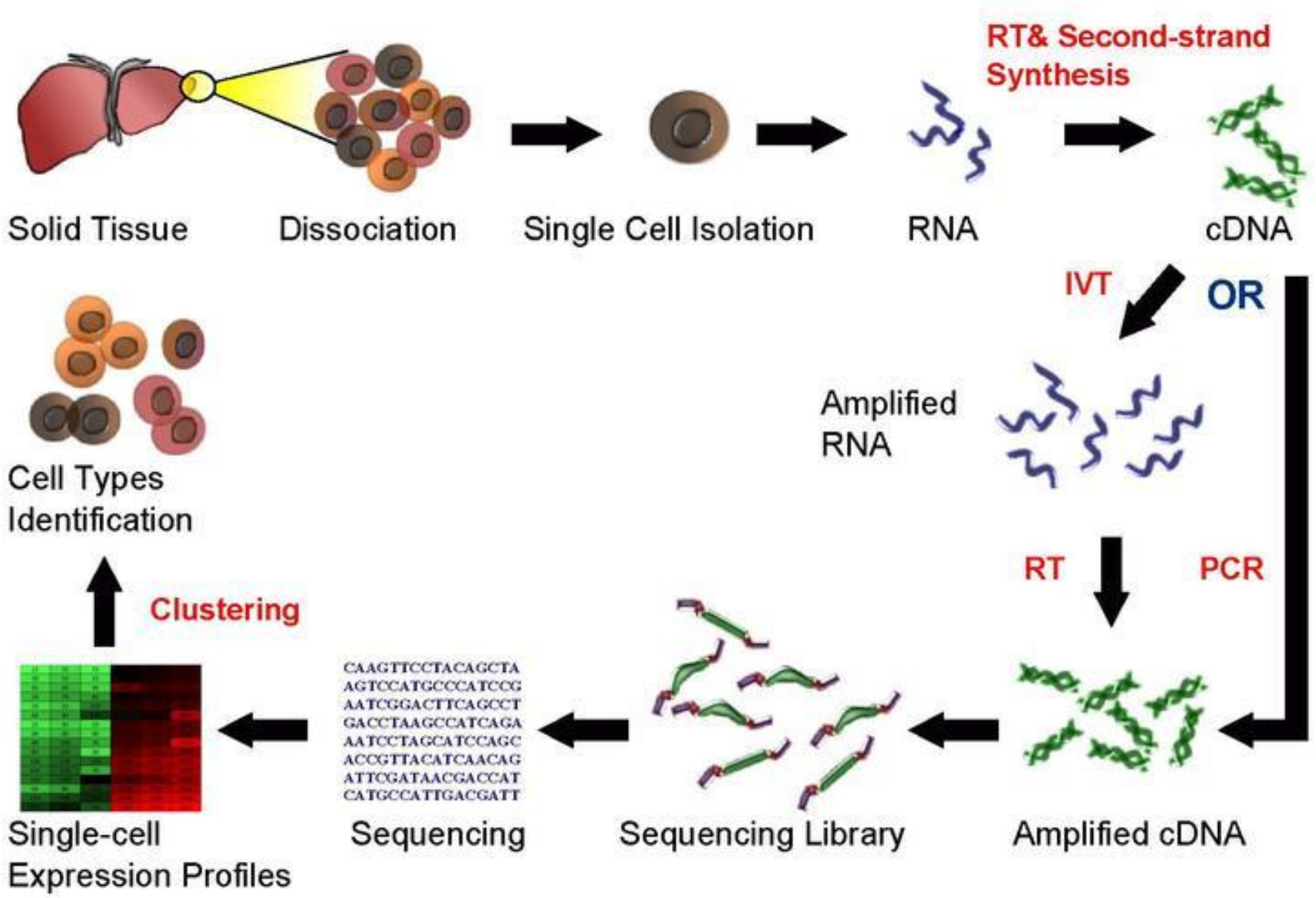
Single cell RNA-seq

- New technology, data analysis methods are actively developed
- Measures distribution of expression levels for each gene across a population of cells
- Allows to study cell-specific changes in transciptome
- Applications
  - Identifying cell sub-populations within a biological condition
  - Studying dynamic processes like differentiation using pseudotime ordering
    - Swithces
    - Branch points
  - Transcriptional regulatory networks
- Datasets range from 10^2 to 10^5 cells

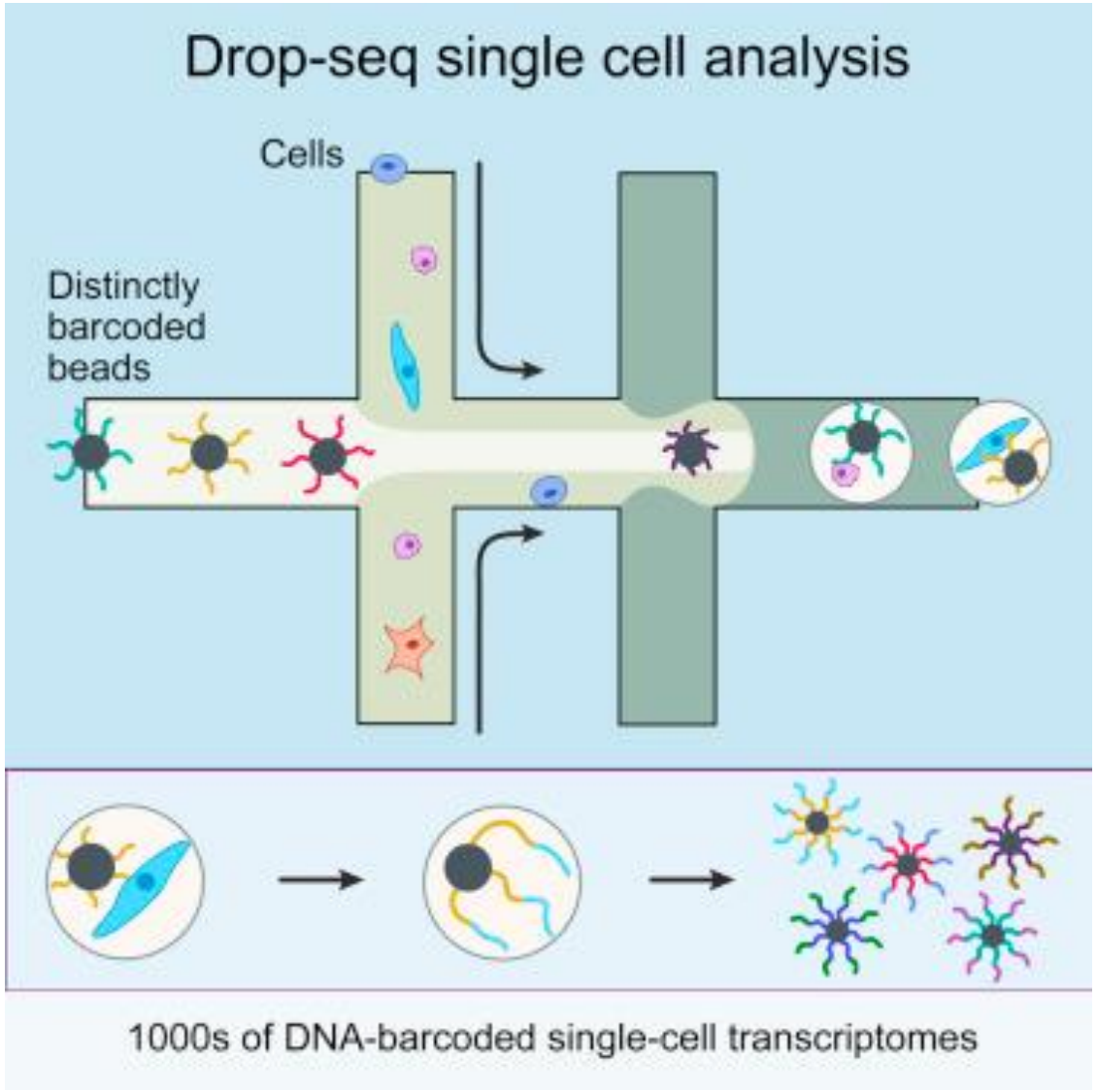
Single cell RNA-seq technology

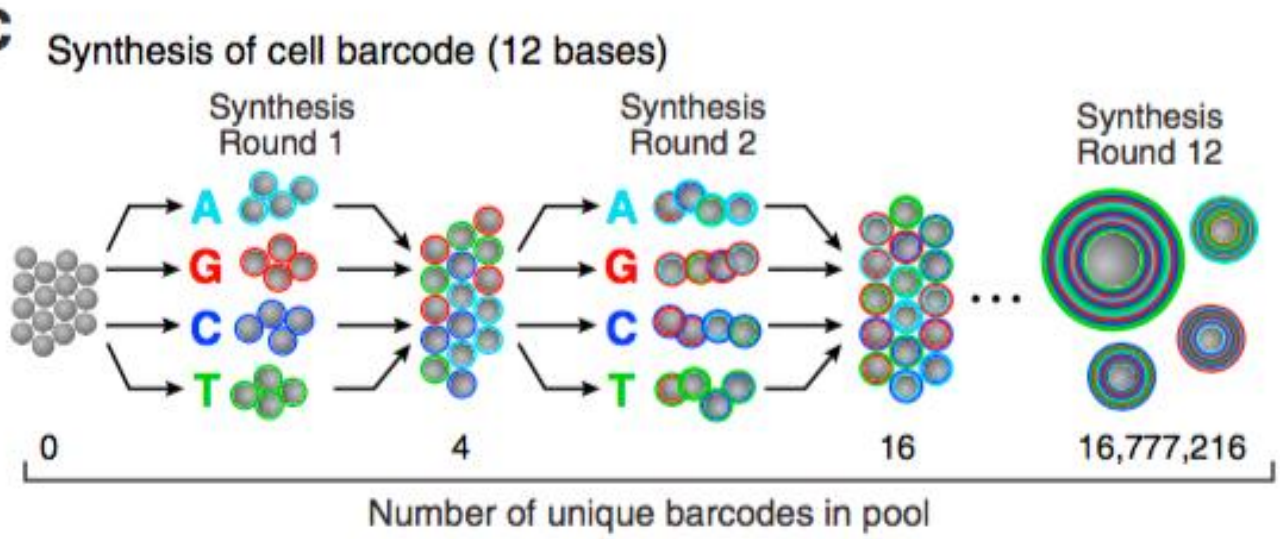
- Protocols:
  - **DropSeq**
  - InDrop
  - CELL-seq
  - SMART-seq
  - SMARTer
  - MARS-seq
  - SCRB-seq
  - Seq-well
  - STRT-seq
- Commercial platforms:
  - Fluidigm C1 (FuGU)
  - 10x Genomics Chromium (FIMM, BTK)

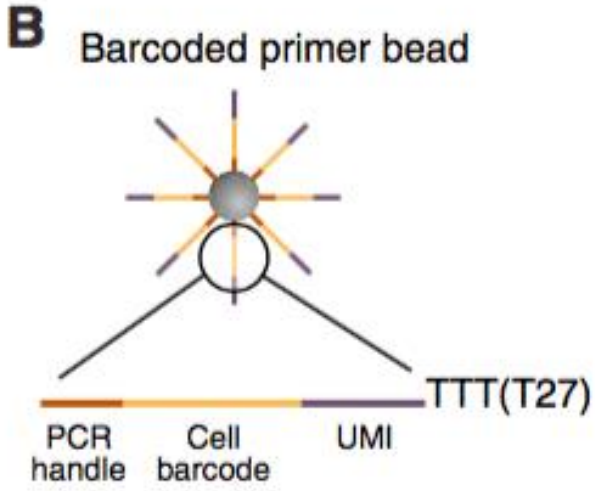
Single Cell RNA Sequencing Workflow

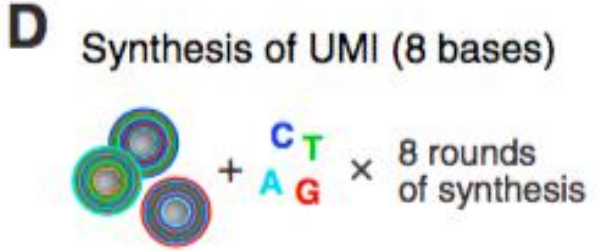


https://hemberg-lab.github.io/scRNA.seq.course/introduction-to-single-cell-rna-seq.html



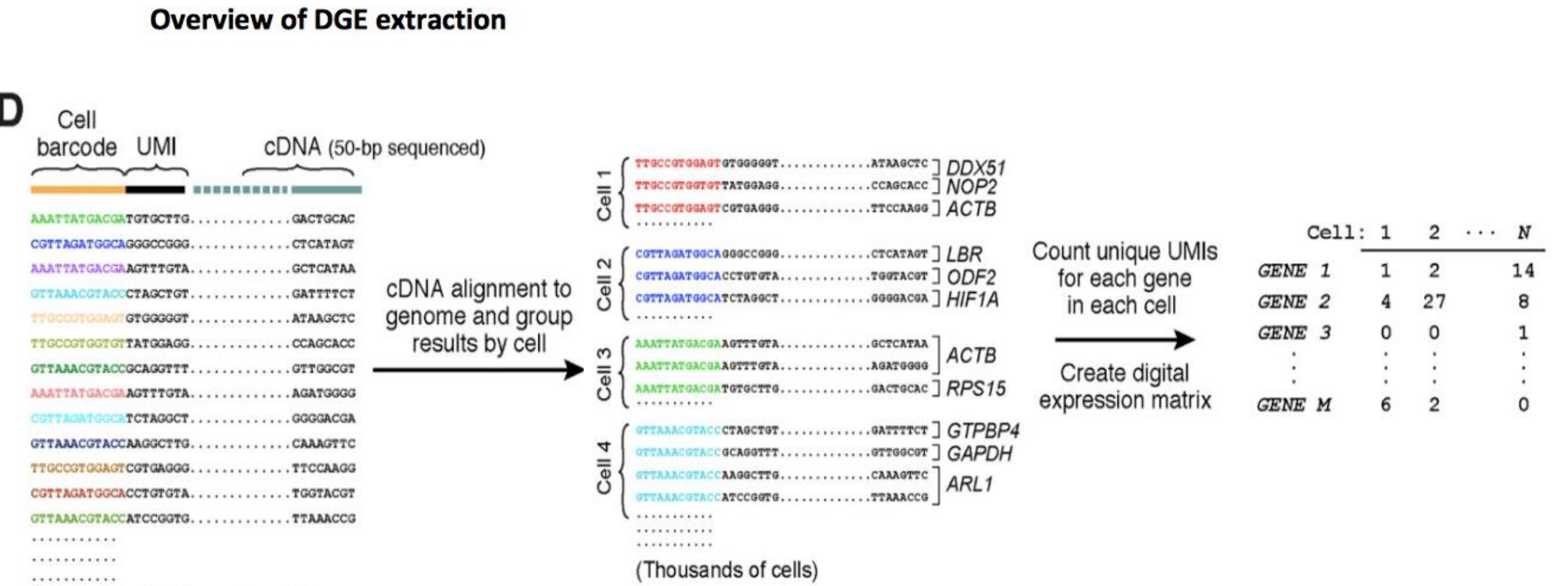






- Millions of the same **cell barcode** per bead
- 4^8 different **molecular barcodes** (UMIs) per bead

DropSeq data preprocessing overview



What can go wrong in Drop-seq?

1. Ideally there is one healthy cell in the droplet so that we get a STAMP (single-cell transcriptome attached to a microparticle). However, sometimes

- There is no cell in the droplet, just ambient RNA
  → Detect "empties" based on the small number of genes expressed and remove
- There are two (or more) cells in a droplet
  → Detect duplets (and multiplets) based on the large number of genes expressed and remove
- The cell in the droplet is broken/dead
  → Detect based on high proportion of reads mapping to mitochondrial genome and remove

2. Sometimes barcodes have synthesis errors in them, e.g. one base is missing

→ Detect by checking the distribution of bases at each position and fix the code or remove the cell

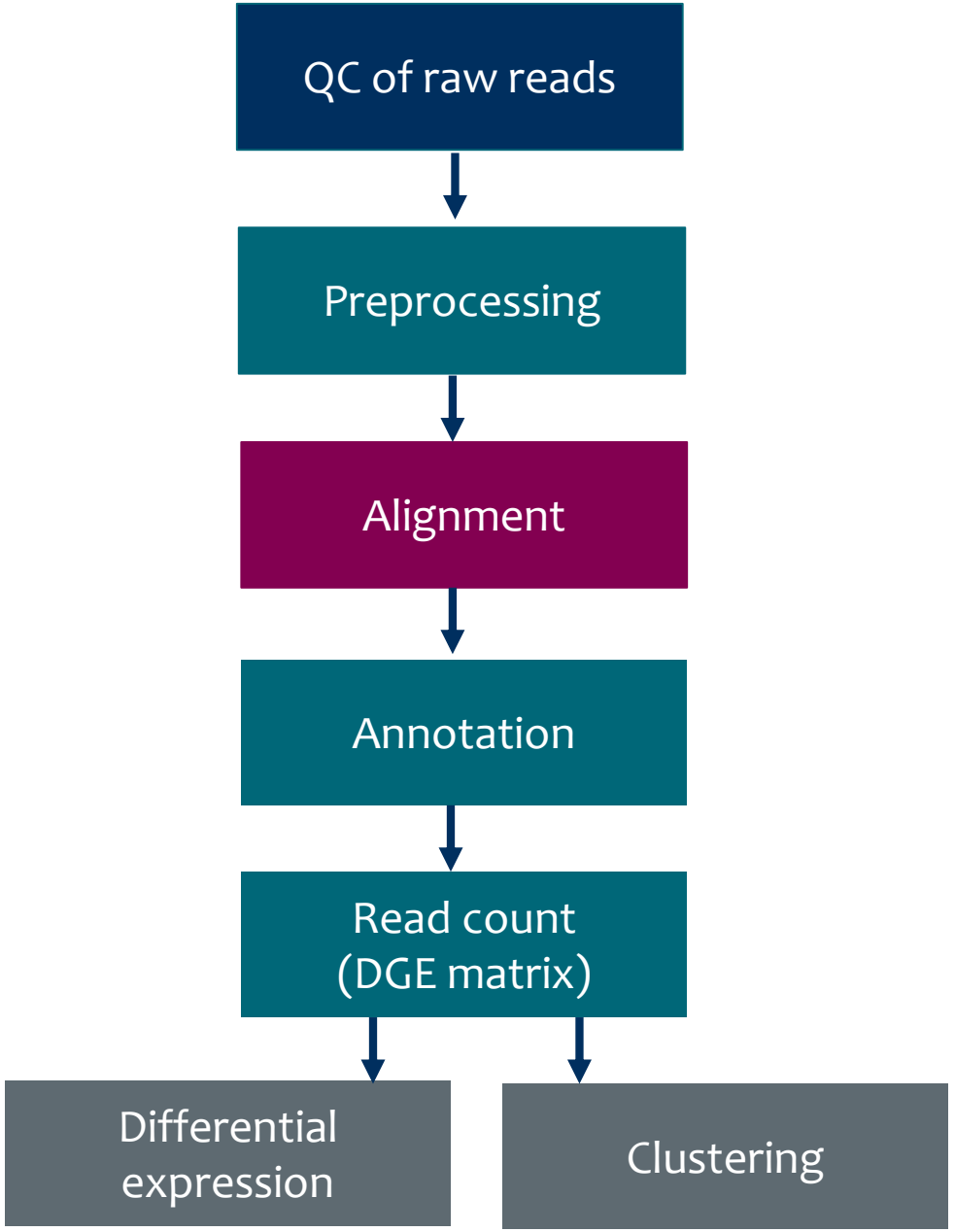
Single cell RNA-seq data analysis

Single cell RNA-seq data is challenging

- The detected expression level for many genes is zero
- Data is noisy. High level of variation due to
  - Capture efficiency (percentage of mRNAs captured)
  - Amplification bias (non-uniform amplification of transcripts)
  - Cells differ in terms of cell-cycle stage and size
- Complex distribution of expression values
  - Cell heterogeneity and the abundance of zeros give rise to multimodal distributions

→ Many methods used for bulk RNA-seq data won't work

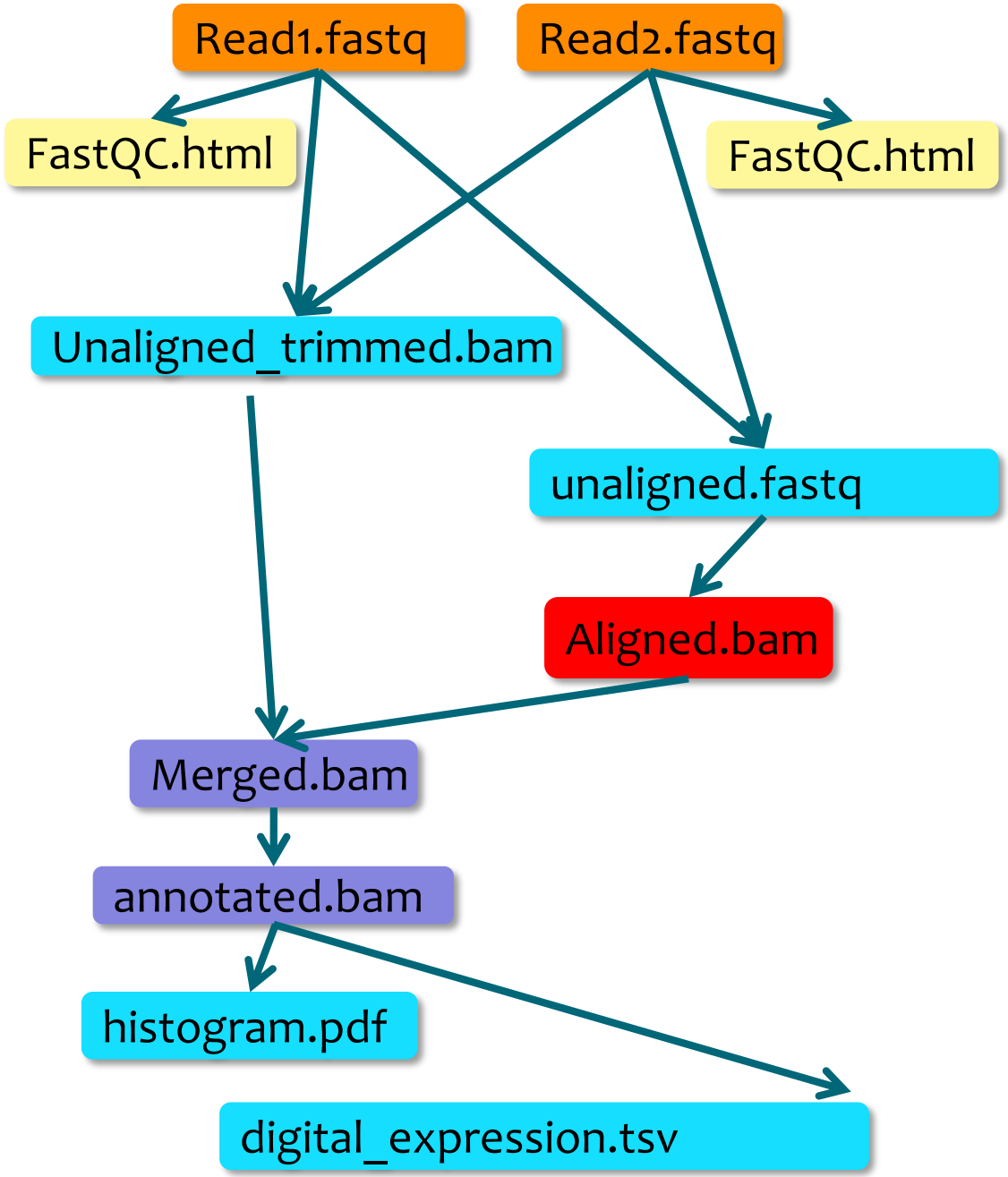
scRNA-seq data analysis steps



DropSeq data preprocessing

From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix



Import files

Quality control / FASTQC

Preprocess:

- FASTQ to BAM
- Tag BAM
- Filter and trim BAM
- BAM to FASTQ
- Filter too short reads

-filter bad quality barcodes
-trim adapter sequences
-trim polyA tails

Aligner (STAR, HISAT)

Merge BAM

Combines the barcode tags and the alignment info

Tag read with gene names

Estimate number of usable cells

Create digital expression matrix

+ Detect bead synthesis errors

From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix

What and why?

Potential problems

- low confidence bases, Ns
- sequence specific bias, GC bias
- adapters
- sequence contamination
- unexpected length of reads
- …

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**
- FastX
- PRINSEQ
- TagCleaner
- …

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
```

Probability:	0.0001
Phred score:	40
ASCII coding:	I (capital i)

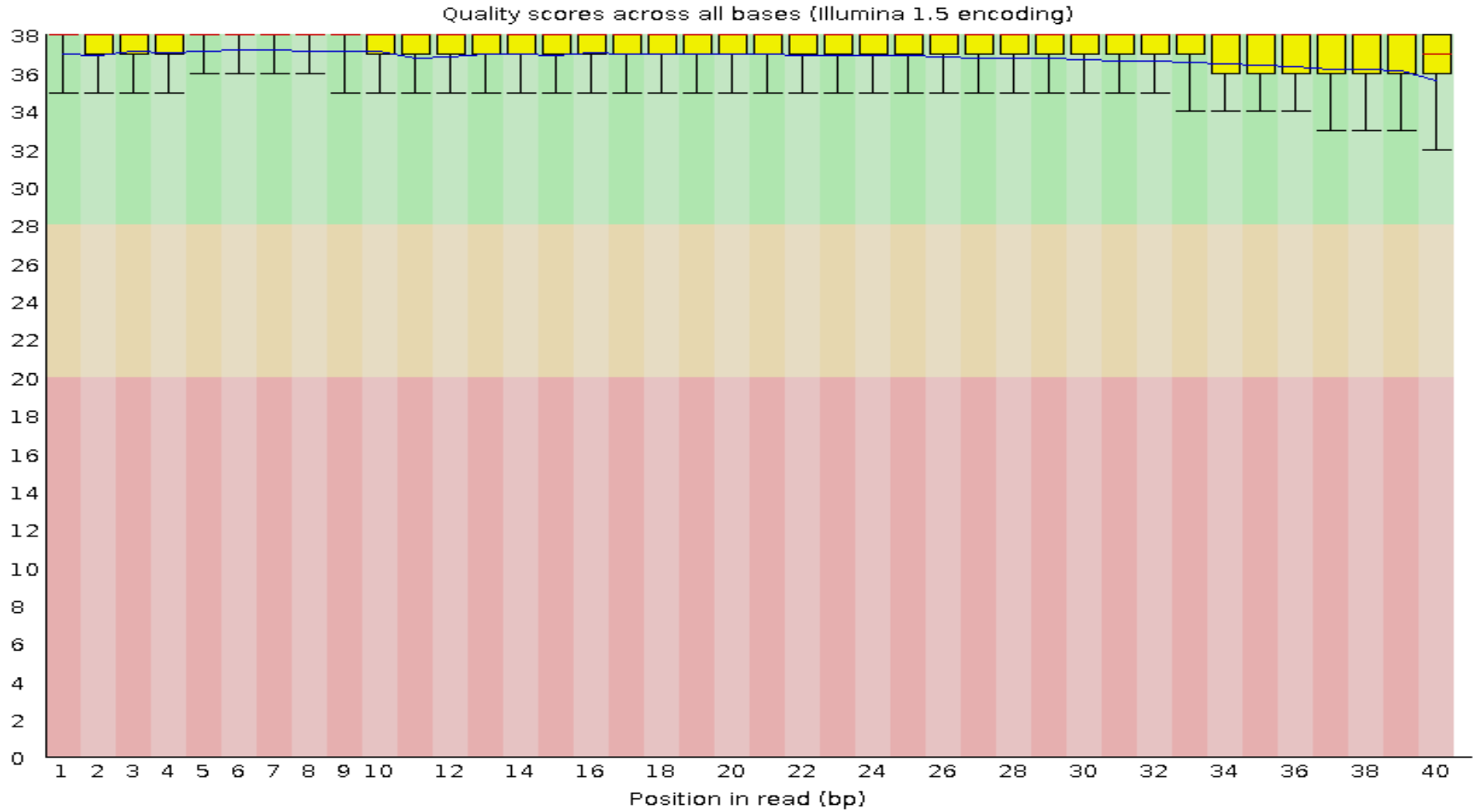
- "Sanger" encoding: numbers are shown as ASCII characters so that 33 is added to the Phred score
  - E.g. 39 is encoded as "H", the 72nd ASCII character (39+33 = 72)
  - Note that older Illumina data uses different encoding
    - Illumina1.3: add 64 to Phred
    - Illumina 1.5-1.7: add 64 to Phred,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

Base quality encoding systems

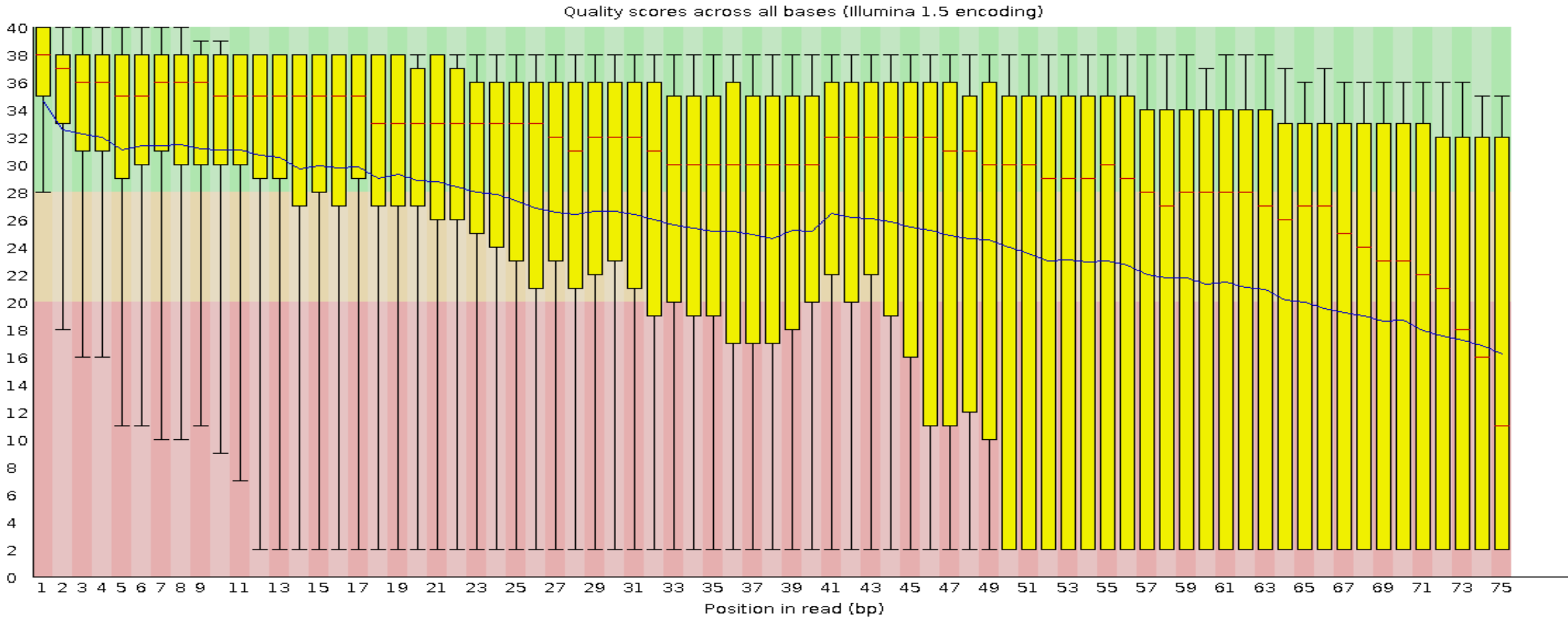
```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................
  ..LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmr
  |                          |    |        |                                  |
 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

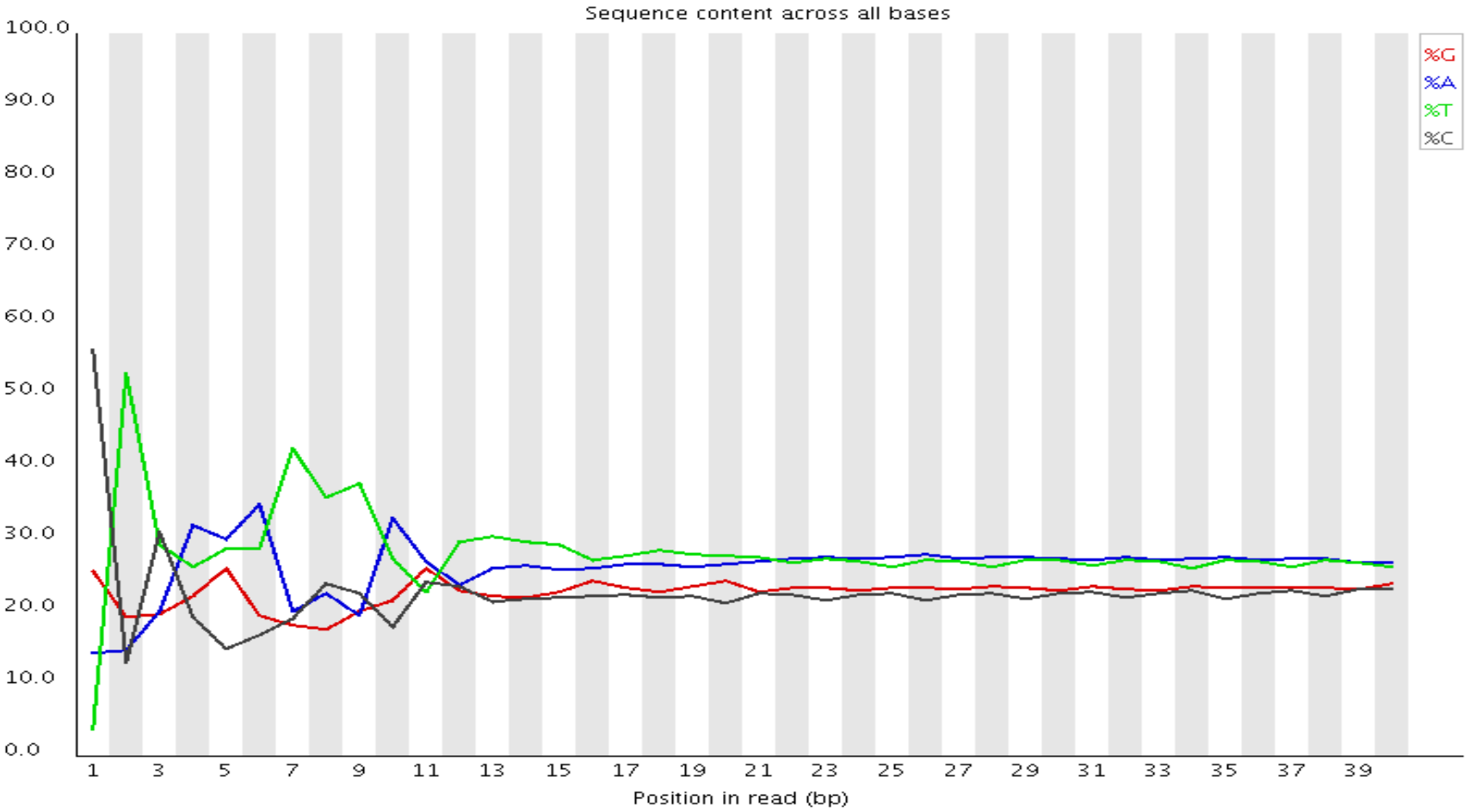
Per position base quality (FastQC)



Per position base quality (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
- Now: the barcodes

Check also the length of the reads

Measure	Value
Filename	reads.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	10157126
Sequences flagged as poor quality	0
Sequence length	20
%GC	51

- Barcode read: should be 20 bp
- Other length => BAM tagging won't work
- To get rid of shorter barcode reads:
  - select **BOTH FASTQ files**
  - tool **Preprocessing / Trim reads with Trimmomatic** with parameter **Minimum length of reads to keep = 20**
  - Continue analysis with the files that are named "--trimmed.fq.gz" (instead of the "--unpaired_trimmed.fq.gz" ones).
- To trim too long barcode reads:
  - after the previous step: select **read1** file, run **Preprocessing / Trim reads with Trimmomatic** with parameter **Number of bases to keep from the start = 20**

From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix

Preprocessing single cell DropSeq FASTQ files

- This tool is a combination of several tools:
  - DropSeq (TagBamWithReadSequenceExtended, FilterBam)
  - Picard (FASTQ to BAM, BAM to FASTQ) and
  - Trimmomatic (MINLEN)
- The steps are:
  1. Convert the FASTQ files into an unaligned BAM file
  2. Tag the reads in the BAM file with the cellular and molecular barcodes
  3. Filter and trim the reads in the BAM file
  4. Convert the tagged and trimmed BAM file back into a FASTQ file for the alignment
  5. Filter out too short reads of the FASTQ file

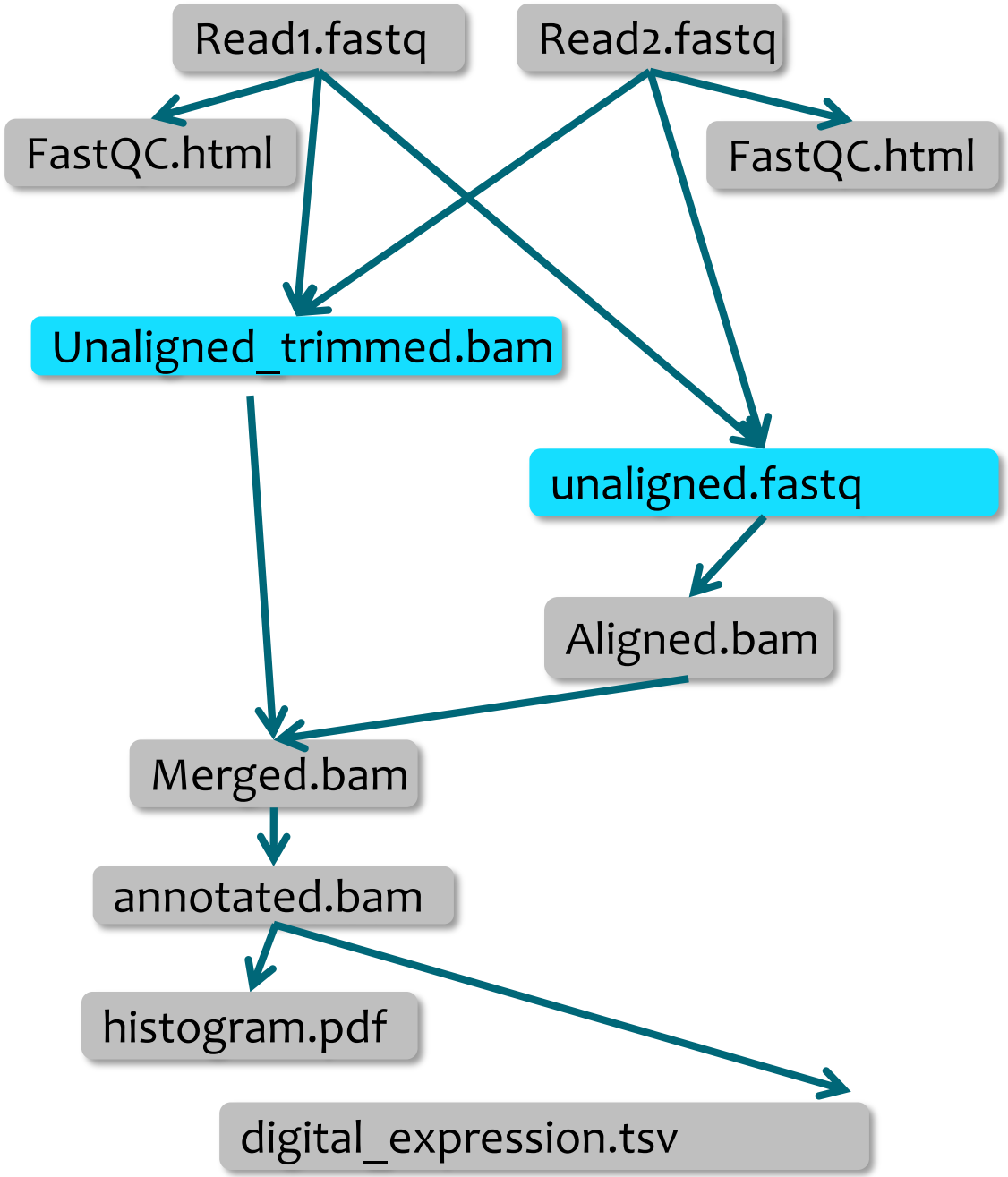
Why FASTQ-BAM-FASTQ-BAM?

-FASTQ format cannot hold the information about the cellular and molecular barcodes. BAM format has tag fields which can be used to hold this information

-In BAM format we can also do some trimming and filtering for the reads.

-However, the aligners take as input only FASTQ format, which is why we need to transform the trimmed & filtered BAM back to FASTQ format.

-Later on, after alignment, we merge the two!



Import files

Quality control / FASTQC

Preprocess:

- FASTQ to BAM
- Tag BAM
- Filter and trim BAM
- BAM to FASTQ
- Filter too short reads

-filter bad quality barcodes
-trim adapter sequences
-trim polyA tails

Aligner (STAR, HISAT)

Merge BAM

Combines the barcode tags and the alignment info

Tag read with gene exon

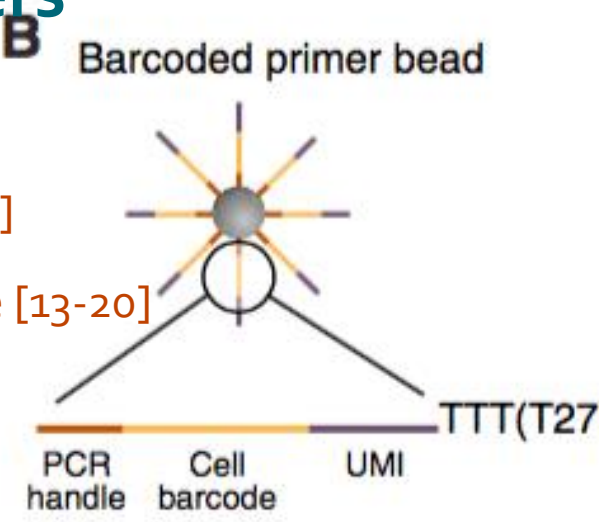
BAM tag histogram

Detect bead synthesis errors

Digital expression

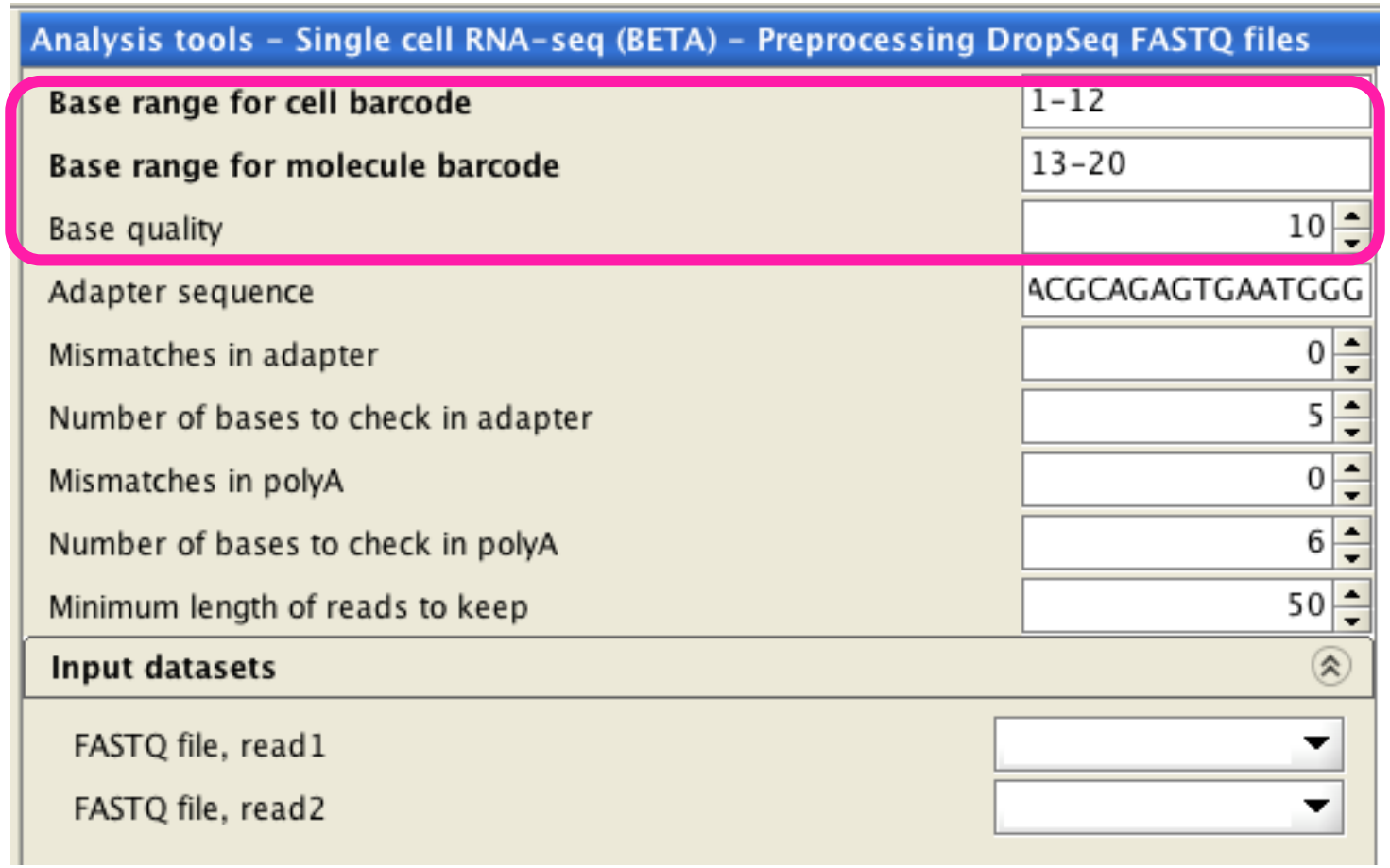
Preprocessing single cell DropSeq FASTQ files, parameters

- Convert the FASTQ files into an unaligned BAM file
- Tag the reads in the BAM file with the cellular and molecular barcodes
  - Base range for cell barcode [1-12]
  - Base range for molecule barcode [13-20]
  - Base quality [10]
- Filter and trim the reads in the BAM file
  - Adapter sequence [AAGCAGTGGTATCAACGCAGAGTGAATGGG]
  - Mismatches in adapter [0]
  - Number of bases to check in adapter [5]
  - Mismatches in polyA [0]
  - Number of bases to check in polyA [6]
- Convert the tagged and trimmed BAM file back into a FASTQ file for the alignment
- Filter out too short reads of the FASTQ file
  - Minimum length of reads to keep [50]



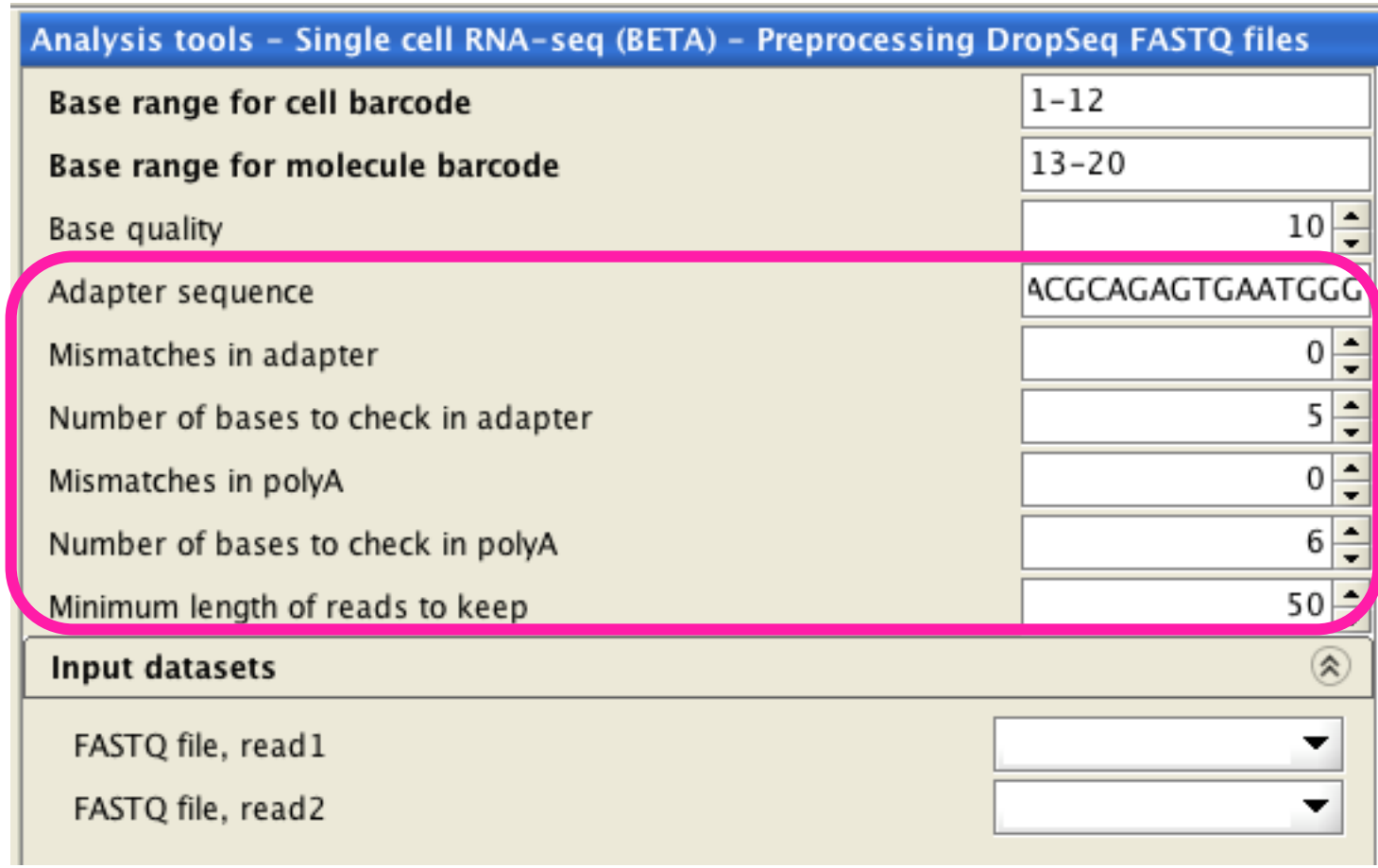
Preprocessing DropSeq FASTQ files –Tagging reads

- BAM tags
  - XM = molecular barcode
  - XC = cellular barcode
  - XQ = number of bases that fall below quality threshold
- After this, we can forget the barcode read
- You need to tell the tool which bases correspond to which barcode



Preprocessing DropSeq FASTQ files –Trimming and filtering

- **Quality**: filter out reads where more than 1 base have poor quality
- **Adapters**: Trim away any user determined sequences
  - SMART adapter as default
  - How many mismatches allowed (0)
  - How long stretch of the sequence there has to be at least (5)
- **polyA**: hard-clip polyA tails
  - How many A's need to be there before clipping happens (6)
  - mismatches allowed (0)
- **Minimum length**: filter out short reads from the FASTQ file

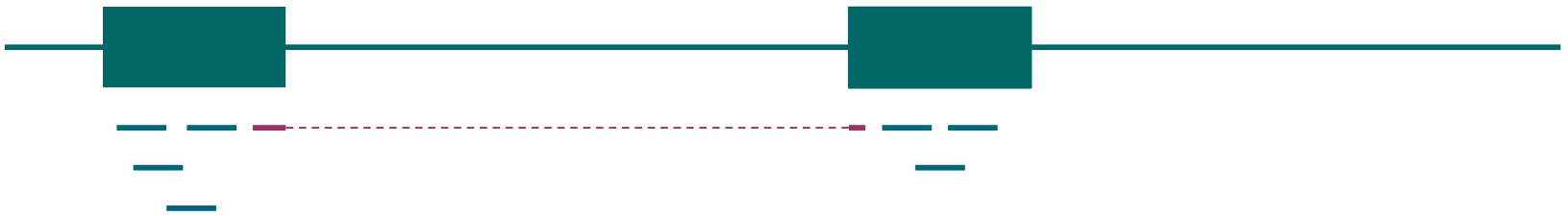


From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix

Alignment to reference genome

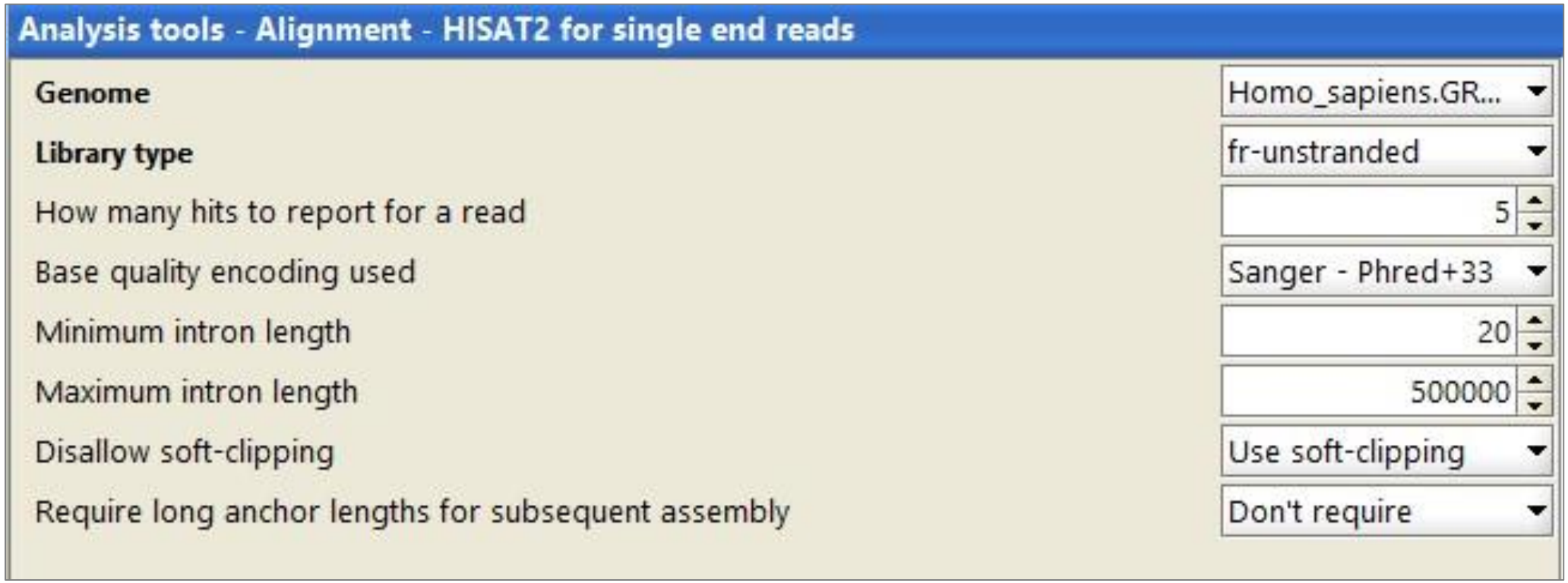
- Goal is to find out where a read originated from
  - Challenge: variants, sequencing errors, repetitive sequence
- Many organisms have introns, so RNA-seq reads map to genome non-contiguously → spliced alignments needed
  - But sequence signals at splice sites are limited and introns can be thousands of bases long
- Splice-aware aligners
  - HISAT2, TopHat
  - STAR



HISAT2

- HISAT = Hierarchical Indexing for Spliced Alignment of Transcripts
- Fast spliced aligner with low memory requirement
- Reference genome is indexed for fast searching
- Uses two types of indexes
  - One global index: used to anchor each alignment (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 reads with short anchors over splice sites)
- Uses splice site information found during the alignment of earlier reads in the same run

HISAT2 parameters

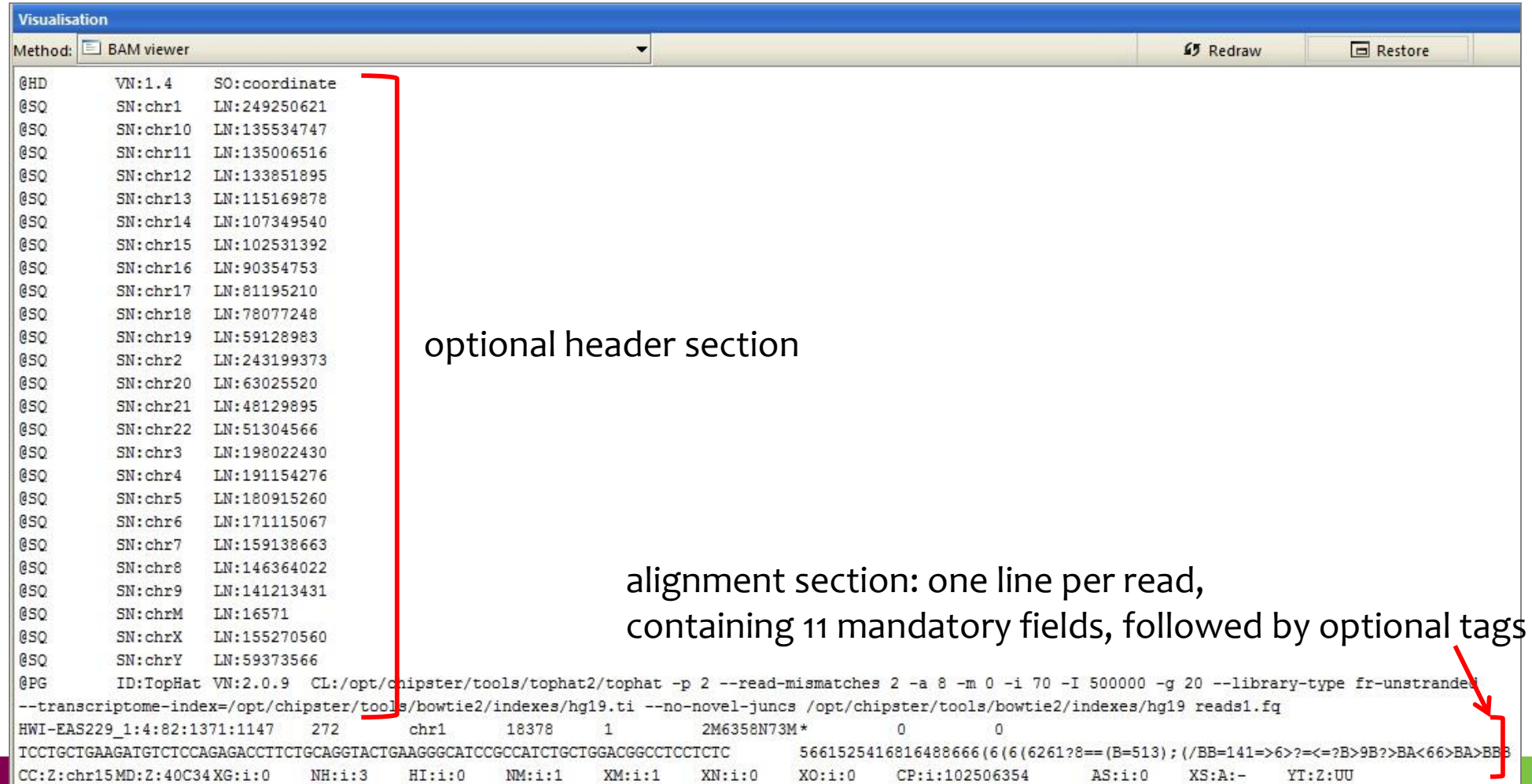


STAR

- **STAR (Spliced Transcripts Alignment to a Reference) uses a 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, for example GENCODE GTFs are recommended
- **Two log files**
  - Log_final.txt lists the percentage of uniquely mapped reads etc.
  - Log_progress.txt contains process summary

BAM file format for aligned reads

- SAM (Sequence Alignment/Map) is a tab-delimited text file containing aligned reads. BAM is a binary (and hence more compact) form of SAM.



optional header section

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

Fields in BAM file

- 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
  AGGGCCGATCTTGGTGCCATCCAGGGGGCCTCTACAAGGAT
  AATCTGACCTGCTGAAGATGTCTCCAGAGACCTT
- base qualities
  ECC@EEF@EB:EECFEECCCBEEEE;>5;2FBB@FBFEEFCF@
  FFFFCEFFFFEE>FFEFC=@A;@>1@6.+5/5
- tags MD:Z:75 NH:i:7 AS:i:-8 XS:A:-

BAM index file (.bai)

- BAM files can be sorted by chromosomal coordinates and indexed for efficient retrieval of reads for a given region.
- The index file must have a matching name. (e.g. reads.bam and reads.bam.bai)
- Genome browser requires both BAM and the index file.
- The alignment tools in Chipster automatically produce sorted and indexed BAMs.
- When you import BAM files, Chipster asks if you would like to preproces them (convert SAM to BAM, sort and index BAM).

Mapping quality

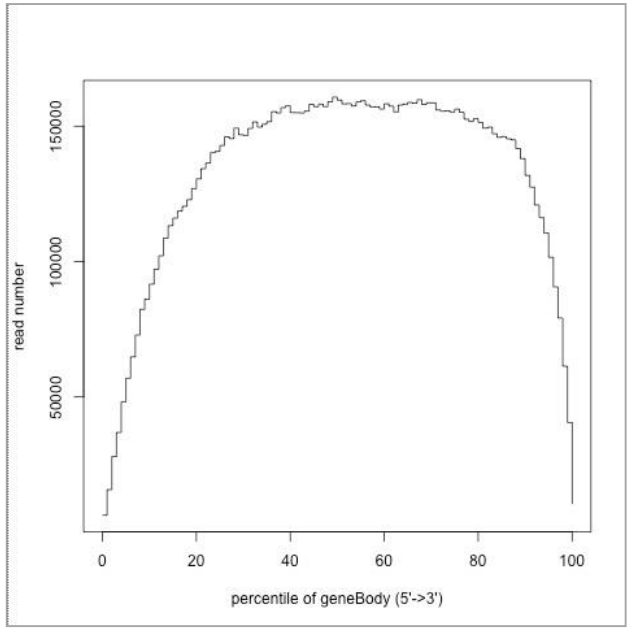
- Confidence in read's point of origin
- Depends on many things, including
  - uniqueness of the aligned region in the genome
  - length of alignment
  - number of mismatches and gaps

- **Should be 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/

Quality check after alignment with RseQC

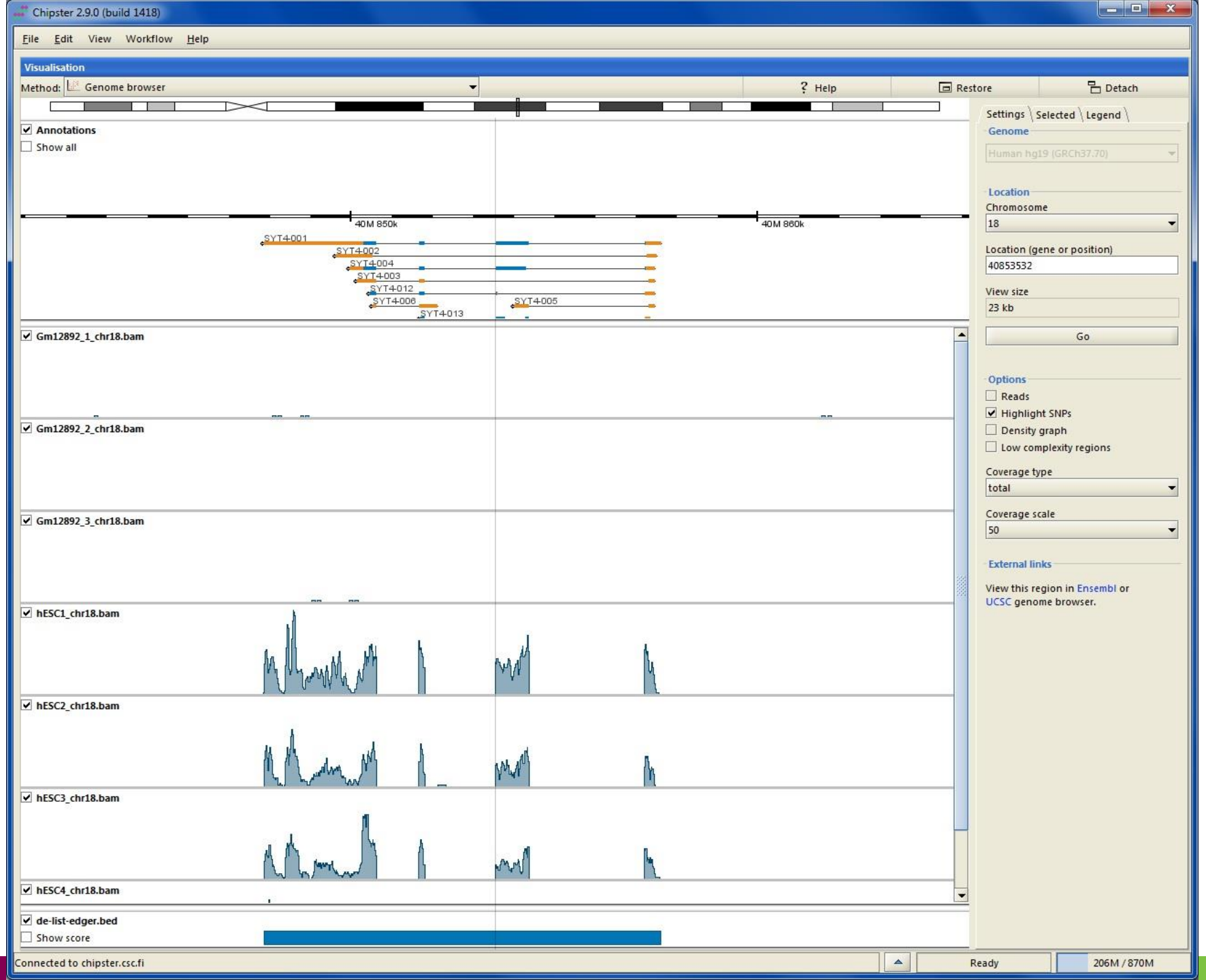
- Checks coverage uniformity, read distribution between different genomic regions, novelty of splice junctions, etc.
- Takes a BAM file and a BED file
  - Chipster has BED files available for several organisms
  - You can also use your own BED if you prefer

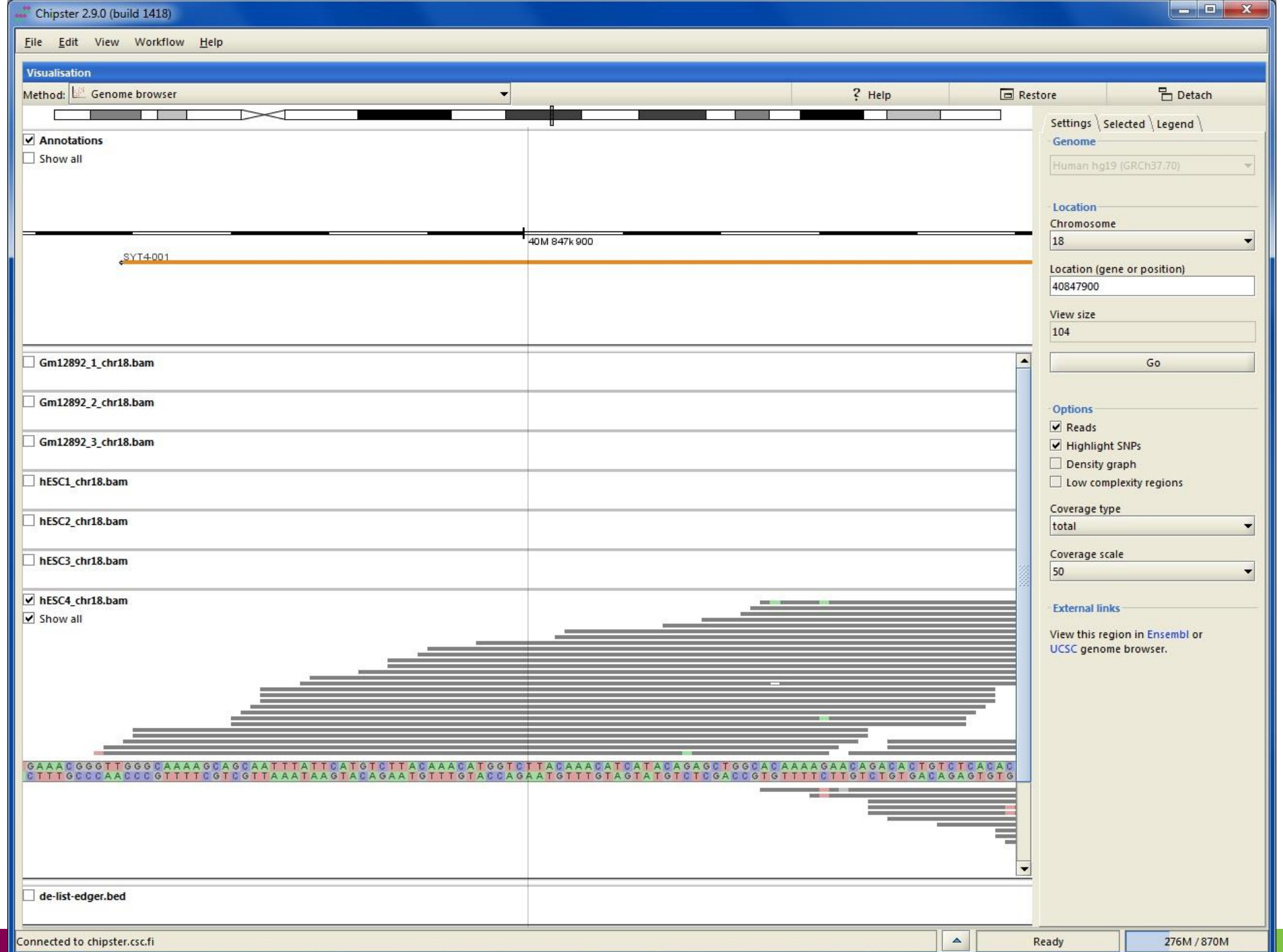



```
read_distribution:

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
=================================================================
```

Visualize alignments in genomic context: Chipster Genome Browser

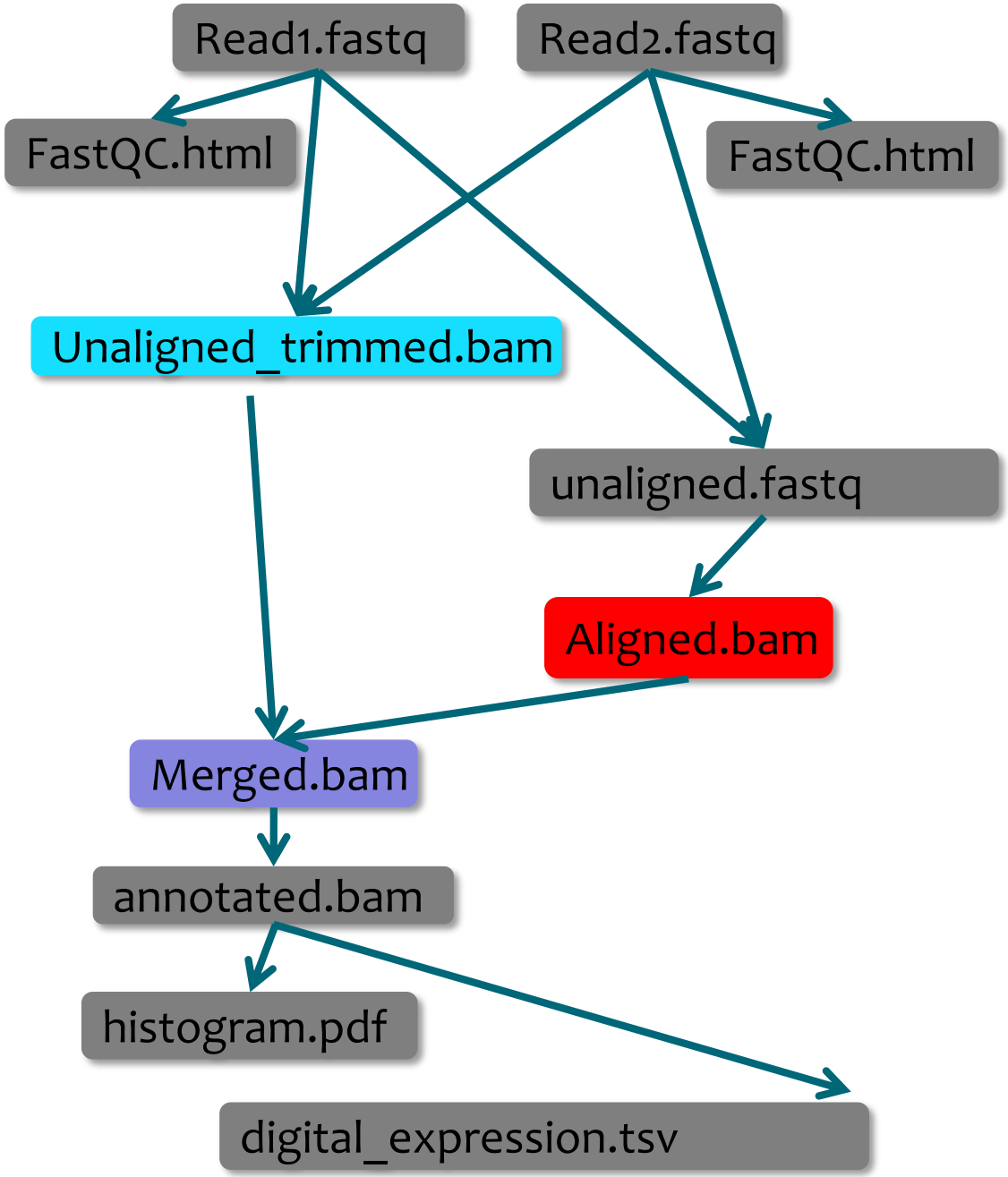
- Integrated with Chipster analysis environment
- Automatic coverage calculation (total and strand-specific)
- Zoom in to nucleotide level
- Highlight variants
- Jump to locations using BED, GTF, VCF and tsv files
- View details of selected BED, GTF and VCF features
- Several views (reads, coverage profile,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


Chipster 2.9.0 (build 1418)
File Edit View Workflow Help
Visualisation
Method: Genome browser
Help
Restore
Detach
Annotations
Show all
40M 850k
40M 860k
SYT4-001
SYT4-002
SYT4-004
SYT4-003
SYT4-012
SYT4-006
SYT4-005
SYT4-013
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
Settings
Selected
Legend
Genome
Human hg19 (GRCh37.70)
Location
Chromosome
18
Location (gene or position)
40853532
View size
23 kb
Go
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


Chipster 2.9.0 (build 1418)
File Edit View Workflow Help
Visualisation
Method: Genome browser
Help
Restore
Detach
Annotations
Show all
40M 847k 900
SYT4-001
Gm12892_1_chr18.bam
Gm12892_2_chr18.bam
Gm12892_3_chr18.bam
hESC1_chr18.bam
hESC2_chr18.bam
hESC3_chr18.bam
hESC4_chr18.bam
Show all
de-list-edger.bed
Settings
Selected
Legend
Genome
Human hg19 (GRCh37.70)
Location
Chromosome
18
Location (gene or position)
40847900
View size
104
Go
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
276M / 870M

From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix



Import files

Quality control / FASTQC

Preprocess:

- FASTQ to BAM
- Tag BAM
- Filter and trim BAM
- BAM to FASTQ
- Trim too short reads

-filter bad quality barcodes
-trim adapter sequences
-trim polyA tails

Aligner (STAR, HISAT)

Merge BAM

Combines the barcode tags and the alignment info

Tag read with gene exon

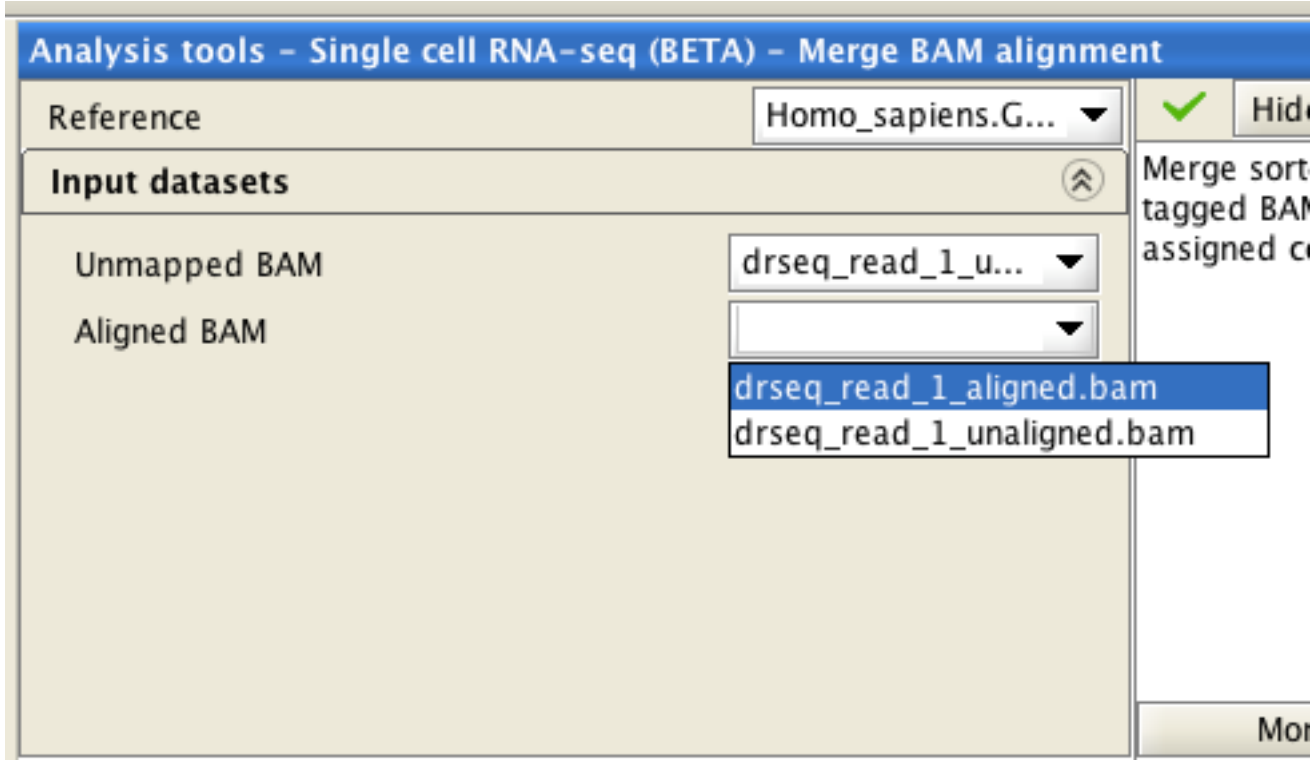
BAM tag histogram

Detect bead synthesis errors

Digital expression

Merge BAM files

- Why was this again…?
  - In the alignment, we lost the molecular and cellular tag information (because aligners only eat FASTQ format). We have that info in the unaligned.BAM file, so now we combine the two.
- Before merging, the tool sorts the files in queryname order
- Secondary alignments are ignored!
- Remember to…
  - Choose the reference
  - Make sure the files are correctly assigned!



From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix

Annotate

- GE-tag = name of the gene, if the read overlaps an exon
- XF-tag = location (intron, exon…)
- Choose the annotation file (GTF) OR use your own

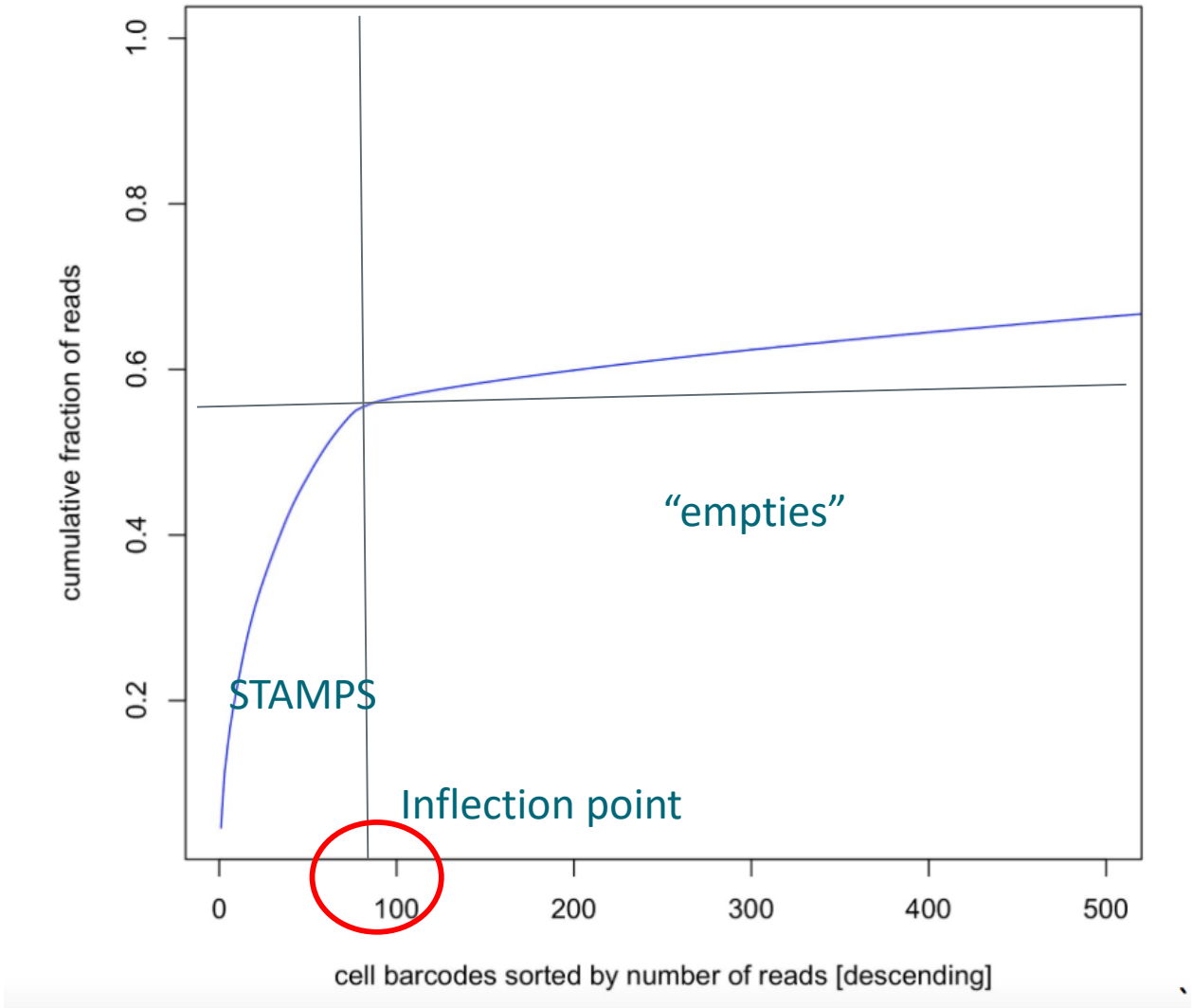
From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix

Estimate the number of usable cells

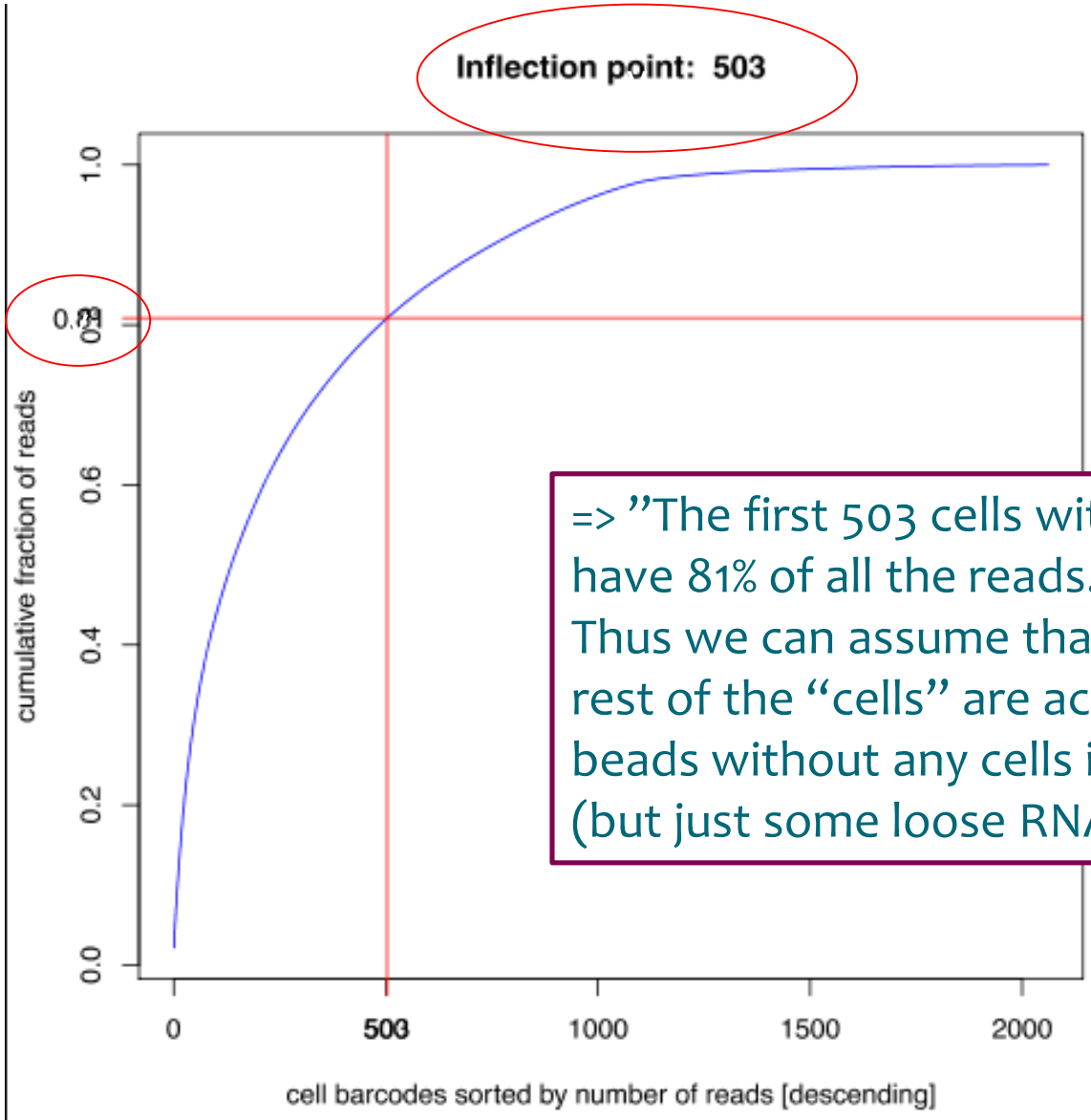
- How many cells do you want to keep?
- To estimate this: the inflection point
  1. extract the number of reads per cell (barcode)
  2. plot the cumulative distribution of reads
  3. select the "knee" of the distribution (knee = inflection point)

  - The number of **STAMPs** (=beads exposed to a cell in droplets): cell barcodes to the left of the inflection point
  - **Empties** (=beads only exposed to ambient RNA in droplets): to the right of the inflection point



Inflection point also computed using the inflection R -package

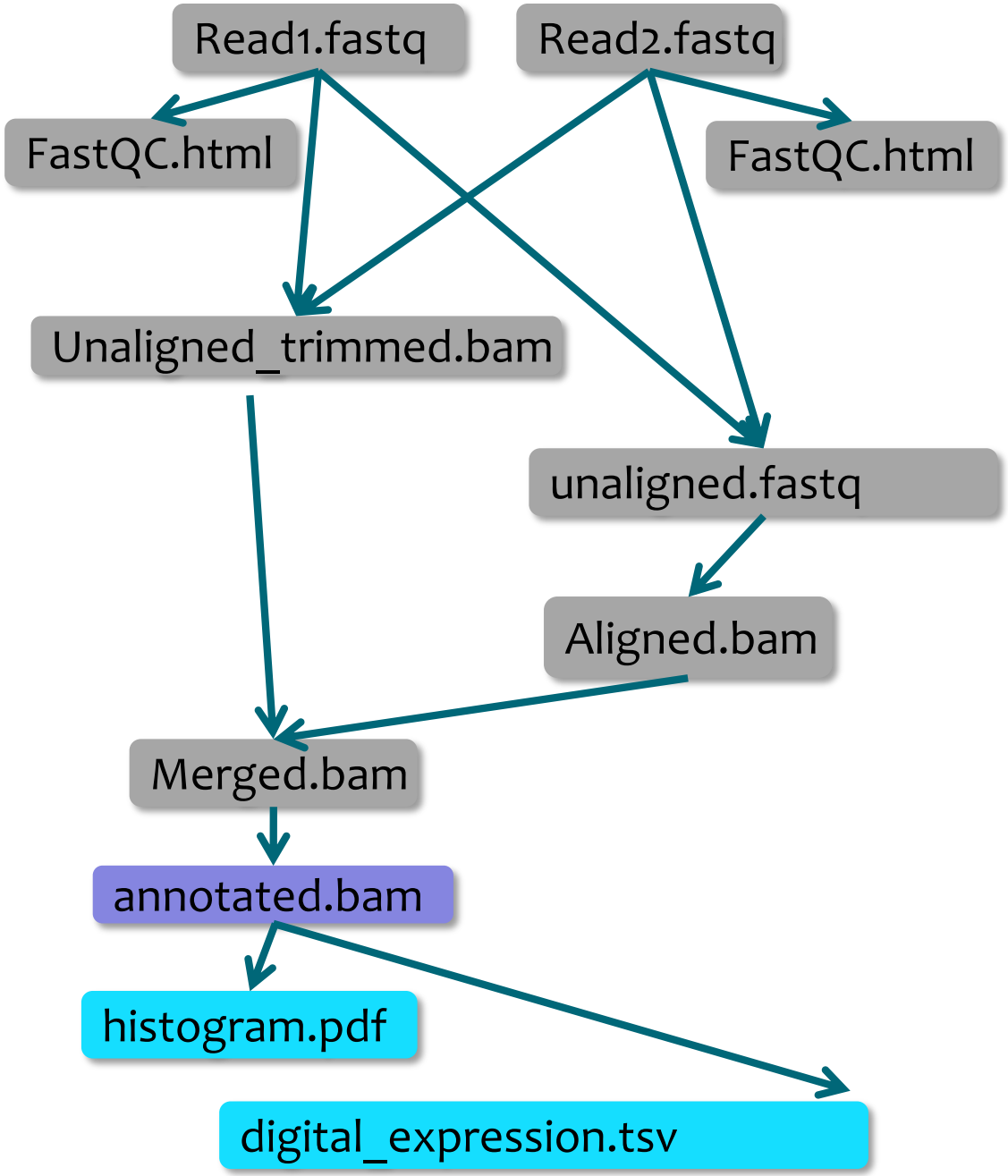
- Sometimes visual estimation can be tricky, so the tool also gives a numerical estimate
- Finds the inflection point using extreme distance estimator (ede) from *inflection* R-package
- Thank you **Dawit Yohannes**!



=> ”The first 503 cells with most reads have 81% of all the reads.”
Thus we can assume that the rest of the “cells” are actually beads without any cells in them (but just some loose RNA)

From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix



Import files

Quality control / FASTQC

Preprocess:

- FASTQ to BAM
- Tag BAM
- Filter and trim BAM
- BAM to FASTQ
- Trim too short reads

-filter bad quality barcodes
-trim adapter sequences
-trim polyA tails

Aligner (STAR, HISAT)

Merge BAM

To combine the barcode tags and the alignment info

Tag read with gene exon

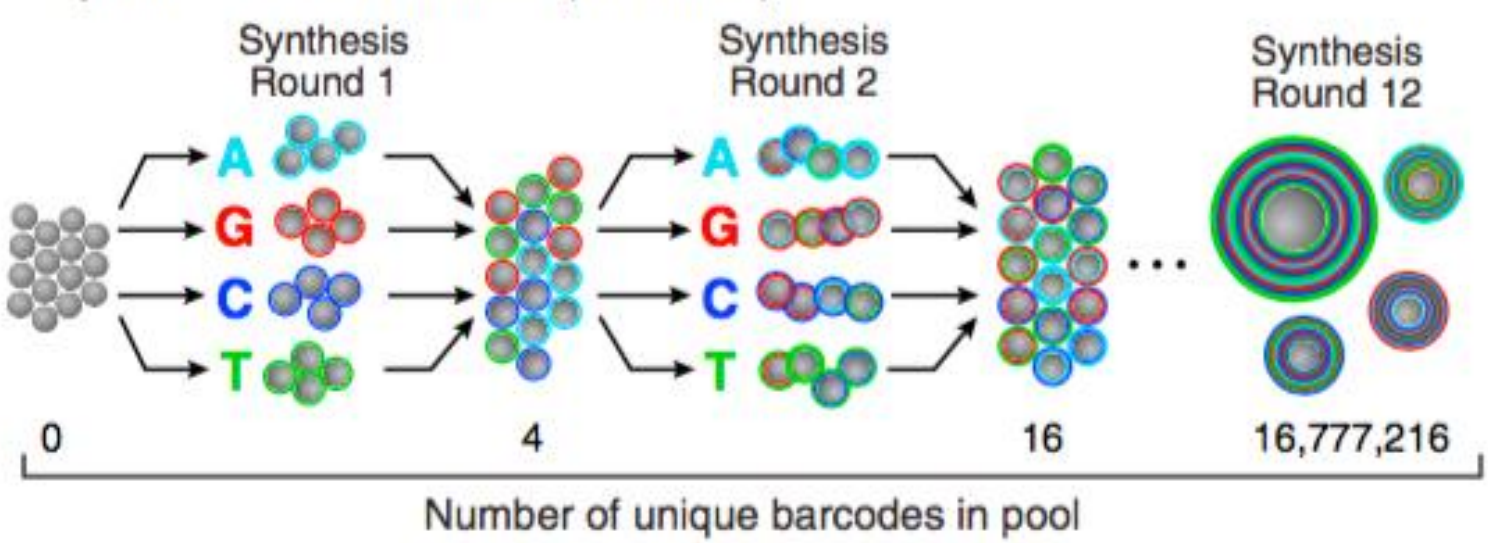
BAM tag histogram

Detect bead synthesis errors

Digital expression

Detect bead synthesis errors

ATTC GAGT TAT? CAGC GTAAT TTTT
Cell (12) UMI (8)



- Known problem with ChemGenes beads: a percentage of beads did not undergo all twelve split-and-pool bases.
  - A **mixed base** at base 12 (=actually 1st base of UMI) and a **fixed T** base at base 20 (=actually the 1st base of the polyT segment)
- DropSeq-tool *DetectBeadSynthesisErrors* fixes this:
  - The last base is trimmed off and all cell barcodes with identical sequence at the first 11 bases are merged
  - If any other UMI base is fixed, the reads with that cell barcode are discarded
- Parameter: number of barcodes on which to perform the correction
  - roughly 2 times the anticipated number cells (empirically found that this allows to recover nearly every defective cell barcode).

Error types

- SYNTHESIS_MISSING_BASE
  - 1 or more bases missing from cell barcode => T's at the end of UMIs
  - Fix: insert an "N" (reading frame fixed) and merge. If more than 1 missing, discard these reads
- SINGLE_UMI_ERROR
  - At each position of the UMIs, single base appears in >80% of the UMIs for that cell.
  - Fix: cell barcodes with this property are dropped
- PRIMER_MATCH
  - Same as with SINGLE_UMI_ERROR, + the UMI matches one of the PCR primers
  - Fix: these barcodes are dropped
- OTHER
  - UMIs are extremely skewed towards at least one base, but not at all 8 positions
  - Fix: these barcodes are dropped

Synthesis statistics

- **synthesis_stats.txt** contains a bunch of useful information:
  1. CELL_BARCODE the 12 base cell barcode
  2. NUM_UMI the number of total UMIs observed
  3. FIRST_BIASED_BASE the first base position where any bias is observed (**-1** for no detected bias)
  4. SYNTH_MISSING_BASE as 3 but specific to runs of T's at the end of the UMI
  5. ERROR_TYPE
  6. For bases 1-8 of the UMI, the observed base counts across all UMIs. This is a "|" delimited field, with counts of the A,C,G,T,N bases.

- **synthesis_stats.summary.txt** contains a histogram of the SYNTHESIS_MISSING_BASE errors*, as well as the counts of all other errors, the number of total barcodes evaluated, and the number of barcodes ignored.

*1 or more bases missing from cell barcode => T's at the end of UMIs

CELL_BARCODE	NUM_UMI	FIRST_BIASED_BASE	SYNTH_MISSING_BASE	ERROR_TYPE	BASE_1	BA
TCATTTAGTCGA	11832	-1	-1	NO ERROR	1564|1783|5254|3231|0	16
					A | C | G | T | N	

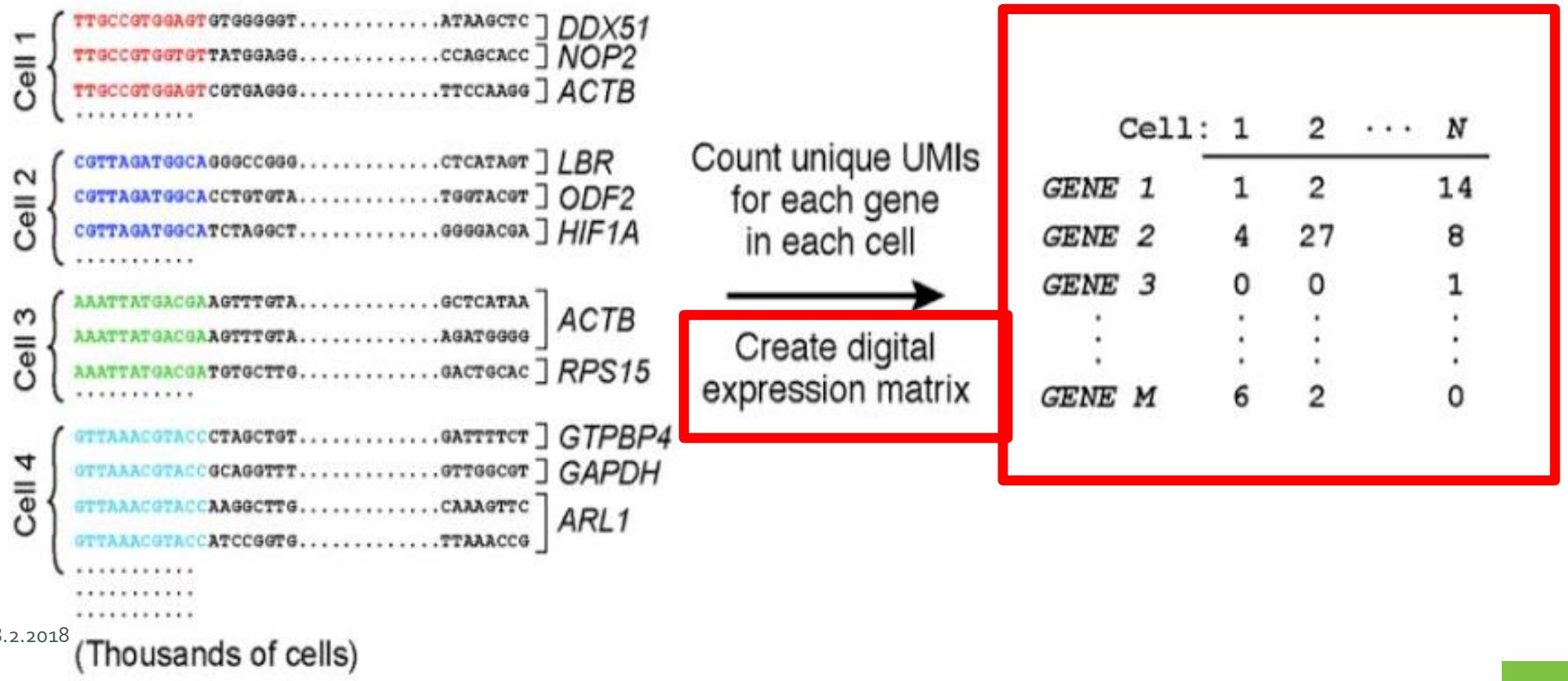
From FASTQ to expression matrix –preprocessing of DropSeq data

- Quality control of raw reads
- Preprocessing: tagging with barcodes & filtering
- Alignment to reference genome
- Merge BAM files
- Annotate with gene names
- Estimate the number of usable cells
- Detect bead synthesis errors
- Generate digital expression matrix

Digital expression matrix

- To digitally count gene transcripts:
  1. a list of HQ UMIs in each gene, within each cell, is assembled
  2. UMIs within edit distance = 1 are merged together
  3. The total number of unique UMI sequences is counted

     => this number is reported as the number of transcripts of that gene for a given cell



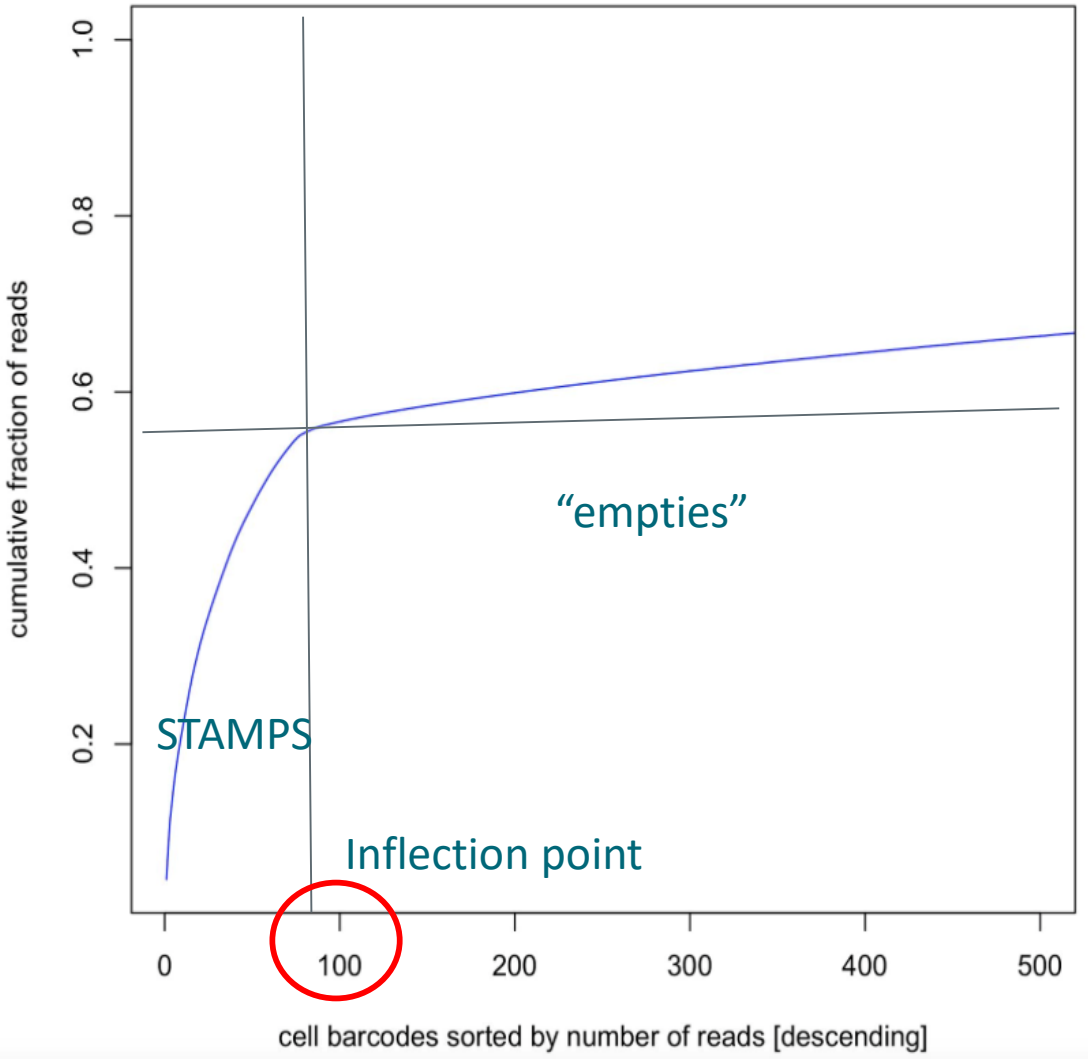
Filtering the data for DGE

NOTE: ”read” is a molecule here, which may or may not have the same/almost same UMI as another molecule

- Why don’t we just take all the cells?
  - the aligned BAM can contain hundreds of thousands o cell barcodes
  - Some of them are “empties”
  - Some cell (barcode)s contain just handful of reads
  - It is painful to deal with a huge matrix.

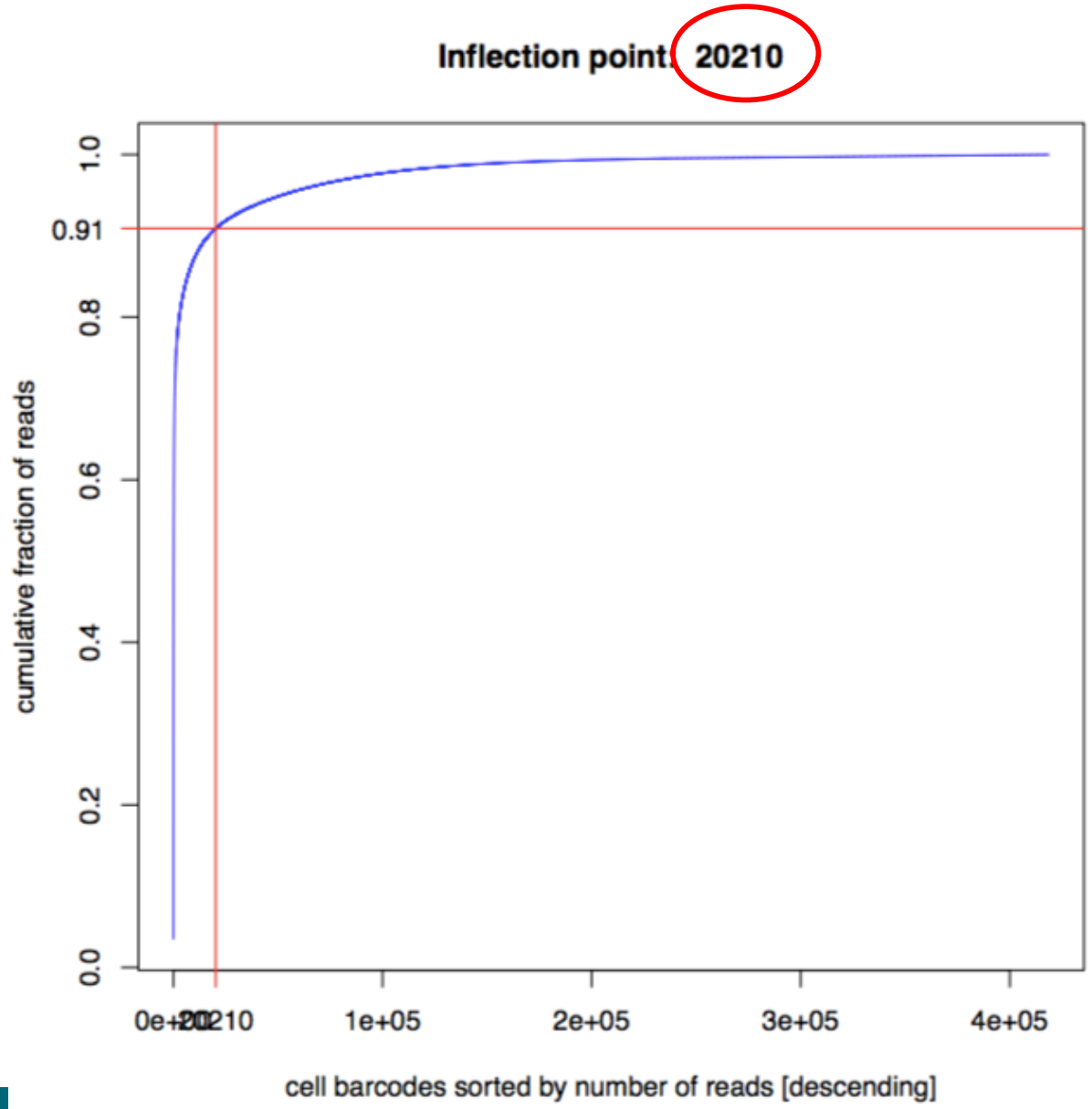
- Filtering based on:
  - Number of core barcodes (**top X cells** with most reads
  - Minimum number of expressed **genes** per cell

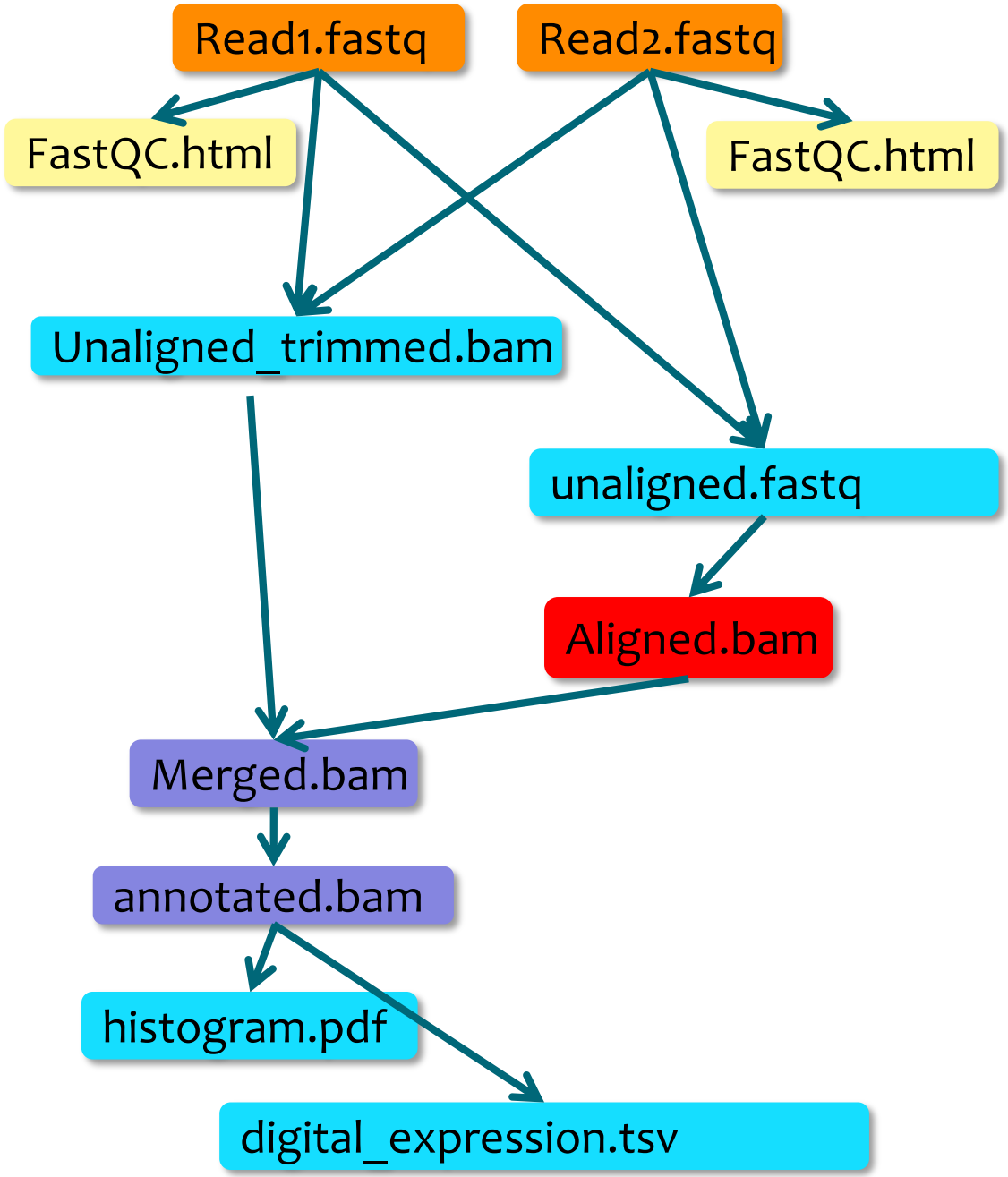
NOTE: You can always also choose to have a bigger number of cells and use that for further analysis (Seurat has it’s own filtering tools)



Filtering the data for DGE –huge number of reads

- What do you do if this happens?
  - Use the "minimum number of expressed genes per cell" as a requirement
  - Do you know how many cells to expect? Use that number!
  - Take a large number of cells and filter in the Seurat tools
- Why this happens?





Import files

Quality control / FASTQC

Preprocess:

- FASTQ to BAM
- Tag BAM
- Filter and trim BAM
- BAM to FASTQ
- Trim too short reads

-filter bad quality barcodes
-trim adapter sequences
-trim polyA tails

Aligner (STAR, HISAT)

Merge BAM

Combines the barcode tags and the alignment info

Tag read with gene exon

BAM tag histogram

Detect bead synthesis errors

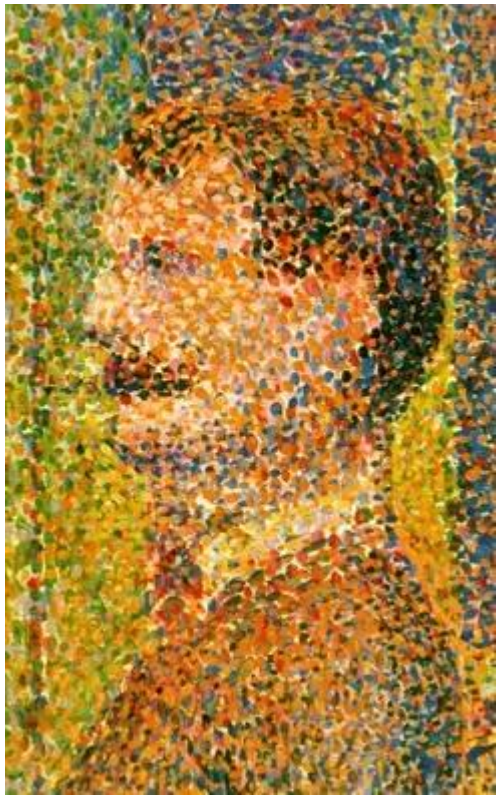
Digital expression

Clustering analysis with Seurat tools

Seurat

http://satijalab.org/seurat

- Seurat combines dimensionality reduction and graph-based partitioning algorithms for unsupervised clustering of single cells.
- The approach can be described briefly:
  1. Identification of highly variable genes
  2. Linear dimensionality reduction (PCA, principal component analysis) on variable genes
  3. Determine significant principal components
  4. Graph based clustering to classify distinct groups of cells
  5. Non-linear dimensional reduction (t-SNE, t-Distributed Stochastic Neighbor Embedding) for cluster visualization
  6. Marker gene discovery, visualization, and downstream analysis



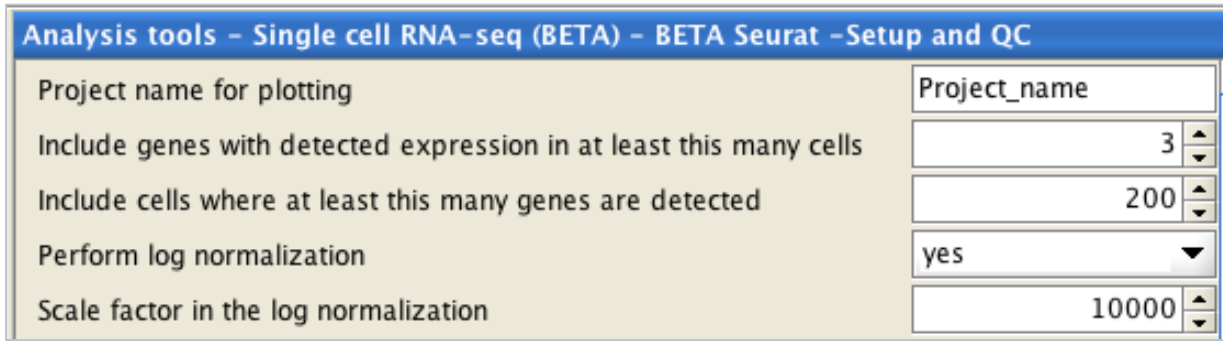
Clustering analysis with Seurat tools

- Setup & preprocessing (filtering, log normalization)
- Quality control
- Filter cells
- Regress unwanted sources of variation
- Detect variable genes
- Linear dimensional reduction (PCA)
- Determine statistically significant principle components (=identify the true dimensions of the data)
- Cluster the cells
- Non-linear dimensional reduction (tSNE)
- Find differentially expressed genes (biomarkers for the clusters)

- Setup & QC
- Filtering, regression & detection of variable genes
- Linear dimensional reduction (PCA)
- Clustering the cells
- Visualize biomarkers

Setting up Seurat object

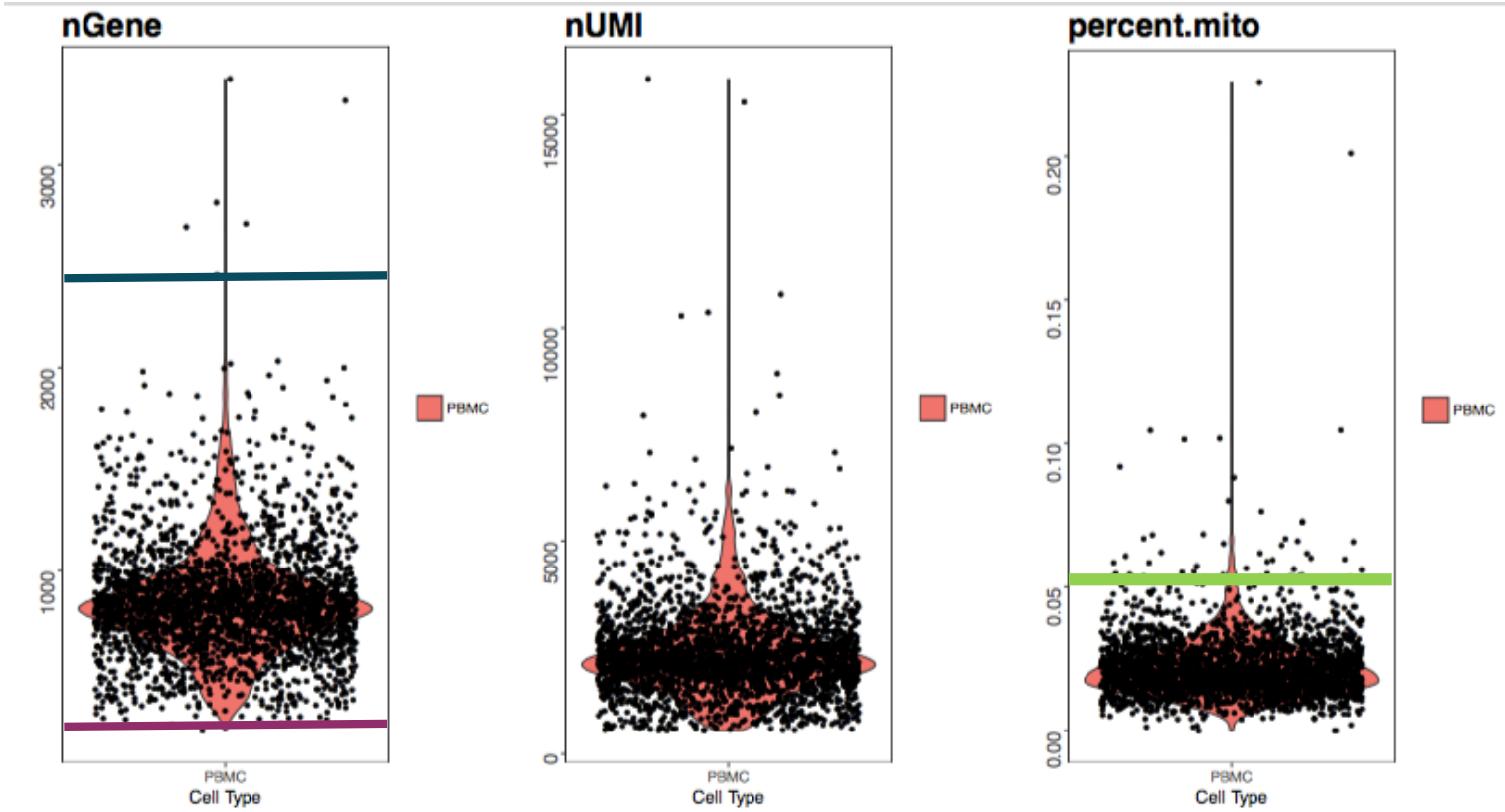
- In these tools, we are working with an R object (.Robj)
  - Can't be opened in Chipster, can be exported & imported to R
- Import one of the following
  - Tar package of three 10X Genomics output files
  - DGE matrix from the DropSeq tools
  - Check that the input file is correctly assigned!



- Log normalization
  - normalizes the gene expression measurements for each cell by the total expression, multiplies this by a scale factor (10,000 by default), and log transforms the result
- Filtering
  - Keep genes which are detected at least in X cells
  - Keep cells where at least Y genes are detected
- Give a name for the project (used in some plots)

Quality control & filtering for empties, multiplets and broken cells

- Empty (no cell in droplet) → low gene count (<200)
- Multiplet (more than one cell in droplet) →large gene count (>2500)
- Broken cell in droplet → large percentage of mitochondrial transcripts (>5%)



Clustering analysis with Seurat tools

- Setup & preprocessing (filtering, log normalization)
- Quality control
- Filter cells
- Regress unwanted sources of variation
- Detect variable genes
- Linear dimensional reduction (PCA)
- Determine statistically significant principle components (=identify the true dimensions of the data)
- Cluster the cells
- Non-linear dimensional reduction (tSNE)
- Find differentially expressed genes (biomarkers for the clusters)

- Setup & QC
- Filtering, regression & detection of variable genes
- Linear dimensional reduction (PCA)
- Clustering the cells
- Visualize biomarkers

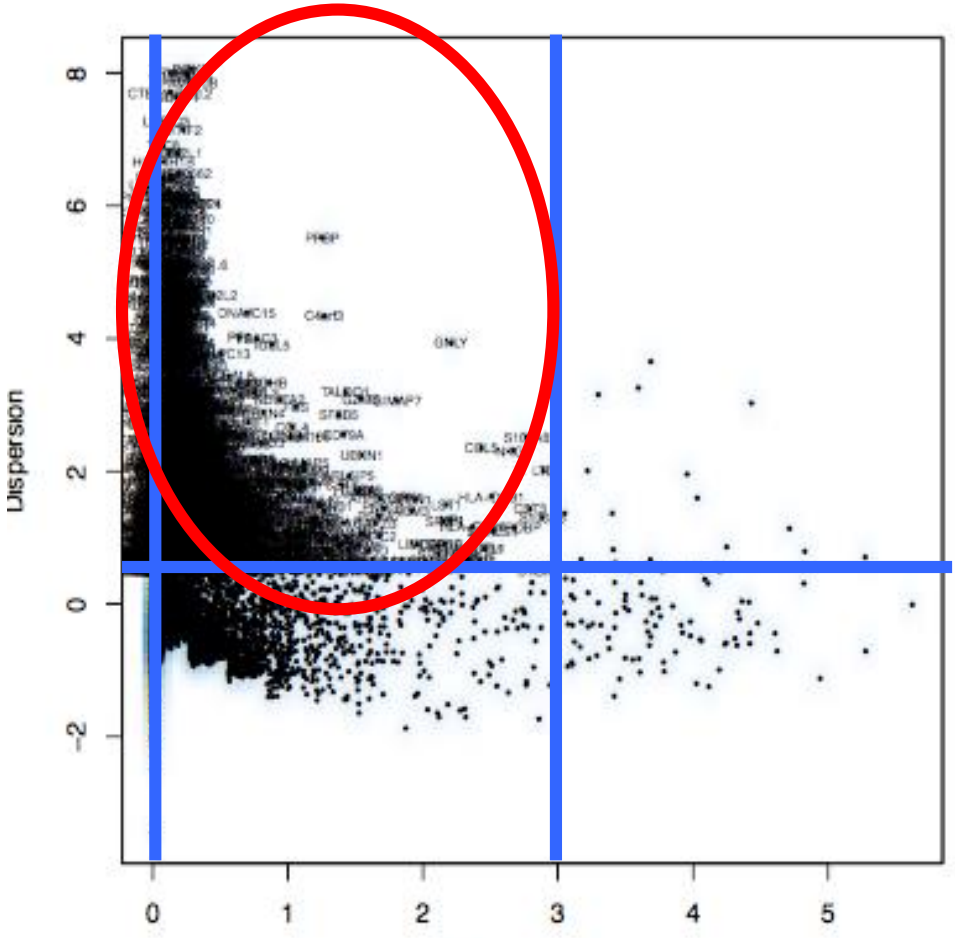
Remove unwanted sources of variation

- Single cell data typically contains 'uninteresting' variation
  - technical noise
  - batch effects
  - cell cycle stage, etc
- Removing this variation improves downstream analysis
- Seurat constructs linear models to predict gene expression based on user-defined variables
  - batch, cell alignment rate, no of detected molecules per cell, mitochondrial transcript percentage
  - Seurat regresses the given variables individually against each gene, and the resulting residuals are scaled
  - scaled z-scored residuals of these models are used for dimensionality reduction and clustering
  - Chipster currently uses the no of detected molecules per cell and mitochondrial transcript percentage

Detection of variable genes

- Downstream analysis focuses on highly variable genes
- Seurat finds them in the following way
  1. calculate the average expression and dispersion for each gene
  2. place genes into bins based on expression
  3. calculating a z-score for dispersion within each bin

- Set the parameters to mark visual outliers on the dispersion plot
  - exact parameter settings may vary based on the data type, heterogeneity in the sample, and normalization strategy
  - The default parameters are typical for UMI data that is normalized to a total of 10 000 molecules



Bottom cutoff on x-axis for identifying variable genes	0.0125
Top cutoff on x-axis for identifying variable genes	3.0
Bottom cutoff on y-axis for identifying variable genes	0.5

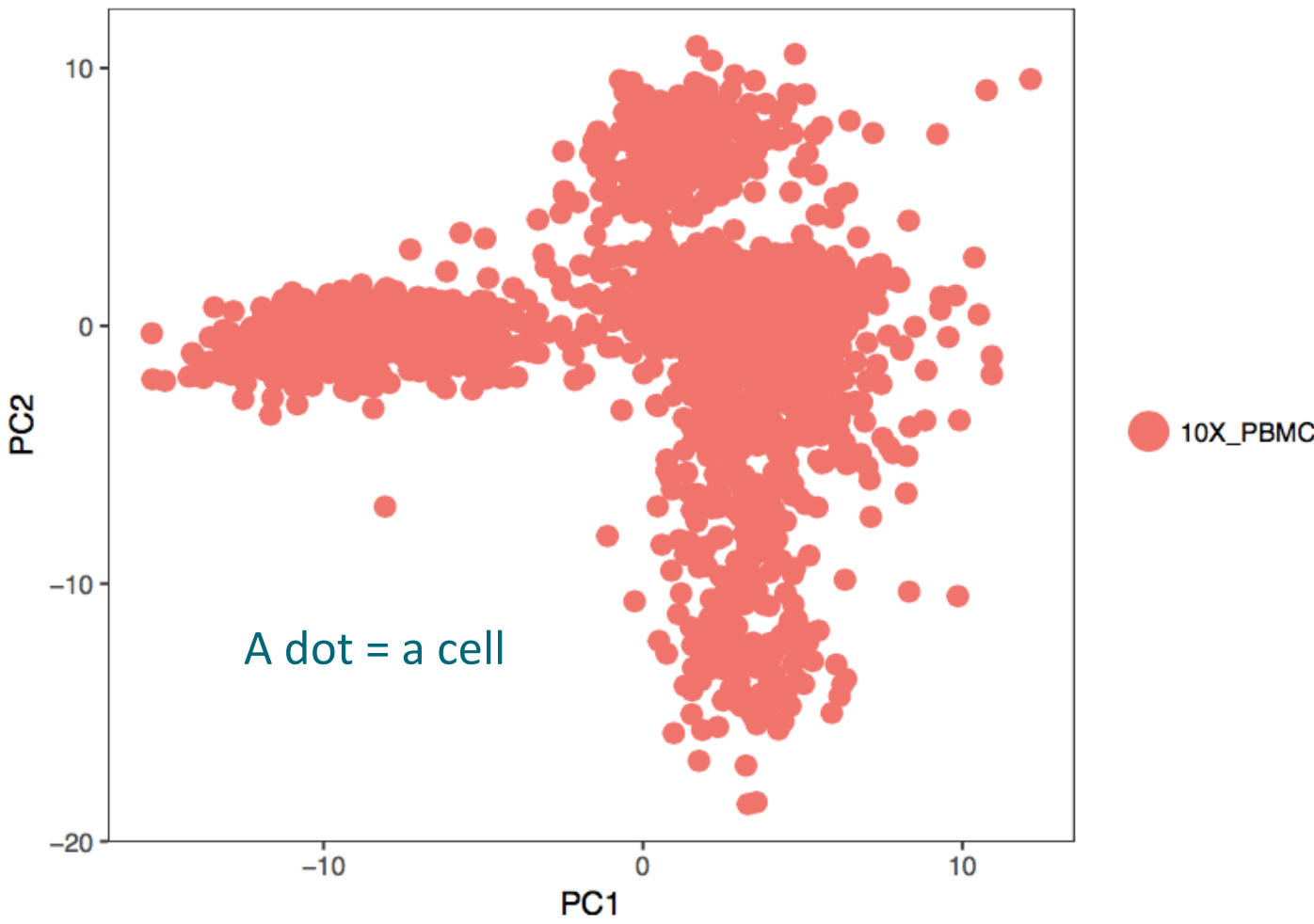
Clustering analysis with Seurat tools

- Setup & preprocessing (filtering, log normalization)
- Quality control
- Filter cells
- Regress unwanted sources of variation
- Detect variable genes
- Linear dimensional reduction (PCA)
- Determine statistically significant principle components (=identify the true dimensions of the data)
- Cluster the cells
- Non-linear dimensional reduction (tSNE)
- Find differentially expressed genes (biomarkers for the clusters)

- Setup & QC
- Filtering, regression & detection of variable genes
- Linear dimensional reduction (PCA)
- Clustering the cells
- Visualize biomarkers

Linear dimensionality reduction (PCA) on variable genes

- Principal Component Analysis = PCA
- Reduce the numerous, possibly correlating variables (=counts for each gene) into a smaller number of linearly uncorrelated dimensions (=principal components)
- Essentially, each PC represents a robust 'metagene' — a linear combination of hundreds to thousands of individual transcripts

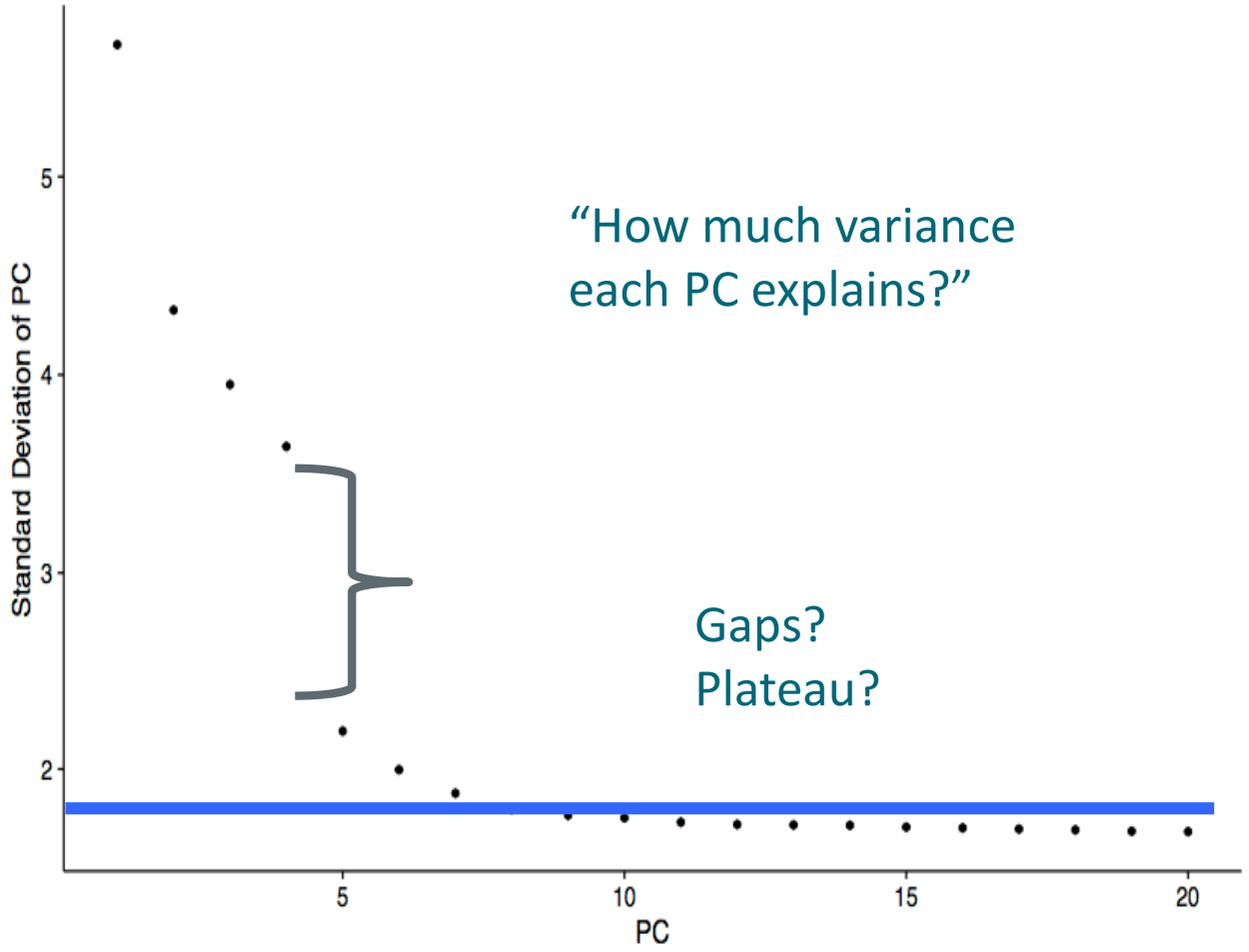


Determine significant principal components

- A key step to this clustering approach involves selecting a set of principal components (PCs) for downstream clustering analysis
- However, estimating the true dimensionality of a dataset is a challenging and common problem in machine learning.
- The tool provides a couple plots to aid in this:
  - Elbow plot
  - PCHeatmap

Determine significant principal components 1: Elbow plot

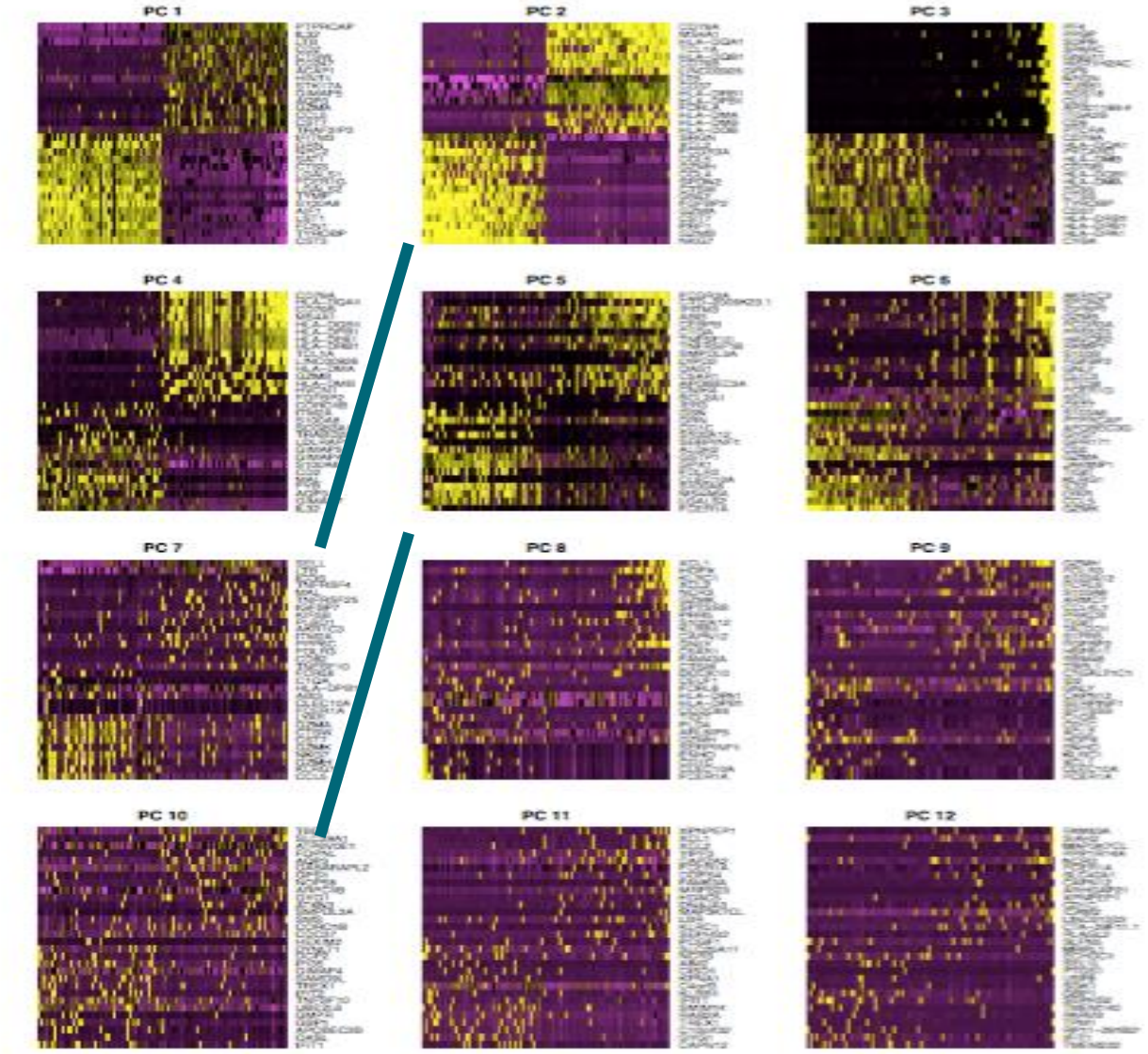
- The **elbow** in the plot tends to reflect a transition from informative PCs to those that explain comparatively little variance.



Determine significant principal components 2: PCHeatmap

- Displays the **extremes** across **both genes and cells**, and can be useful to help exclude PCs that may be driven primarily by ribosomal/mitochondrial or cell cycle genes.

"Is there still a difference between the extremes?"



Clustering analysis of with Seurat tools

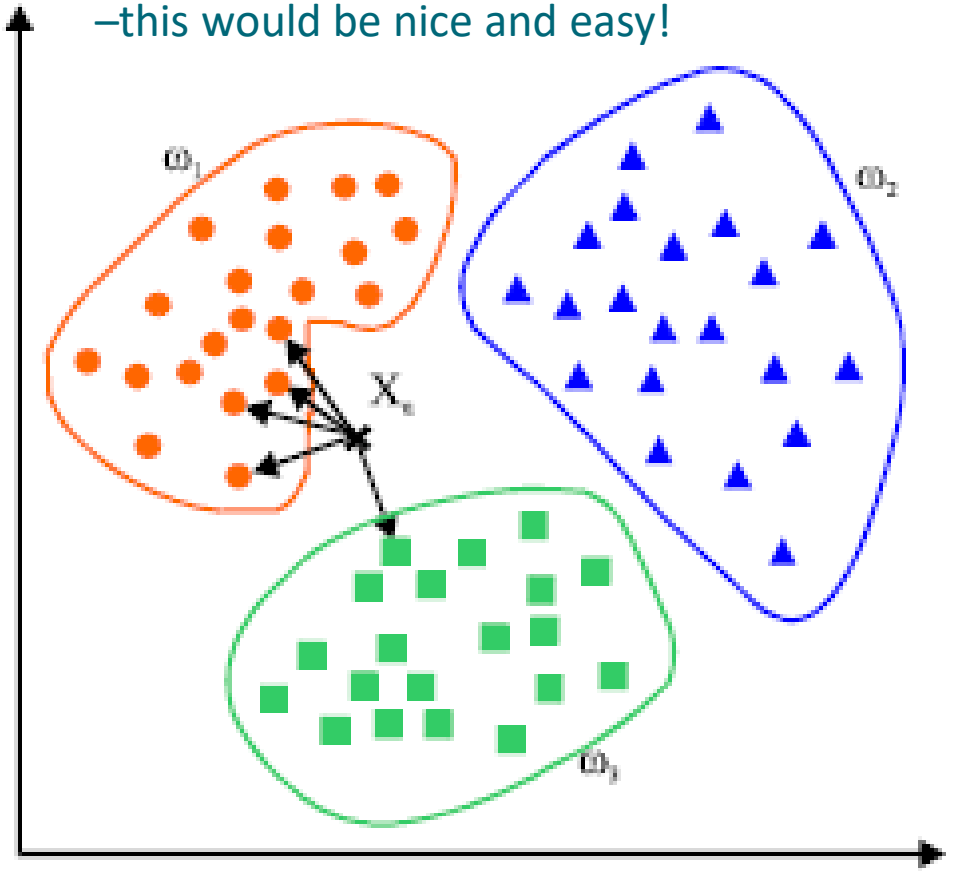
- Setup & preprocessing (filtering, log normalization)
- Quality control
- Filter cells
- Regress unwanted sources of variation
- Detect variable genes
- Linear dimensional reduction (PCA)
- Determine statistically significant principle components (=identify the true dimensions of the data)
- Cluster the cells
- Non-linear dimensional reduction (tSNE)
- Find differentially expressed genes (biomarkers for the clusters)

- Setup & QC
- Filtering, regression & detection of variable genes
- Linear dimensional reduction (PCA)
- Clustering the cells
- Visualize biomarkers

Clustering -why is it so tricky?

- Need to use **unsupervised** methods (we don't know beforehand, how many clusters there are)
- Our data is big and complex:
  - Lots of cells
  - Lots of dimensions (=genes)
  - Lots of noise (both technical and biological)
- …which is why:
  - the algorithm is a bit tricky, and
  - We reduce and select the dimensions to use in clustering (= we do the PCA and select X components to use, instead of using all thoudans of genes)

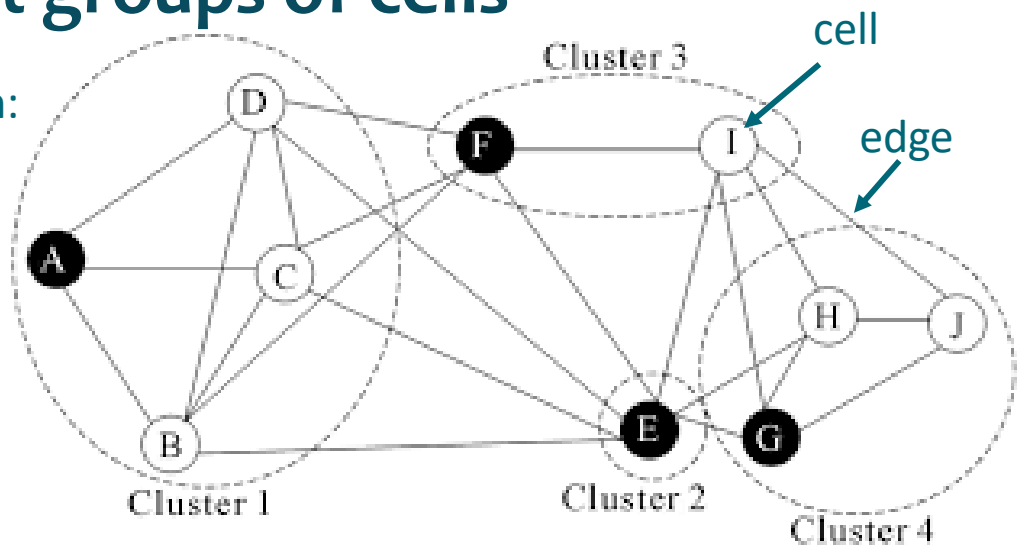
KNN in 2D with know clusters –this would be nice and easy!



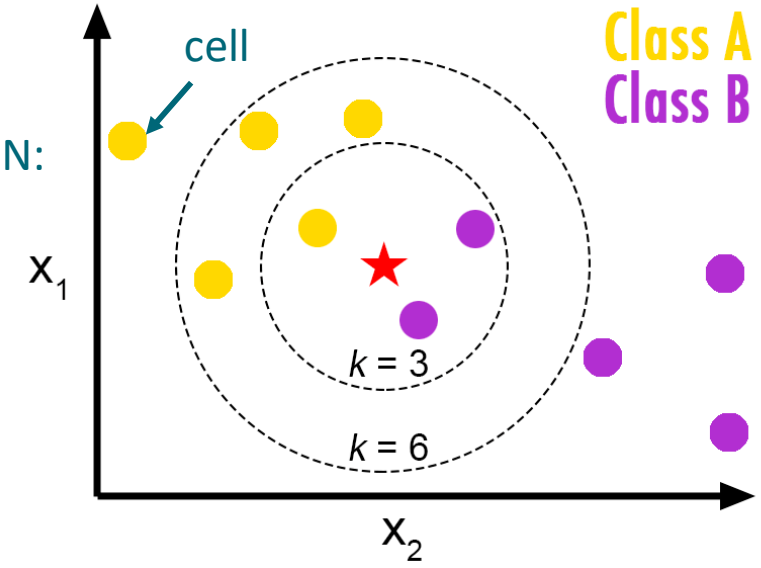
Graph based clustering to classify distinct groups of cells

- Seurat clustering = similar to
  - **SNN-Cliq** (C. Xu and Su, Bioinformatics 2015) and
  - **PhenoGraph** (Levine et al. Cell 2015)
- ...which are graph based methods:
  1. Identify **k-nearest neighbours** of each cell
     - Distance measure: Euclidean + Jaccard distance
  2. Calculate the number of **Shared Nearest Neighbours (SNN)** between each pair of cells
  3. Build the graph: add an **edge** between cells, if they have at least one SNN
  4. Clusters: group of cells with many edges between them
     - **Smart Local Moving algorithm** (SLM, Blondel et al., Journal of Statistical Mechanics)

Graph:



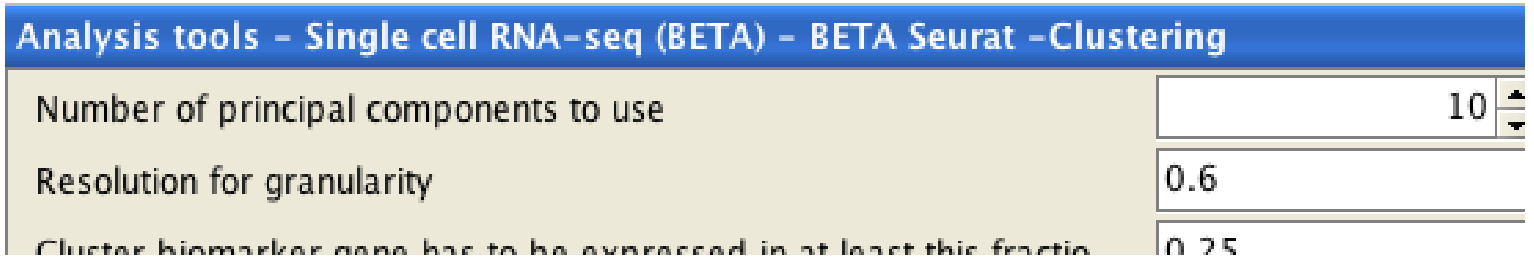
KNN:



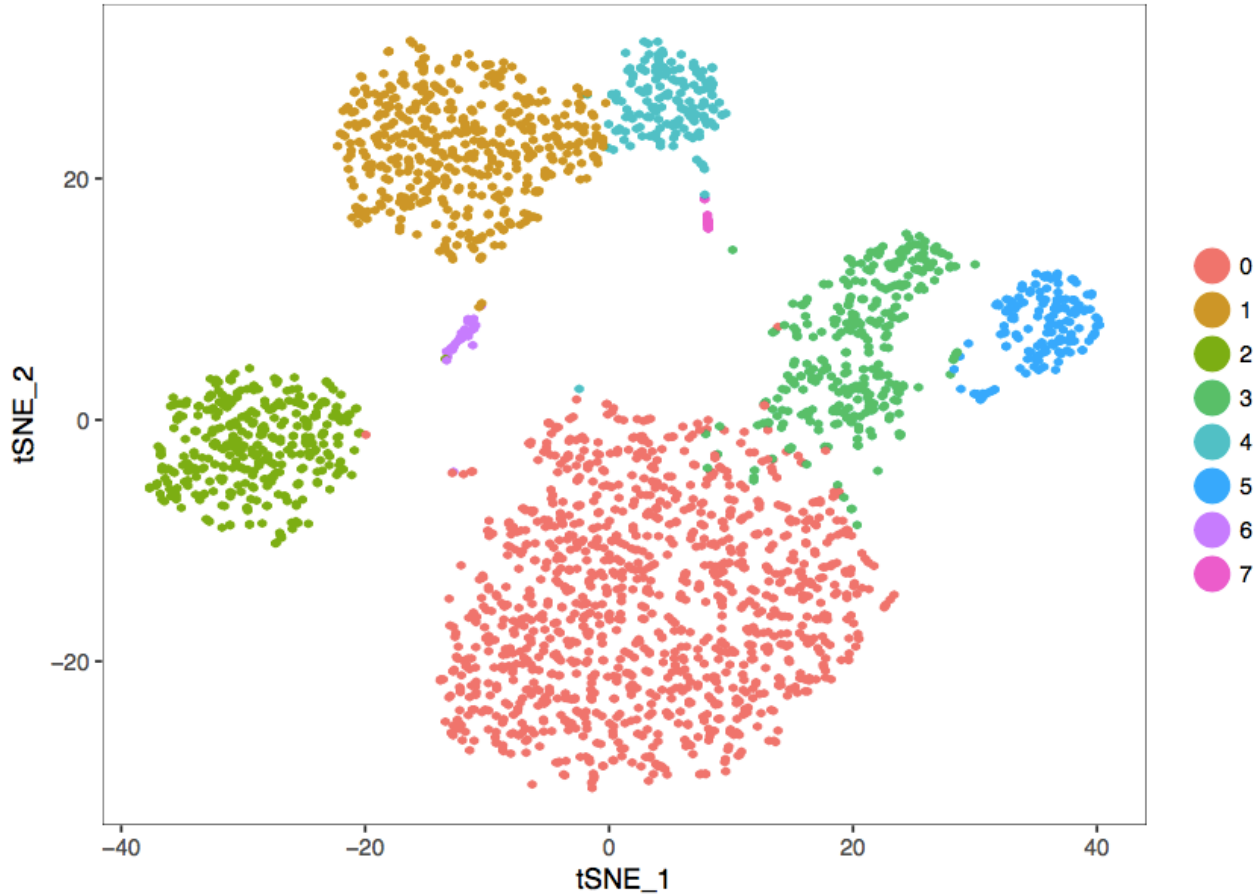
Clustering parameters

FindClusters(object, genes.use = NULL, reduction.type = "pca", **dims.use = NULL**, k.param = 30, k.scale = 25, plot.SNN = FALSE, prune.SNN = 1/15, print.output = TRUE, distance.matrix = NULL, save.SNN = FALSE, reuse.SNN = FALSE, force.recalc = FALSE, modularity.fxn = 1, **resolution = 0.8**, algorithm = 1, n.start = 100, n.iter = 10, random.seed = 0, temp.file.location = NULL)

- Lots of parameters!
- **Dims.use** = number of principal components
- **Resolution** = "granularity":
  - increased values lead to a greater number of clusters.
  - Values 0.6-1.2 typically returns good results for single cell datasets of around 3K cells
  - Optimal resolution often increases for larger datasets
  - (If you get very few or very many clusters, try adjusting)



Non-linear dimensional reduction (t-SNE) for cluster visualization

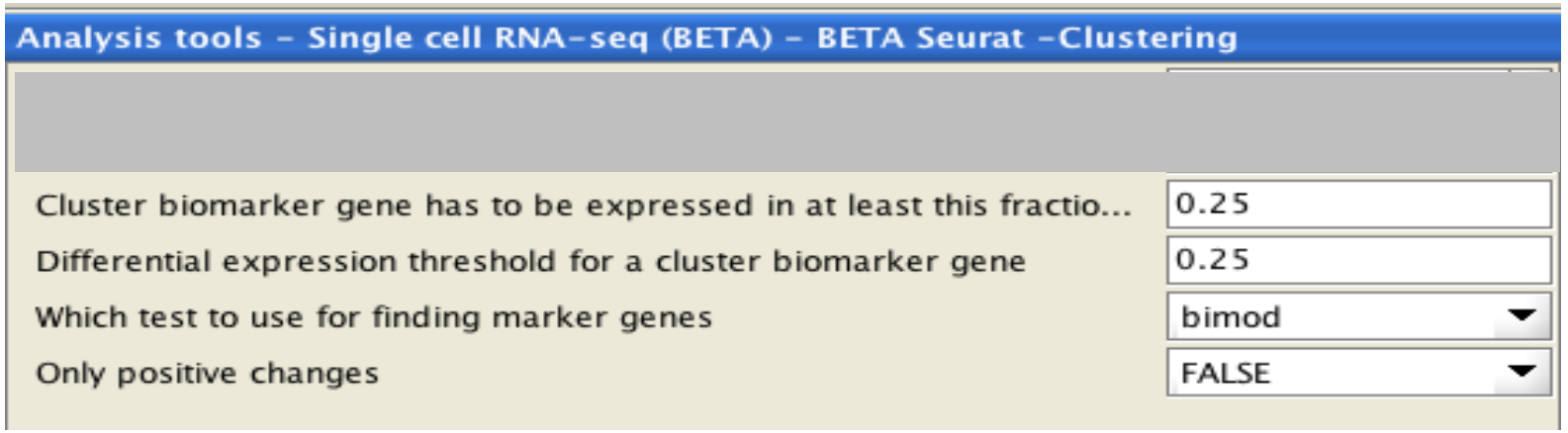


- T-SNE = t-distributed Stochastic Neighbor Embedding
  - Non-linear algorithm, different transformations to different regions
- Why do we now use t-SNE, why not the good old PCA?
  - We give the selected PCA "pseudo-genes" to t-SNE
  - T-SNE also reduces dimensions, but it is "more faithful to the original data"
  - PCA can find clusters too, but t-SNE does just that – it reduced the dimensions so that the clusters become visible
- Good text about readingT-SNE's: https://distill.pub/2016/misread-tsne/

Finding differentially expressed genes (biomarkers for the clusters)

- Parameters:
  - **min.pct** : requires a gene to be detected at least THIS minimum percentage in either of the two groups of cells (default: 0.25)
  - **thresh.test** : requires a gene to be differentially expressed (on average) by THIS amount between the two groups (default: 0.25)
  - You can set both of these to zero, but with a dramatic increase in time (this will test a large number of genes that are unlikely to be highly discriminatory)
  - Options for tests: bimod, roc, Students t-test, Tobit-test, Poisson, negative-binomial distribution

NOTE:
We are now comparing a cluster to all other cells.
So for example cluster 1 vs all others.



Markers for a particular cluster

- You can filter the resulting list to get only the biomarkers for a certain cluster:

Utilities / Filter table by column value

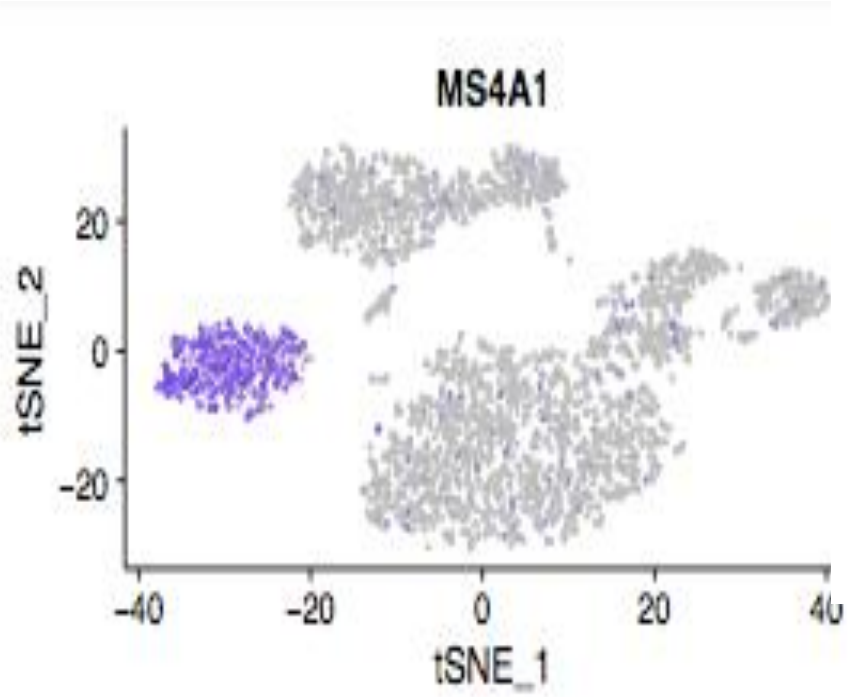
Column to filter by	cluster
Does the first column have a title	no
Cutoff	2.0
Filtering criteria	equal-to

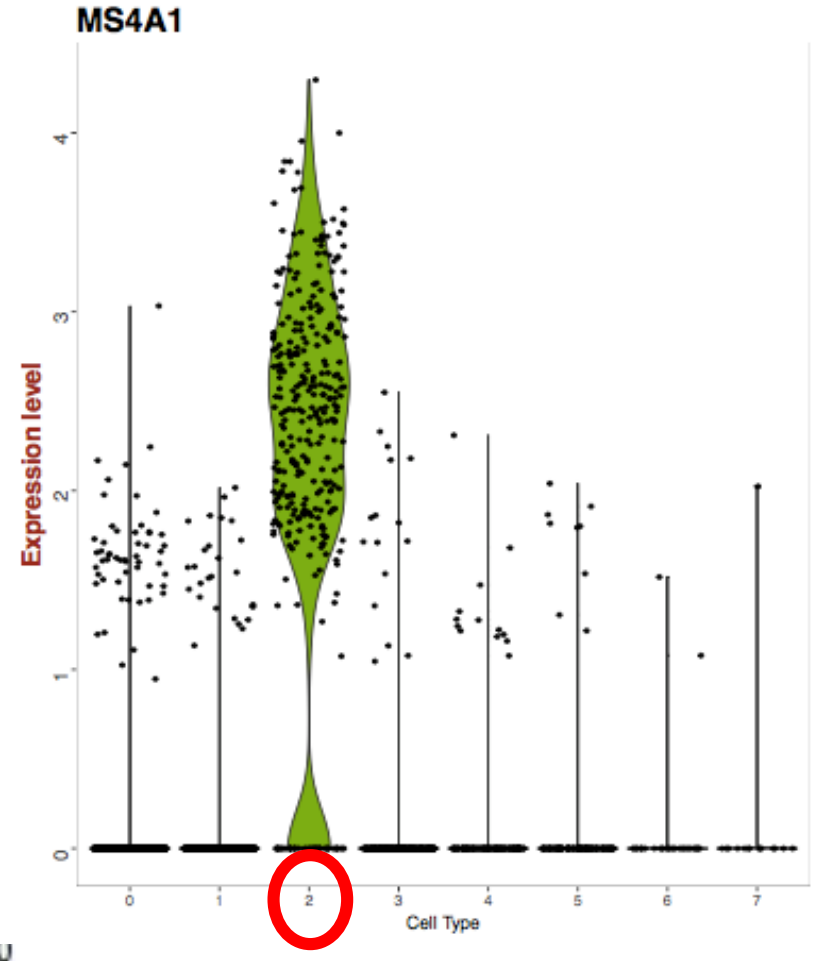
Showing 320 rows of 320 and all 7 columns

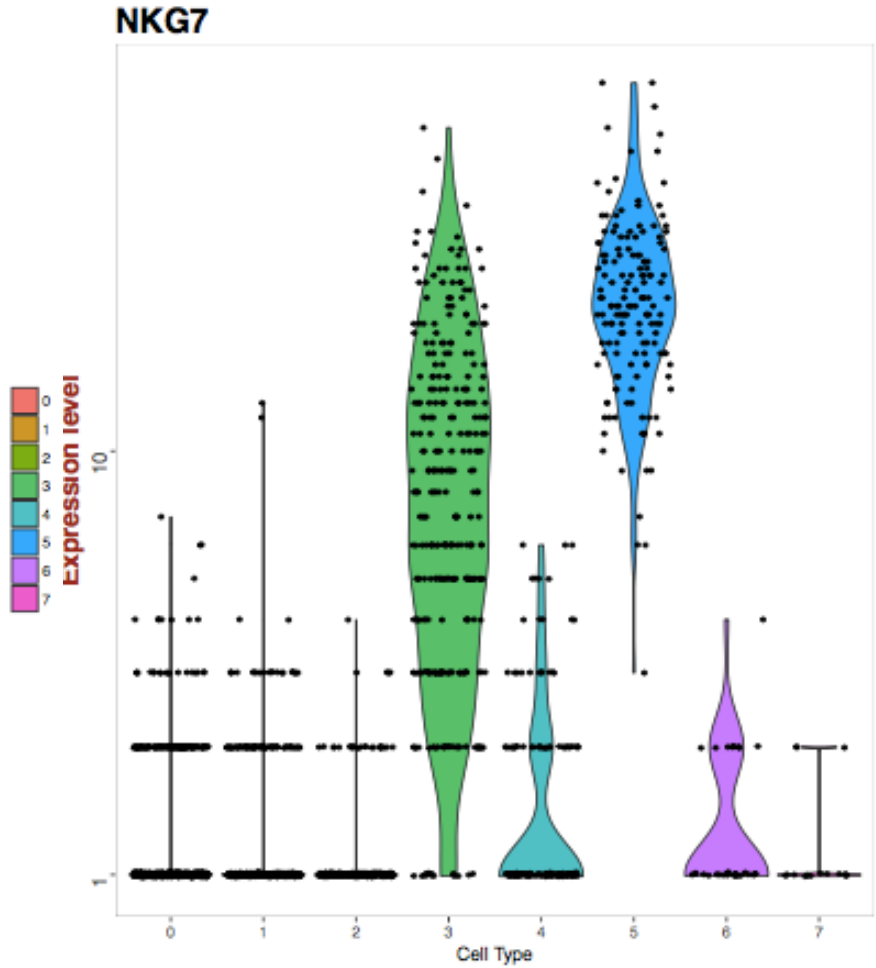
	p_val	avg_diff	pct.1	pct.2	cluster
TCL1A	1.030004e-191	2.4922363	0.624	0.022	2
HLA-DRB12	1.030181e-176	1.4341113	0.985	0.381	2
LINC00926	2.78283e-175	1.9663931	0.566	0.009	2
HLA-DPA12	6.494113e-166	1.3878677	0.98	0.421	2
VPREB3	1.41582e-150	1.6848535	0.493	0.006	2
S100A41	1.067153e-145	-2.110098	0.367	0.879	2
BANK1	1.547914e-111	1.4315893	0.481	0.037	2
HLA-DRB52	4.812796e-111	1.3080622	0.854	0.276	2
S100A61	9.253069e-108	-1.6521713	0.359	0.846	2

Visualize biomarkers

- Select a marker gene from the lists







Clustering analysis with Seurat tools

- Setup & preprocessing (filtering, log normalization)
- Quality control
- Filter cells
- Regress unwanted sources of variation
- Detect variable genes
- Linear dimensional reduction (PCA)
- Determine statistically significant principle components (=identify the true dimensions of the data)
- Cluster the cells
- Non-linear dimensional reduction (tSNE)
- Find differentially expressed genes (biomarkers for the clusters)

- Setup & QC
- Filtering, regression & detection of variable genes
- Linear dimensional reduction (PCA)
- Clustering the cells
- Visualize biomarkers