

# RNA-seq Quality Control

Before the analysis begins

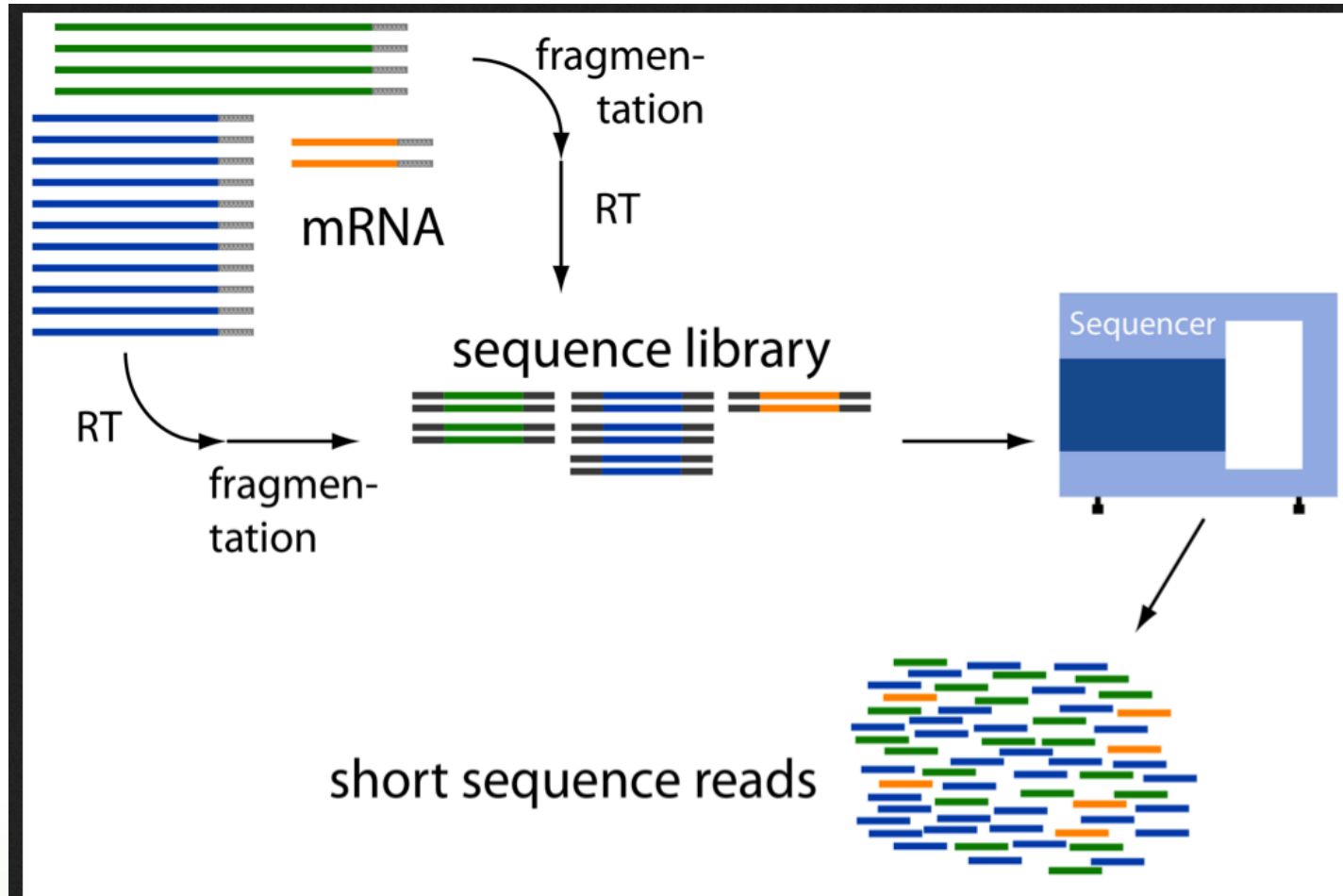
[johan.reimegard@scilifelab.se](mailto:johan.reimegard@scilifelab.se)

Enabler for Life Sciences

# Overview

- Introduction
- FastQC – read based QC
- RseQC – mapping based QC
- PCA
- Spike-in controls
- Experimental design

# RNA-seq libraries



What could go wrong?

# What could go wrong?

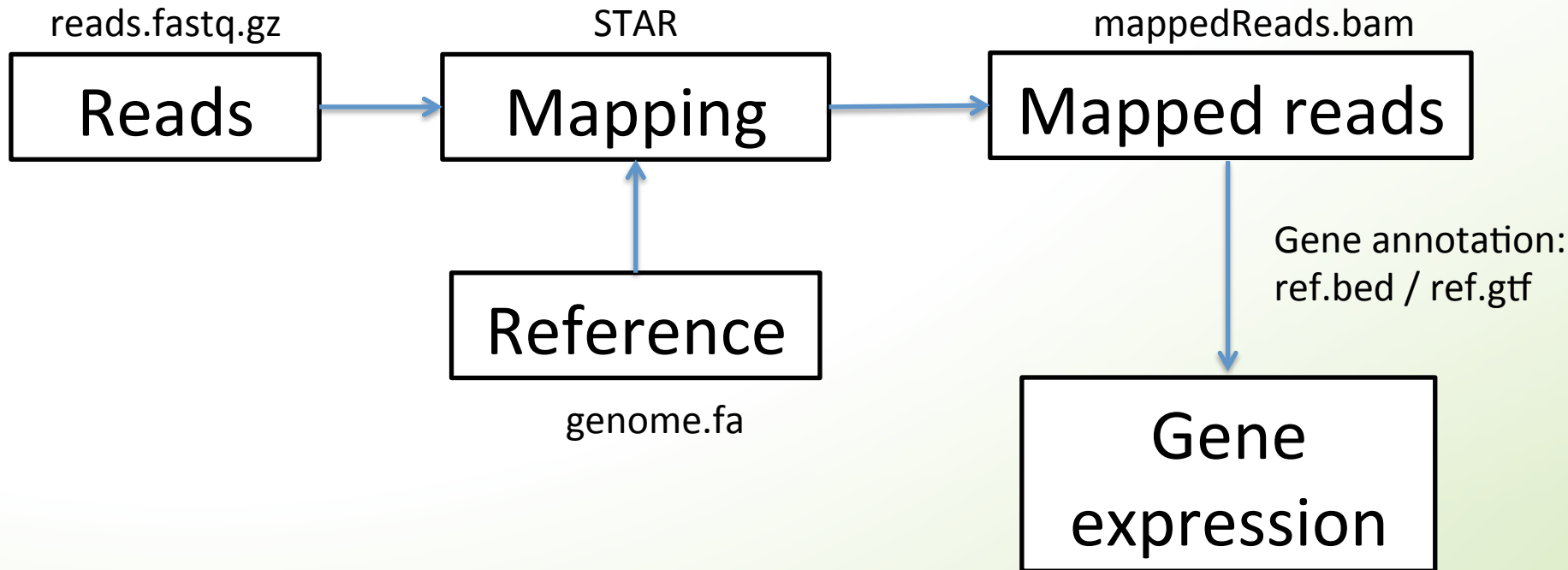
- RNA quality:
  - Degradation
  - Contaminations (pathogens or other sources)
  - GC-bias
  - Nuclear vs organellar reads
- Library prep:
  - Failed reactions
  - RNA / Adapter ratios – primer dimers
  - Clonal duplicates
  - Chimeric reads
  - Contaminations
- Sequencing:
  - Base calling errors
  - Uncalled bases
  - Low quality bases (3' end)
  - Contaminations
  - Sequence complexity

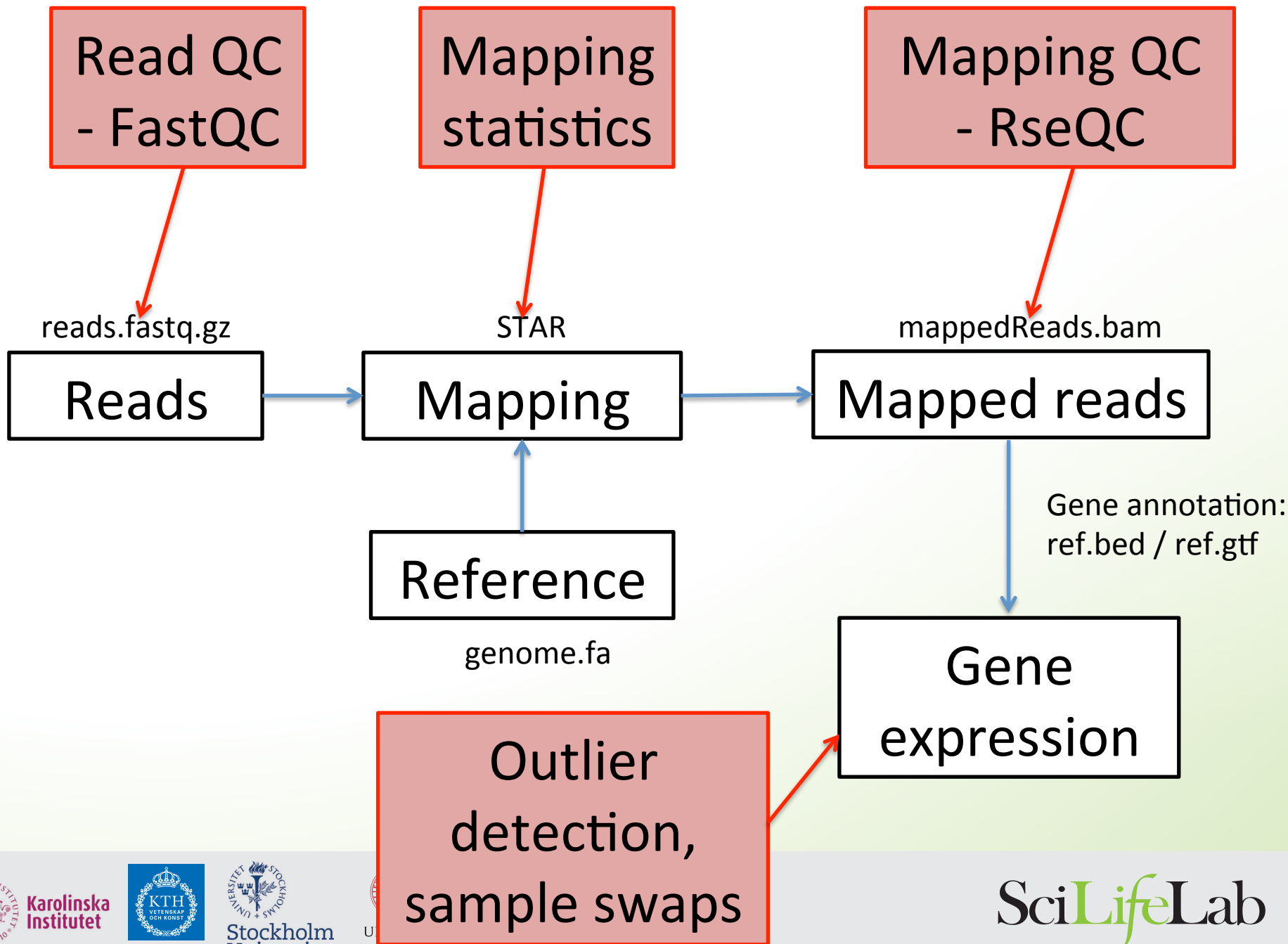
# From samples to RNA to reads

-might not be what you think they are

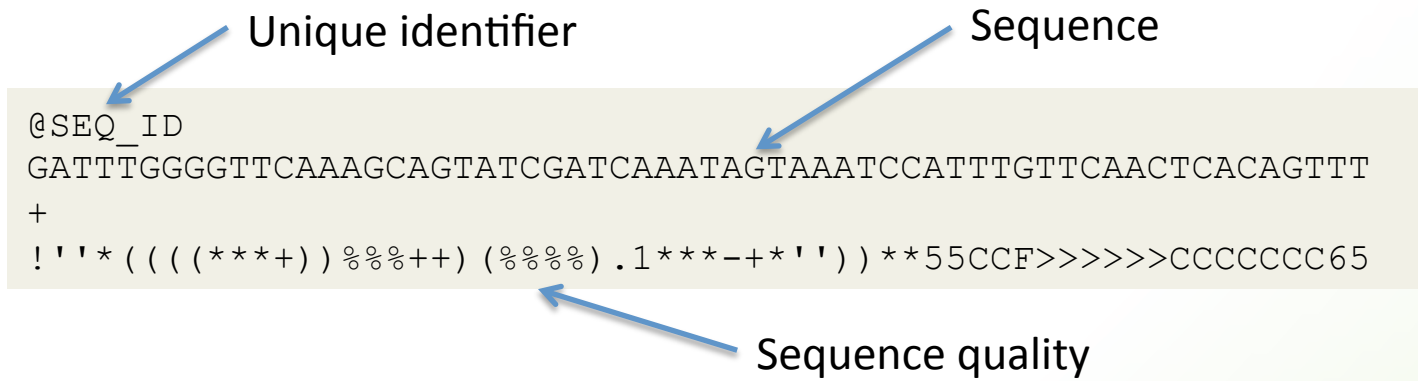
- Mixing samples
  - 30 samples with 5 steps from samples to reads has 24 300 000 potential mix ups of samples
  - Error rate 1/ 100 with 5 steps suggest that one of every 20 sample is mislabeled
- Experiments go wrong
  - 30 samples with 5 steps from samples to reads has 150 potential steps for errors
  - Error rate 1/100 with 5 steps suggest that one of every 20 samples the reads does not represent the sample
- Combine the two steps and approximately one of every 10 samples are wrong

# RNA-seq analysis workflow





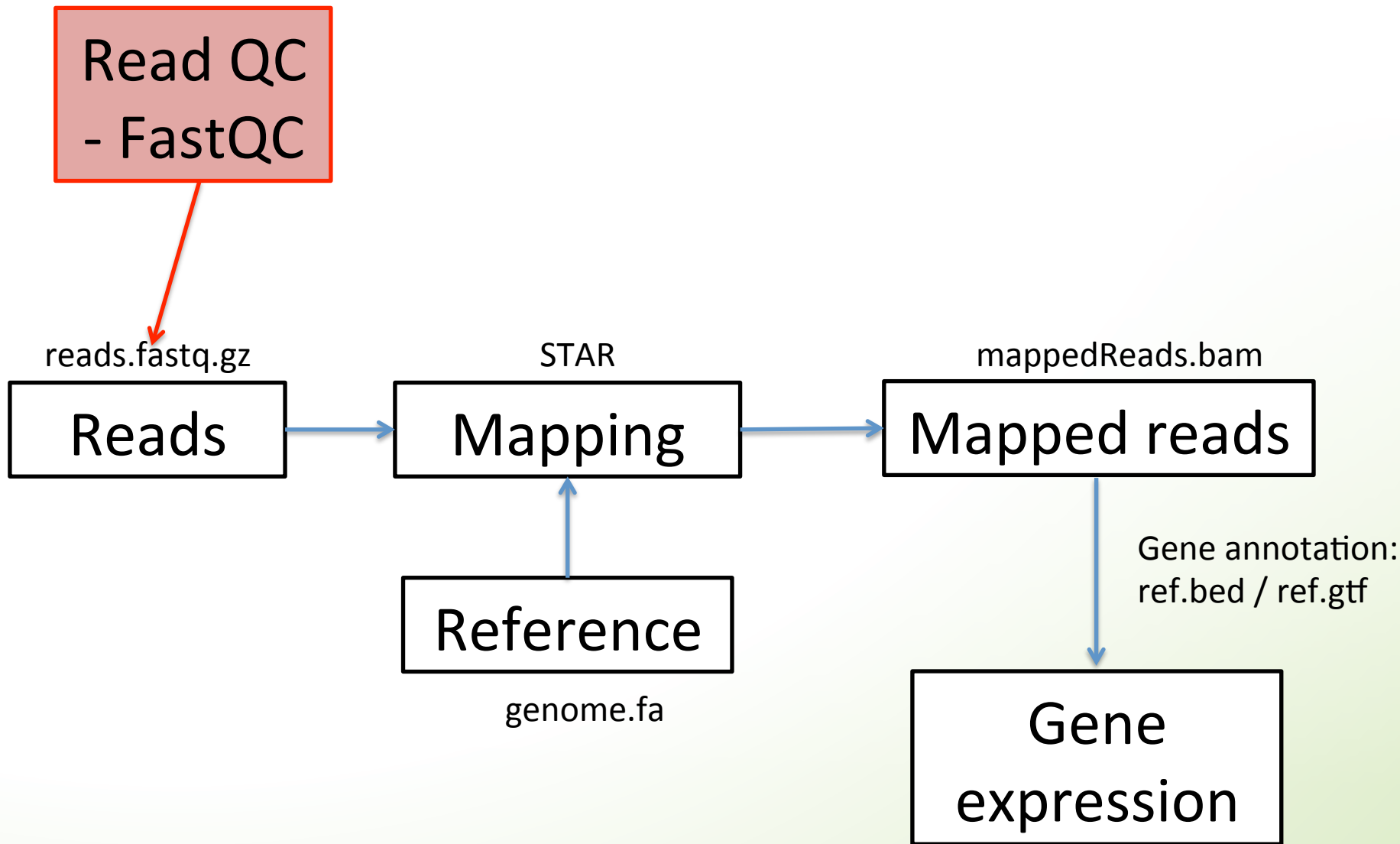
# Fastq – read file format



Paired end data usually in format sampleX\_1.fastq and sampleX\_2.fastq with same SEQ\_ID for both mate pairs, followed by /1 and /2 (or \_f and \_r)







# Basic read metrics with FastQC

A program that analyses some of the basic metrics on fastq raw read files.

- Quality
- Length
- Sequence bias
- GC content
- Repeated sequences
- Adapter contamination

## Code

```
$ module load bioinfo-tools
$ module load FastQC/0.11.2

$ fastqc -o outdir seqfile.fastq
# multiple files:
$ fastqc -o outdir seqfile_*.fastq
```

<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>

# FastQC report

## FastQC Report

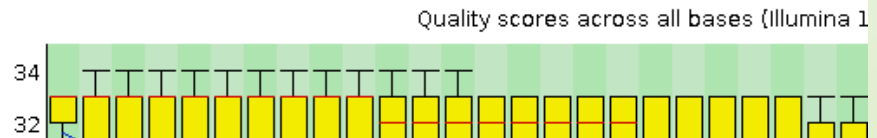
### Summary

- ✓ [Basic Statistics](#)
- ✗ [Per base sequence quality](#)
- ✗ [Per tile sequence quality](#)
- ✓ [Per sequence quality scores](#)
- ! [Per base sequence content](#)
- ! [Per sequence GC content](#)
- ✓ [Per base N content](#)
- ✓ [Sequence Length Distribution](#)
- ! [Sequence Duplication Levels](#)
- ! [Overrepresented sequences](#)
- ✓ [Adapter Content](#)
- ! [Kmer Content](#)

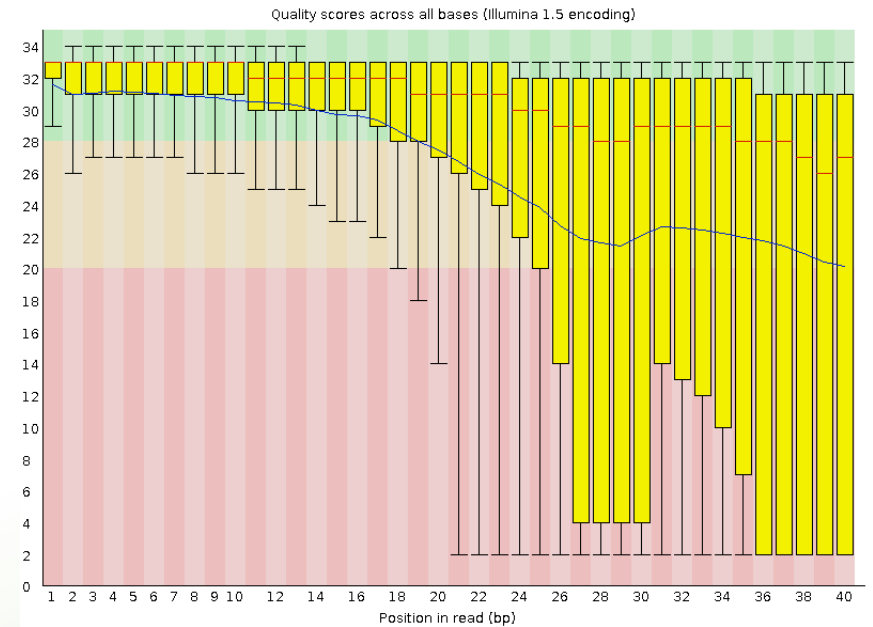
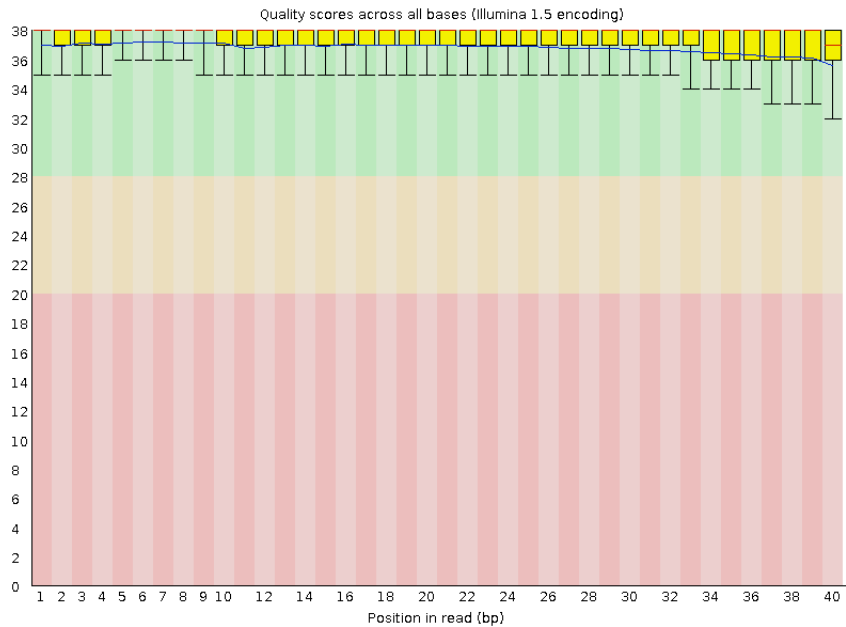
### Basic Statistics

Measure	Value
Filename	bad_sequence.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	395288
Sequences flagged as poor quality	0
Sequence length	40
%GC	47

### ✗ Per base sequence quality

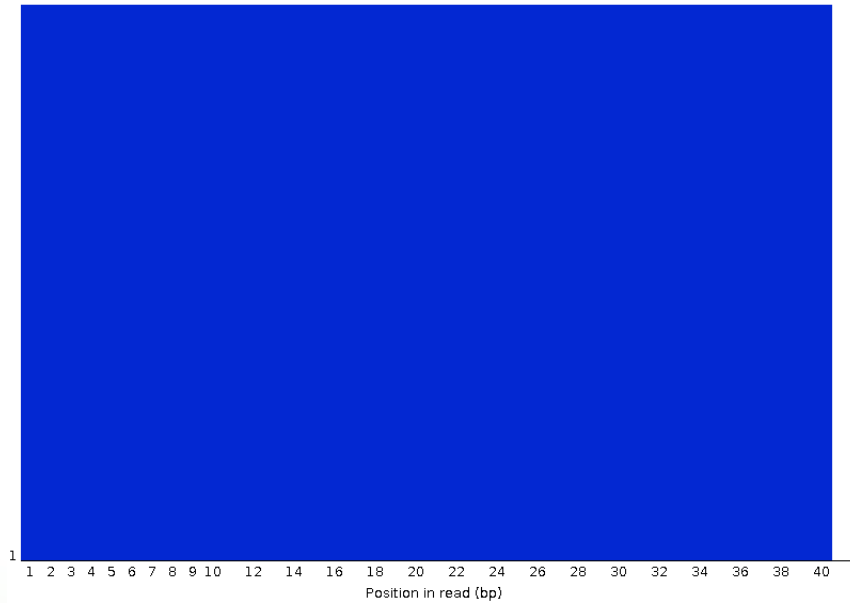


# Per base sequence quality

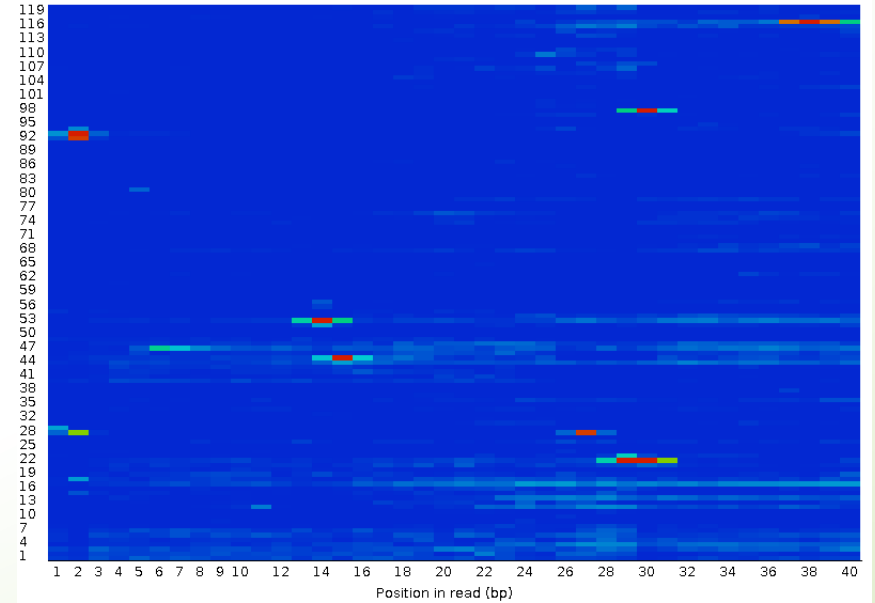


# Per tile sequence quality

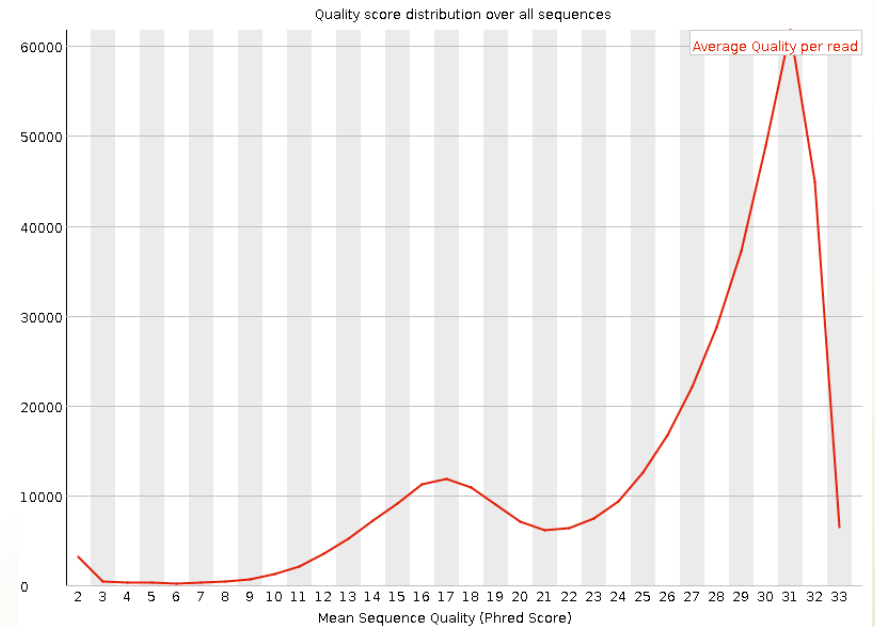
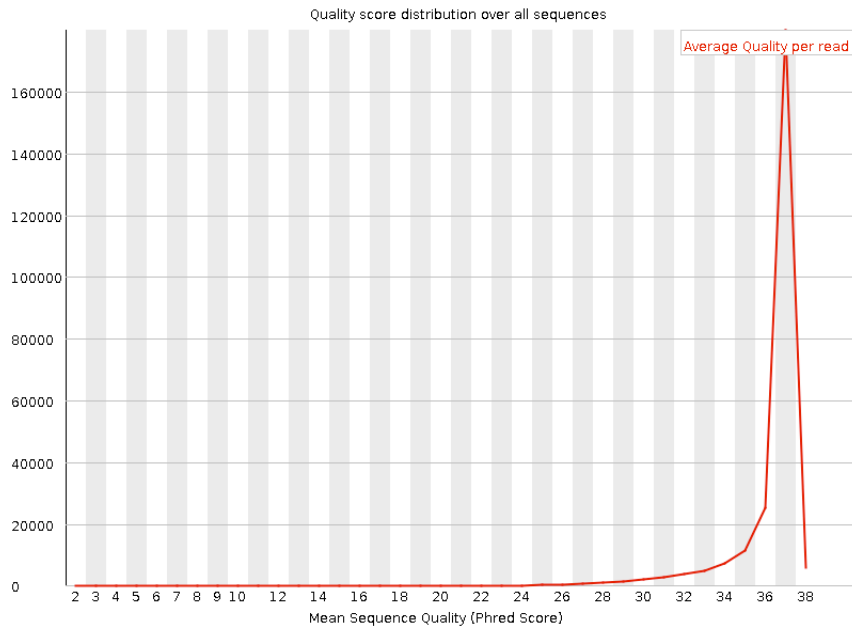
Quality per tile



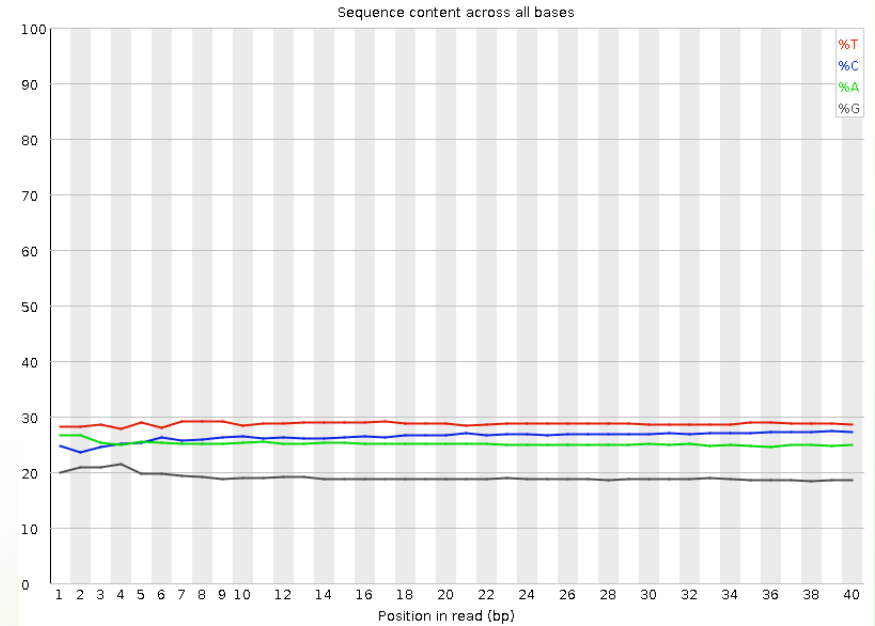
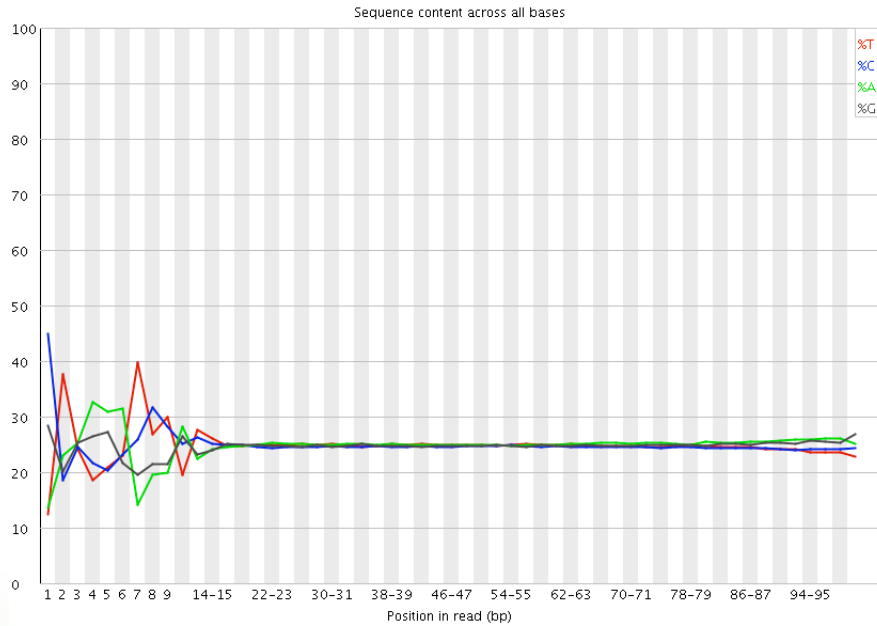
Quality per tile



# Per sequence quality scores

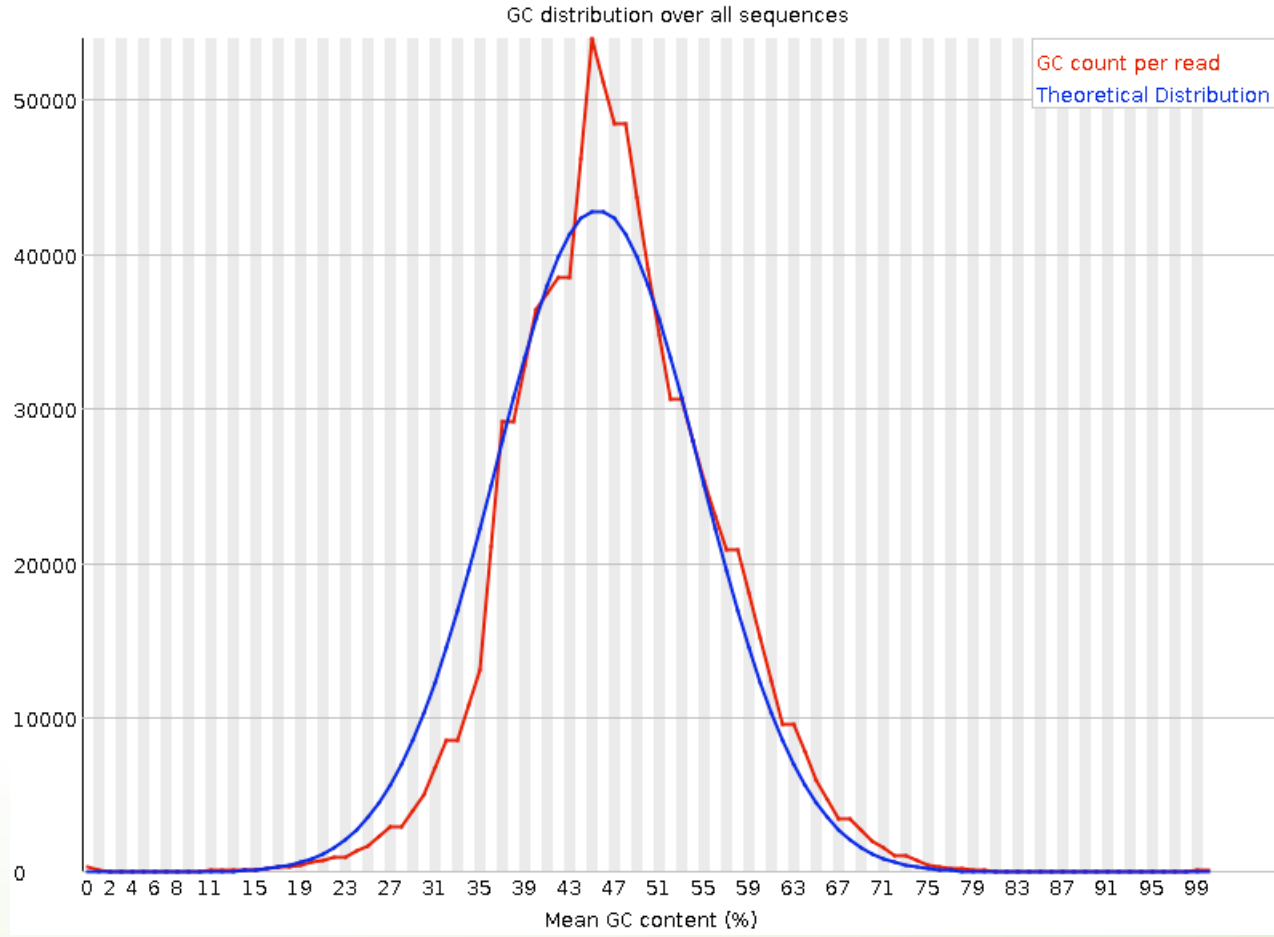


# Per base sequence content

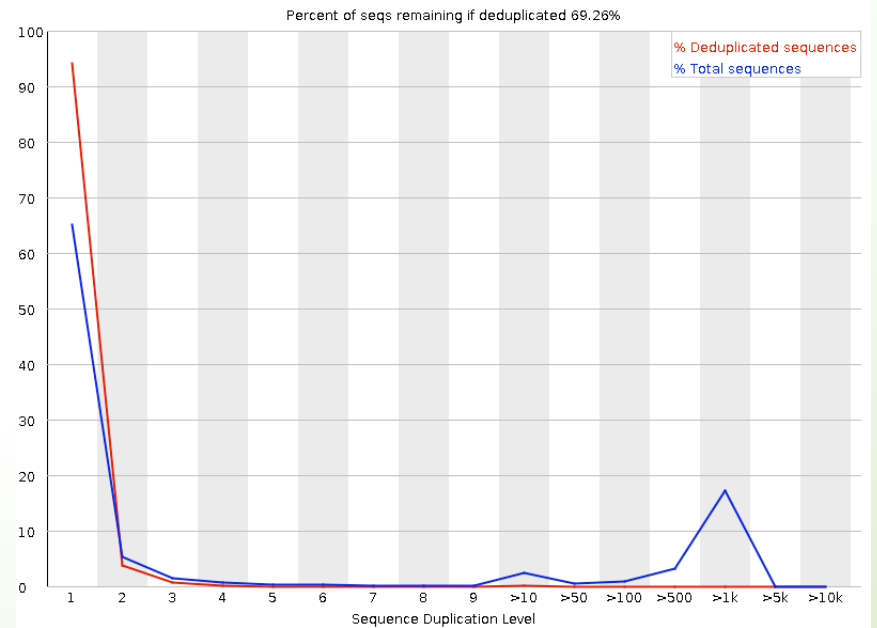
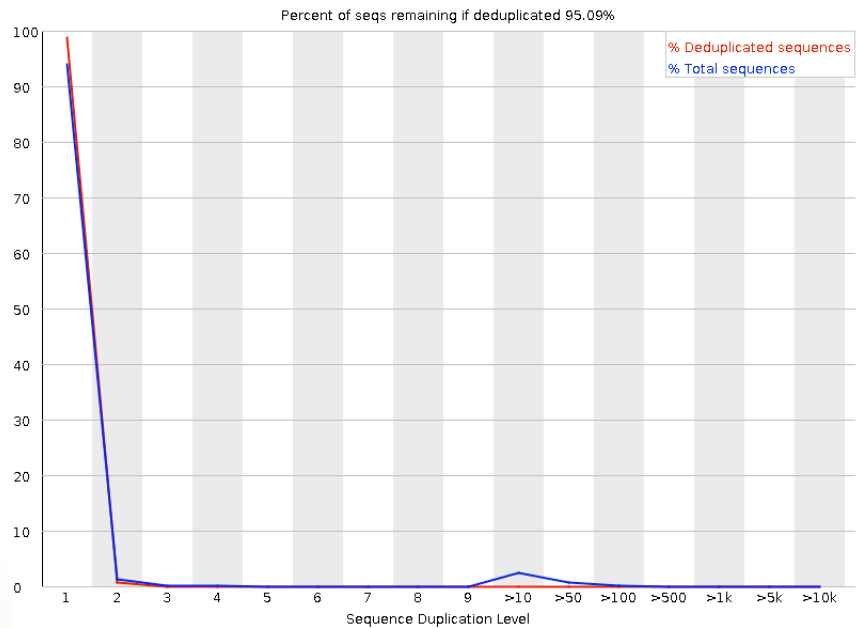




# Per sequence GC content



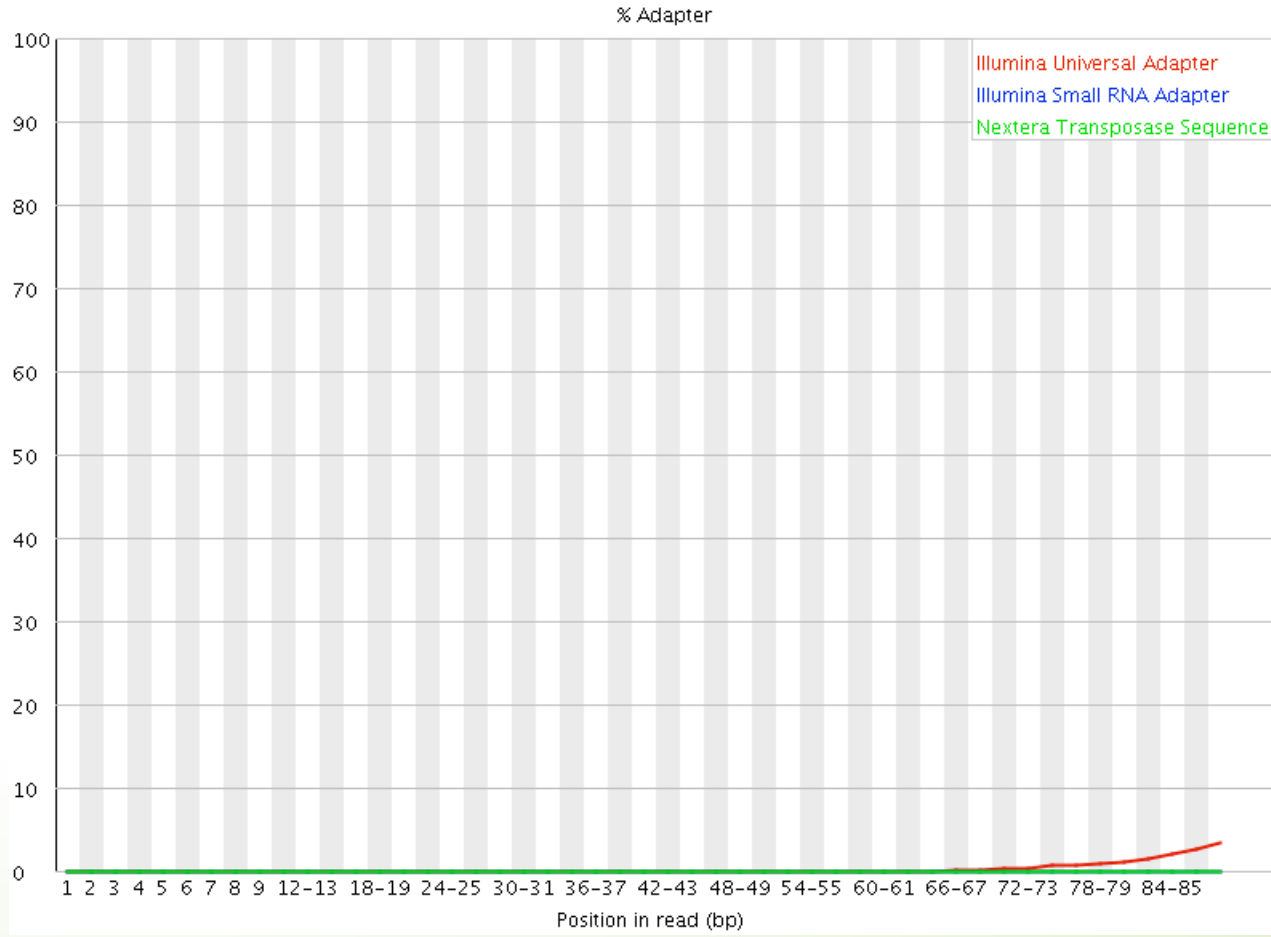
# Sequence Duplication Levels



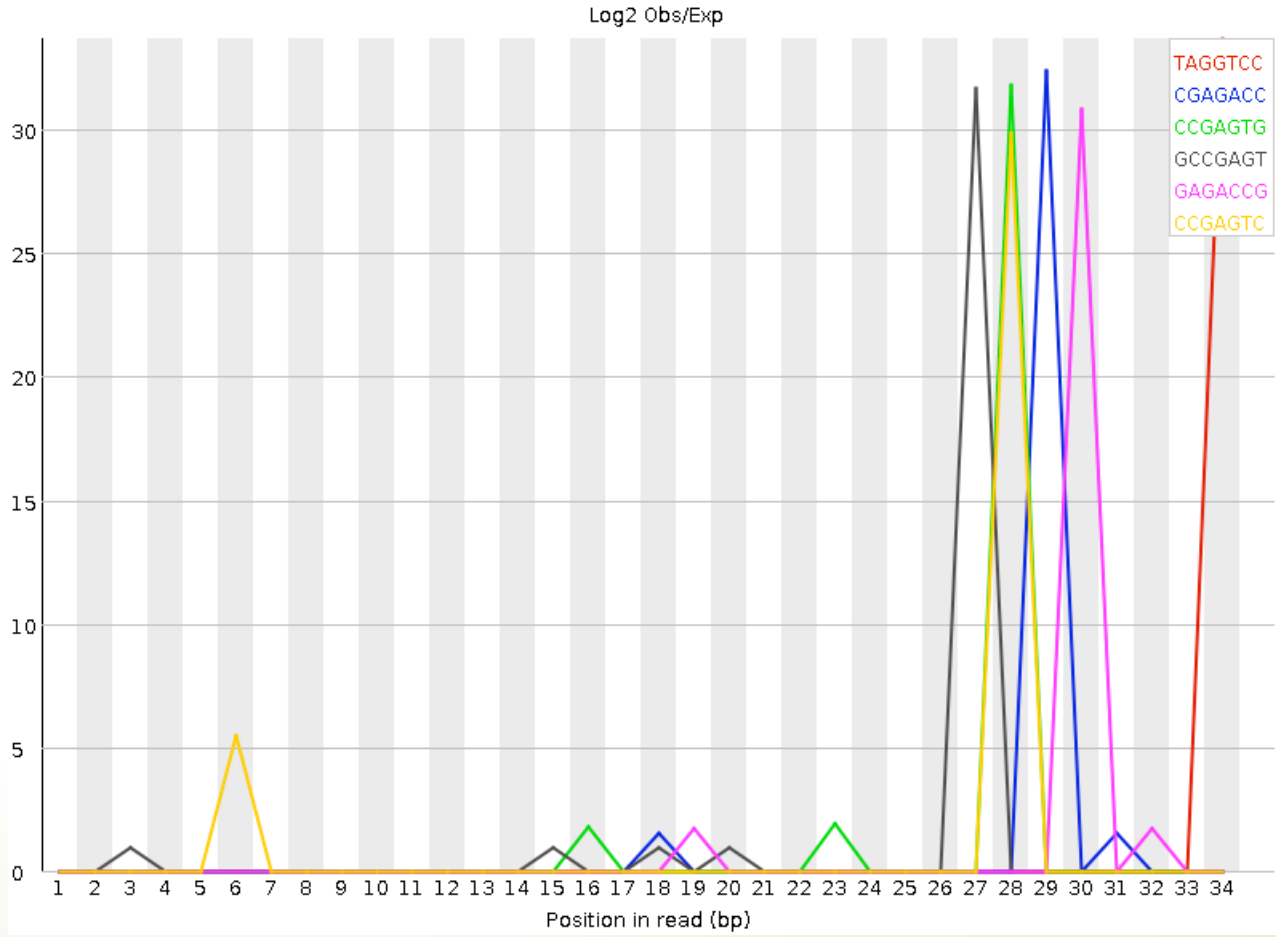
# Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AGAGTTTATCGCTTCCATGACGCAGAAGTTAACACTTTC	2065	0.5224039181558763	No Hit
GATTGGCGTATCCAACCTGCAGAGTTTATCGCTTCCATG	2047	0.5178502762542754	No Hit
ATTGGCGTATCCAACCTGCAGAGTTTATCGCTTCCATGA	2014	0.5095019327680071	No Hit
CGATAAAAATGATTGGCGTATCCAACCTGCAGAGTTTAT	1913	0.4839509420979134	No Hit
GTATCCAACCTGCAGAGTTTATCGCTTCCATGACGCAGA	1879	0.47534961850600066	No Hit
AAAAATGATTGGCGTATCCAACCTGCAGAGTTTATCGCT	1846	0.4670012750197325	No Hit
TGATTGGCGTATCCAACCTGCAGAGTTTATCGCTTCCAT	1841	0.46573637449150995	No Hit
AACCTGCAGAGTTTATCGCTTCCATGACGCAGAAGTTAA	1836	0.46447147396328753	No Hit
GATAAAAATGATTGGCGTATCCAACCTGCAGAGTTTATC	1831	0.4632065734350651	No Hit
AAATGATTGGCGTATCCAACCTGCAGAGTTTATCGCTTC	1779	0.45005160794155147	No Hit
ATGATTGGCGTATCCAACCTGCAGAGTTTATCGCTTCCA	1779	0.45005160794155147	No Hit
AATGATTGGCGTATCCAACCTGCAGAGTTTATCGCTTCC	1760	0.4452449859343061	No Hit
AAAATGATTGGCGTATCCAACCTGCAGAGTTTATCGCTT	1729	0.4374026026593269	No Hit
CGTATCCAACCTGCAGAGTTTATCGCTTCCATGACGCAG	1713	0.43335492096901496	No Hit
ATCCAACCTGCAGAGTTTATCGCTTCCATGACGCAGAAG	1708	0.43209002044079253	No Hit

# Adapter Content

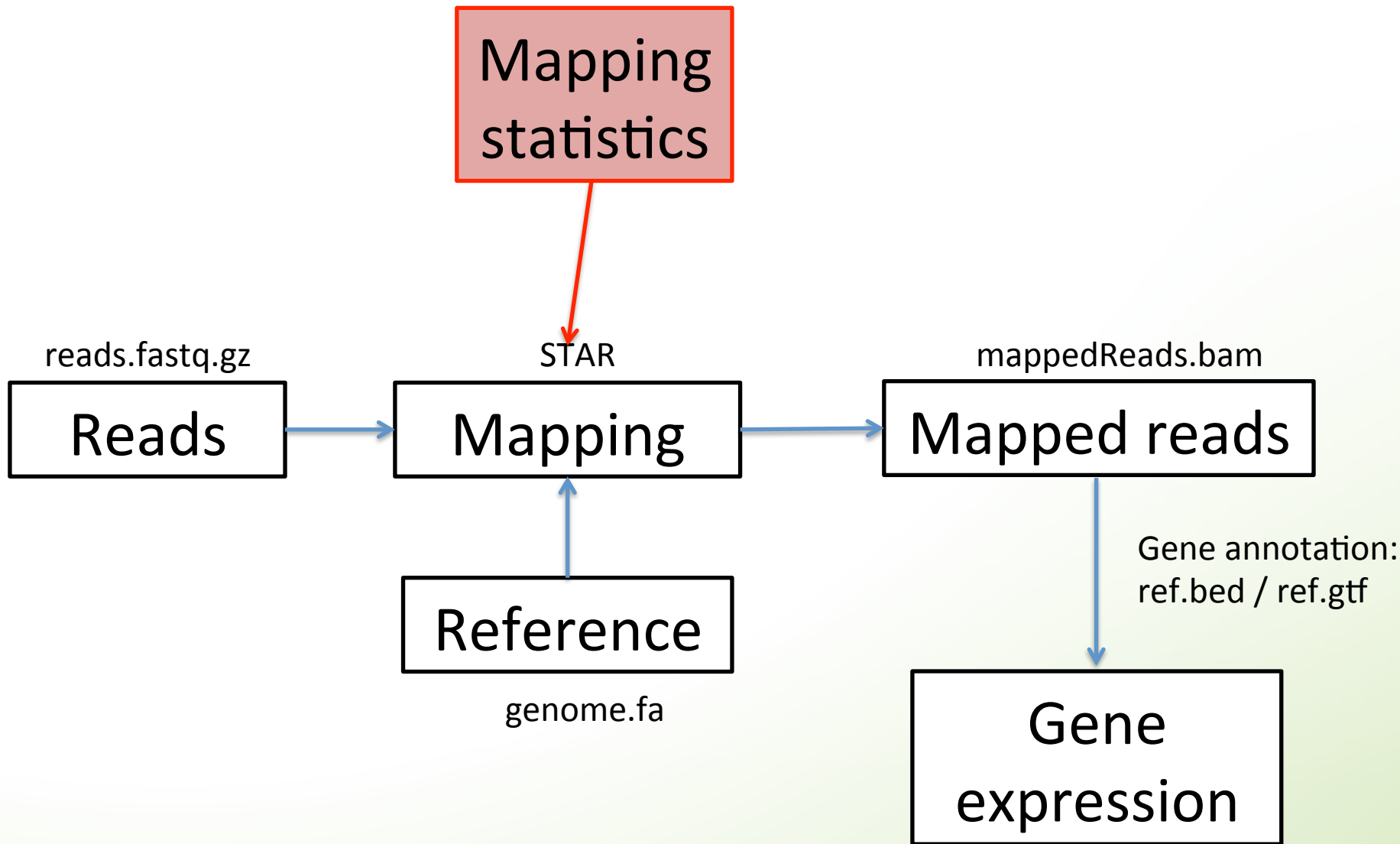


# Kmer content



# Failed FastQC – what to do?

- Try to figure out why
  - If problem seem to be related to problems during sequencing – resequence!
  - If problem is related to library prep – rerun if possible.
- You can filter out the low quality reads
  - Adapter trimming (cutadapt)
  - Filter low phred score reads (samtools, jaccard)
- If you have enough reads after filtering the data may still be useful.
- But be careful to do equal trimming on all samples!



# SAM format

```
HWI-ST1018:7:1101:1648:2188#0 99 chr1 115275270 255 1S100M = 115275321 152
NTTCTATATTGGTTGCTCGCTCTAATTTGTCACGTCGGTCTGTTGAAATATTAACCTAACATGGTCACCTTCCAGCAGGGTCACCTTGGATTTTCGTATCT BS
\cceeeggggghhhhhbghhhhhhhhhhhfhhhhhhhhfhhhhhhhhfhhhhfhhhhffgghhg\Z^ddeeedbdbdcacbabcbcccbbcc^abc] NH:i:1
HI:i:1 AS:i:194 nM:i:0
HWI-ST1018:7:1101:1648:2188#0 147 chr1 115275321 255 101M = 115275270 -152
AAACCTAACATGGTCACCTTCCAGCAGGGTCACCTTGGATTTTCGTATCTTTGTCTCCAAAGGGAAGTTCTTTAGGGATCACAAAGTCNANTTTGNTNNGTC
BBcxbdccccccccbbccccddcddeeeecggggghihiiiiifhifhfgiiahhhhhihhhiiiiihiiiiihiiiiihhhihgd]RBRBec]QBQBBbbb NH:i:1
HI:i:1 AS:i:194 nM:i:0
HWI-ST1018:7:1101:2039:2206#0 99 chr19 14574483 255 1S72M85N28M = 14574529
232 NCCTTCCGCAACCCTGTCATTGAGAGGATTCCTCGGCTCCGACGGCAGAAGAAAATTTCTCCAAGCAGCAAGGGAAGGCGTTCCAGCGTGCTAGGCAGAT
BP\cceeefgggghhighiiiiiiiiihhhiiiiiiiiihiggeeeddddbbbcccb^[\`accccccccX]acccc^acc]bc^b_a] NH:i:1
HI:i:1 AS:i:203 nM:i:0 XS:A:+
HWI-ST1018:7:1101:2039:2206#0 147 chr19 14574529 255 26M85N75M = 14574483
-232
GAAGAAAATTTTCTCCAAGCAGCAAGGGAAGGCGTTCCAGCGTGCTAGGCAGATGAACATCGATGTCGCCACGTGGGTGCGGCTGCTCCGGAGGCTCATCC ]ccdccb
bbacbcaccccbcccccccccccccbccacccccccdd_dddeeeeeeggggiiiiihiiiiiiiiihiiiiiiiiifgfgggeeeebbb NH:i:1 HI:i:1
AS:i:203 nM:i:0 XS:A:+
```

More details on:

<http://samtools.github.io/hts-specs/SAMv1.pdf>

<http://genome.sph.umich.edu/wiki/SAM>



# BAM/SAM file formats

- All mapped reads with location in genome, mapping information etc. (<https://samtools.github.io/hts-specs/SAMv1.pdf>)
- SAM (Sequence Alignment/Map) format – alignment.sam
- BAM is a compressed sam format – alignment.bam
- A bam-file (always) needs to be indexed and sorted - alignment.bam.bai
- Samtools – a simple program for converting between bam/sam, indexing, sorting, filtering, etc.

## Code

```
$ module load bioinfo-tools  
$ module load samtools
```



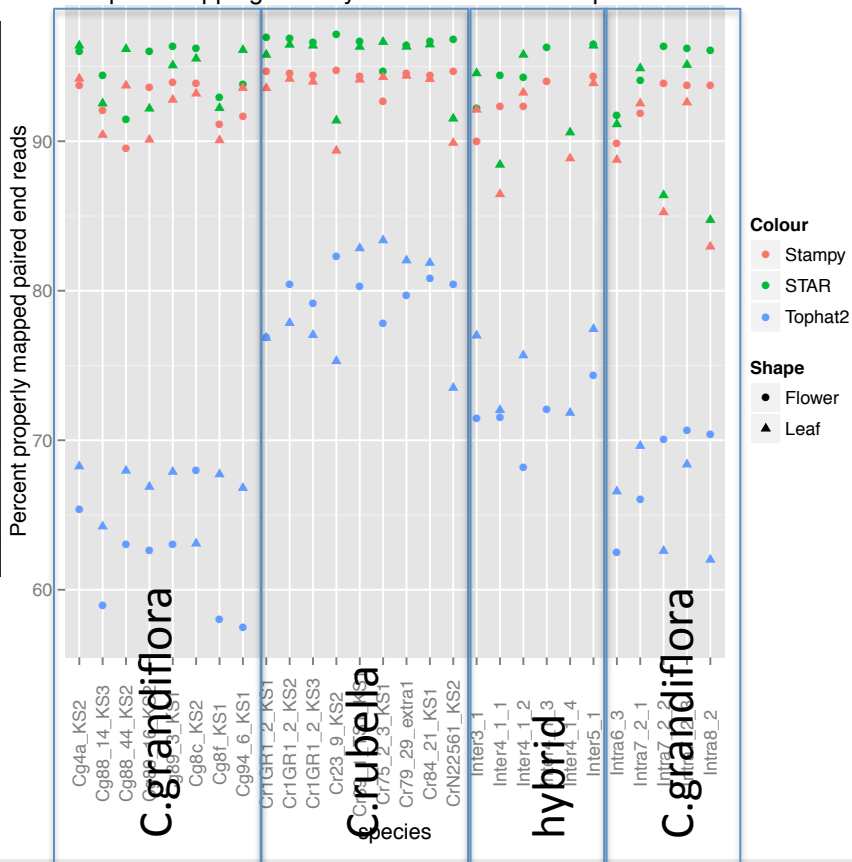
# Bad mapping – what to do?

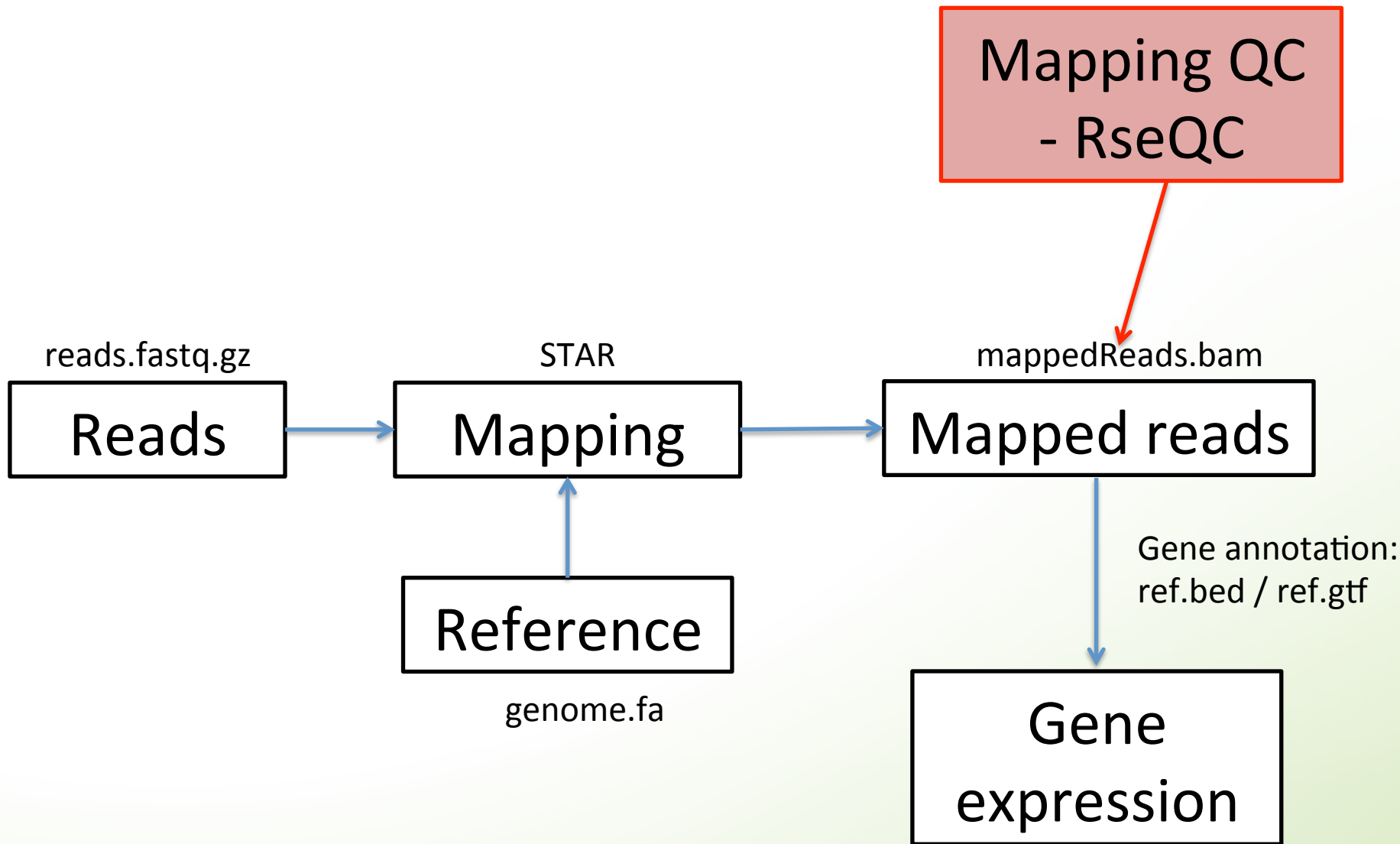
- First step – try to figure out why it failed. With the use of FastQC/RseQC/Mapping logs.
  - Perhaps also look for contaminant species
  - Redo library prep controlling for possible errors
- Low mapping, but not completely failed.
  - Figure out why!
  - Is it equal for all samples?
  - Could it introduce any bias in the data?

# More variation when using top hat 2 with default settings than when using STAR or Stampy with default setting



Compare mapping efficacy of different RNA-seq assemblers





# After mapping - RseqQC package

- General sequence QC:
  - sequence quality
  - nucleotide composition bias
  - PCR bias and
  - GC bias
- RNA-seq specific QC:
  - evaluate sequencing saturation
  - mapped reads distribution
  - coverage uniformity
  - strand specificity
  - Etc..
- Some tools for file manipulations

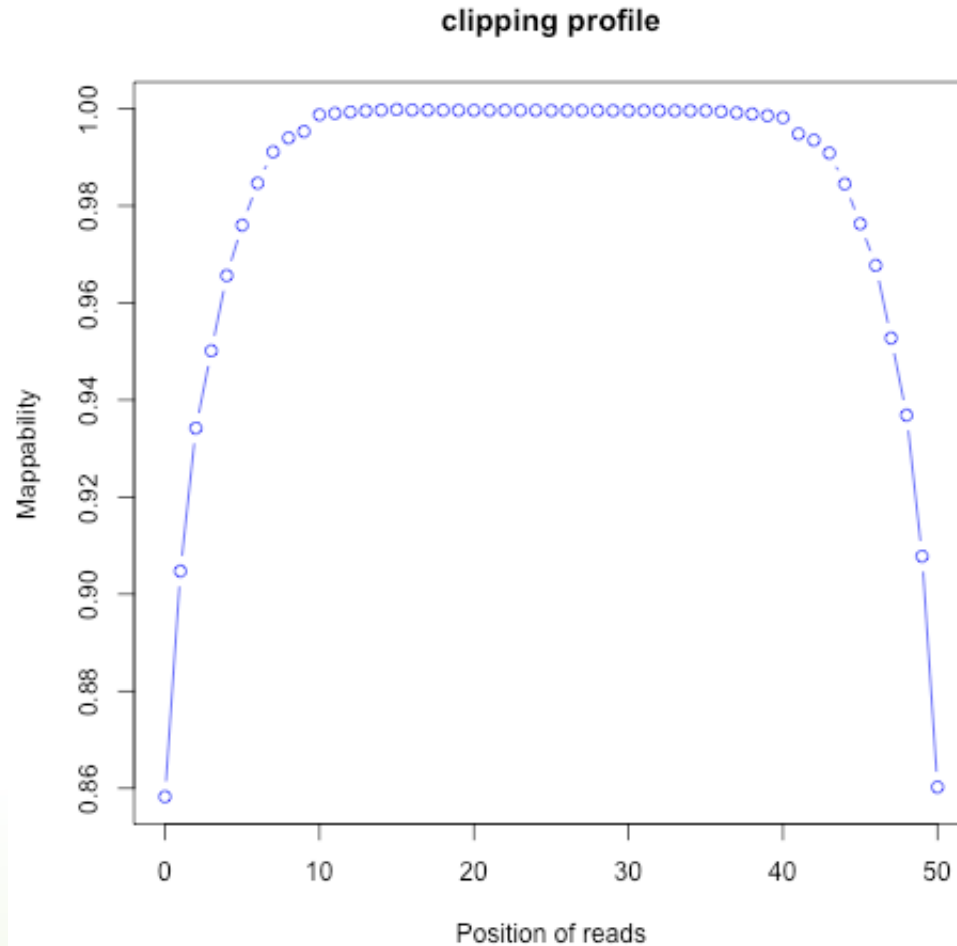
## Code

```
$ module load bioinfo-tools
$ module load rseqc/2.4

$ geneBody_coverage.py -r
ref.bed12 -i mappedReads.bam -o
genecoverage
```

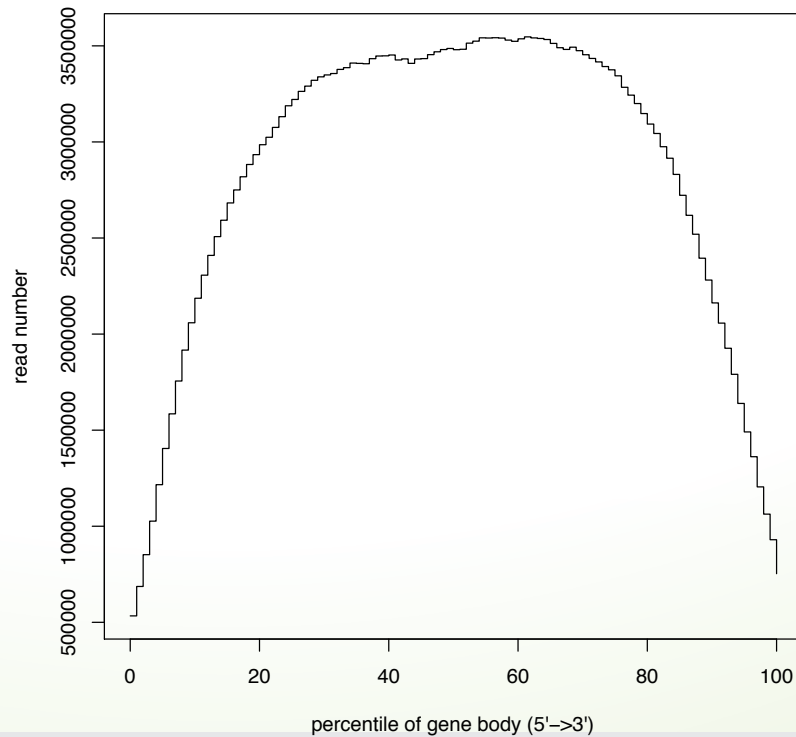
<http://rseqc.sourceforge.net/>

# Soft clipping - clipping\_profile.py

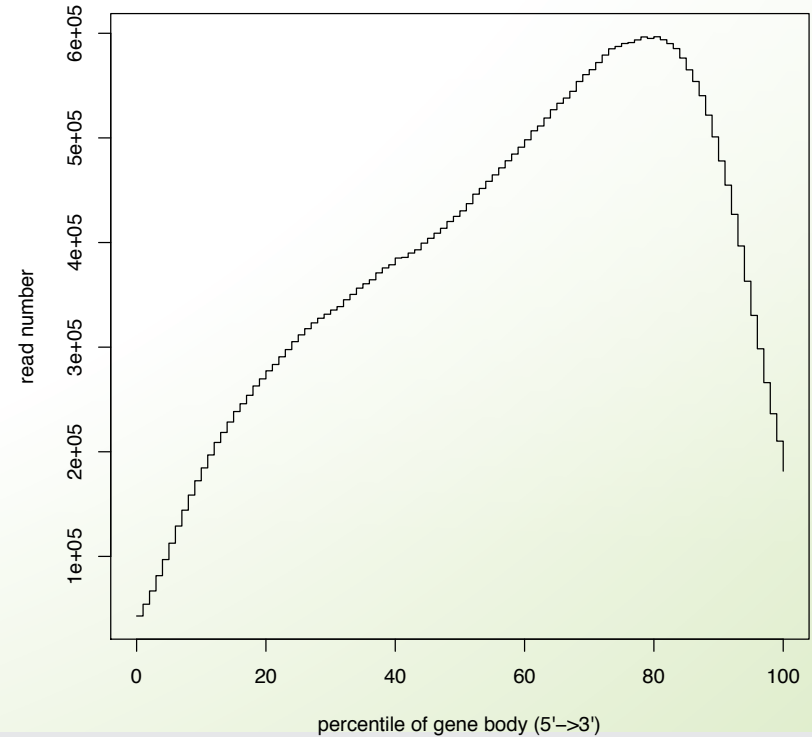


# Gene coverage - geneBody\_coverage.py

## Not degraded

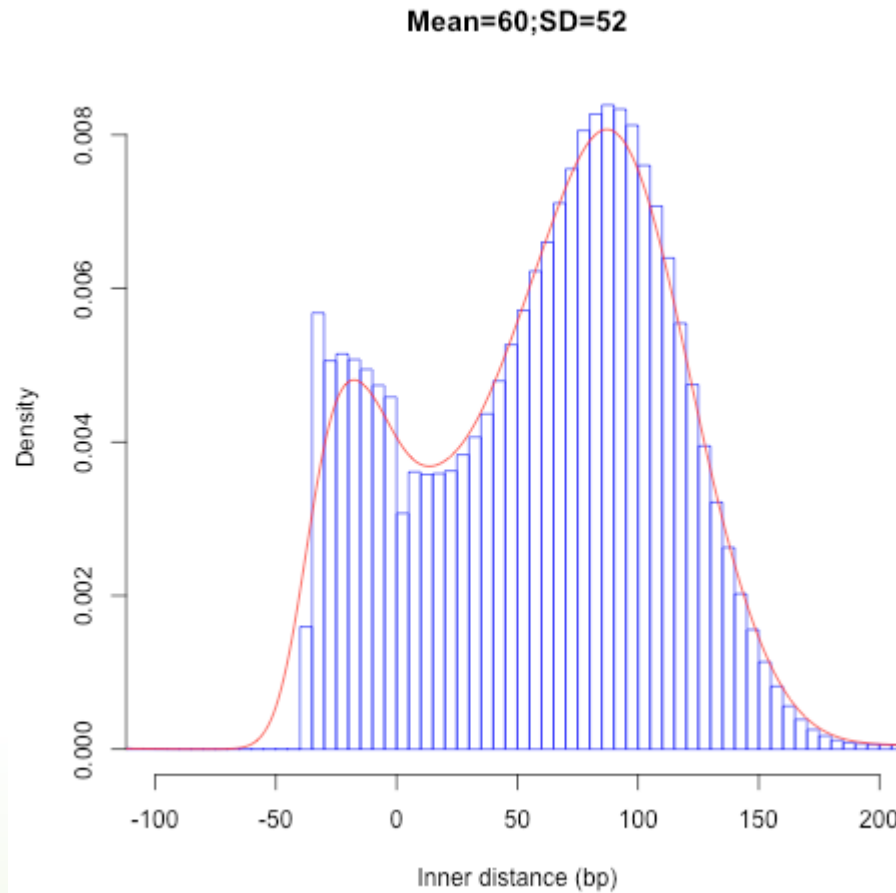


## Degraded





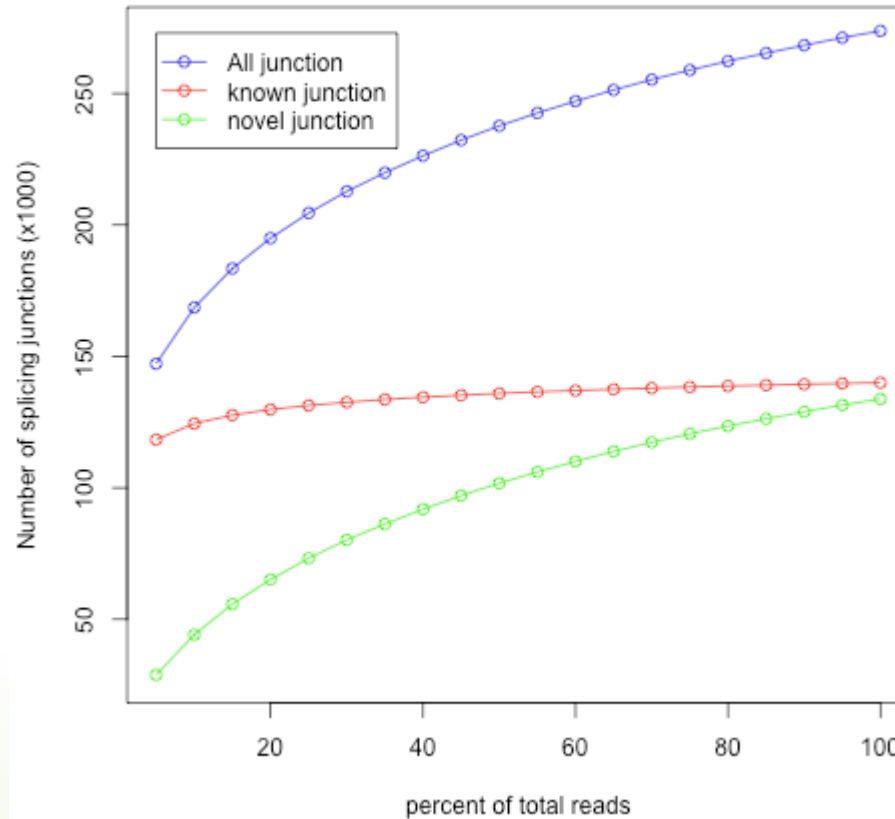
# Distance between PE-reads - inner\_distance.py



# Where in the genome do your reads map? - read\_distribution.py

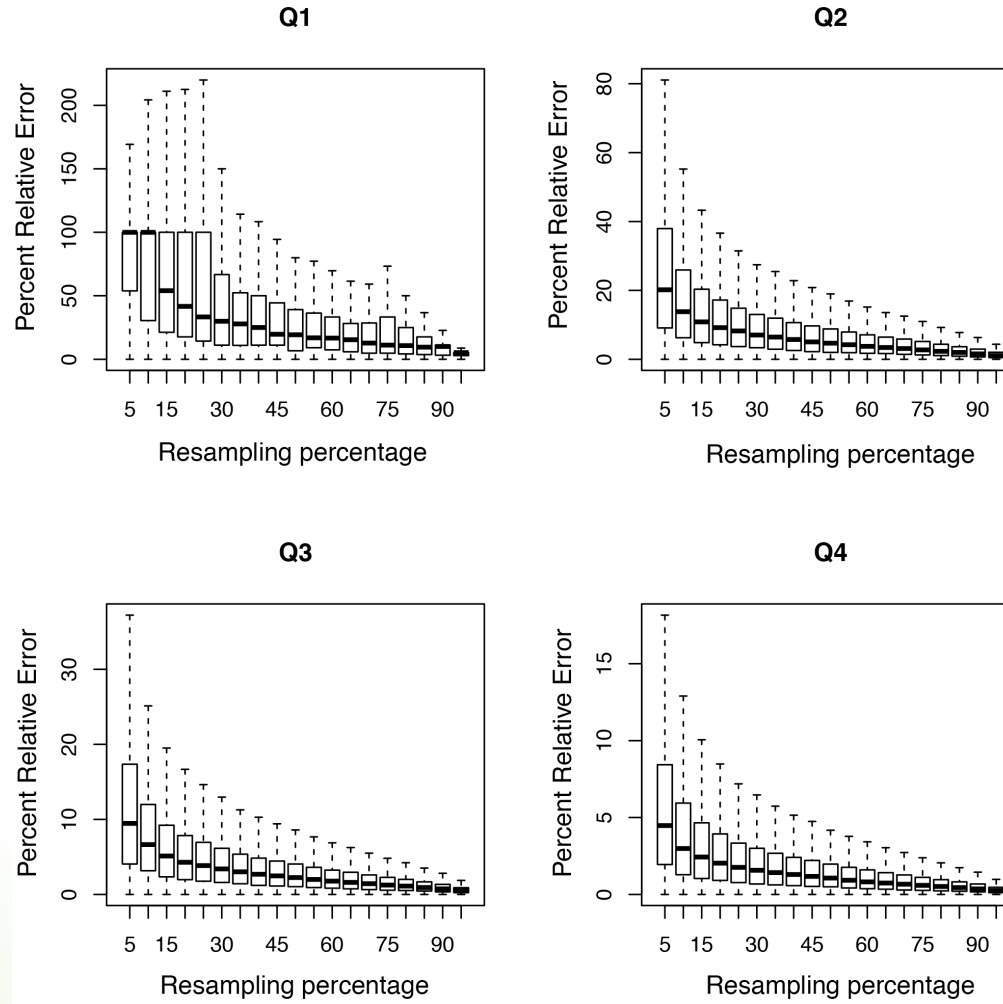
Group	Total_bases	Tag_count	Tags/Kb
CDS_Exons	33302033	20002271	600.63
5'UTR_Exons	21717577	4408991	203.01
3'UTR_Exons	15347845	3643326	237.38
Introns	1132597354	6325392	5.58
TSS_up_1kb	17957047	215331	11.99
TSS_up_5kb	81621382	392296	4.81
TSS_up_10kb	149730983	769231	5.14
TES_down_1kb	18298543	266161	14.55
TES_down_5kb	78900674	729997	9.25
TES_down_10kb	140361190	896882	6.39

# Known and novel splice junctions – junction\_saturation.py or junction\_annotation.py



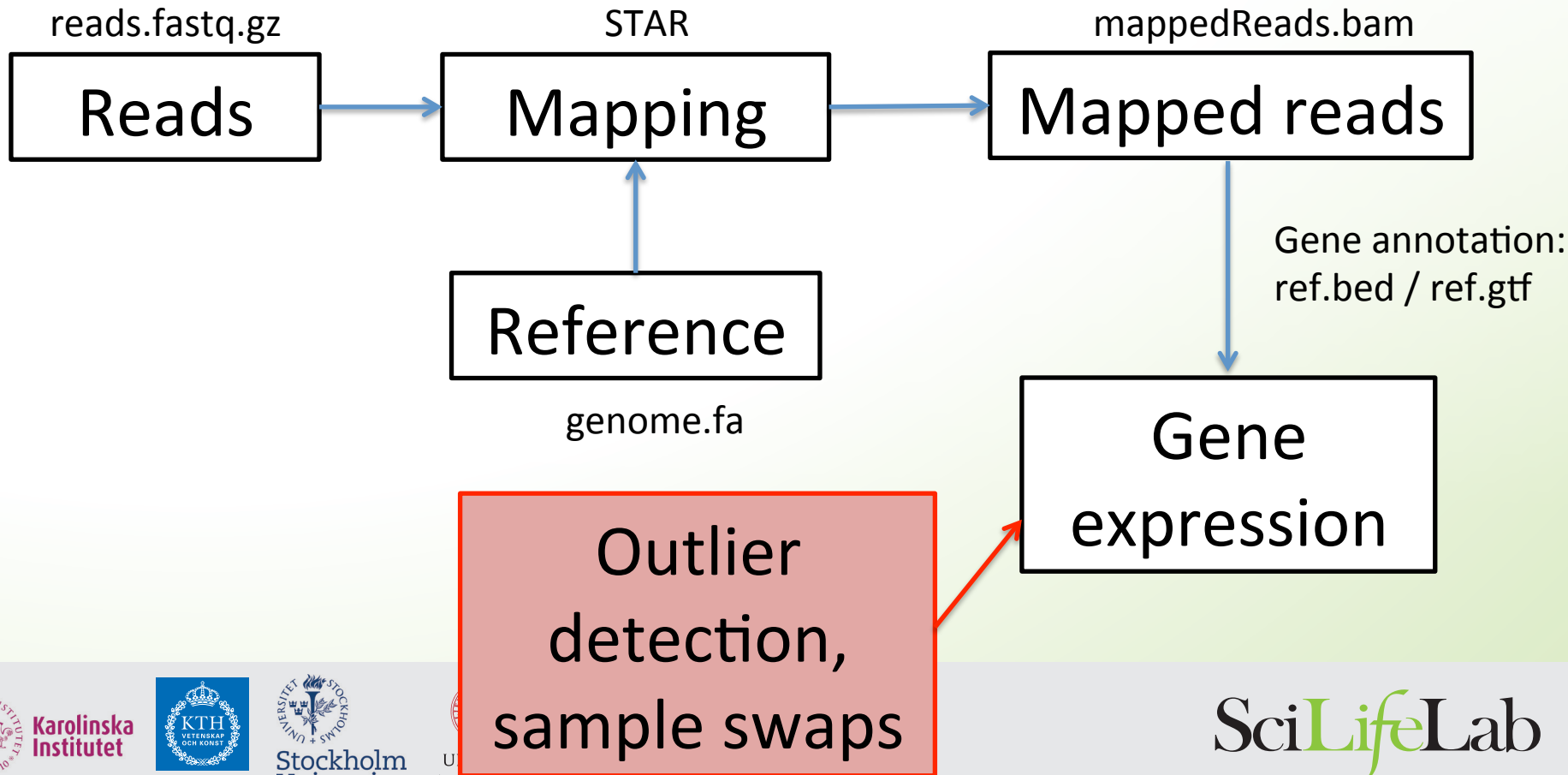
# Gene detection subsampling - RPKM\_saturation.py

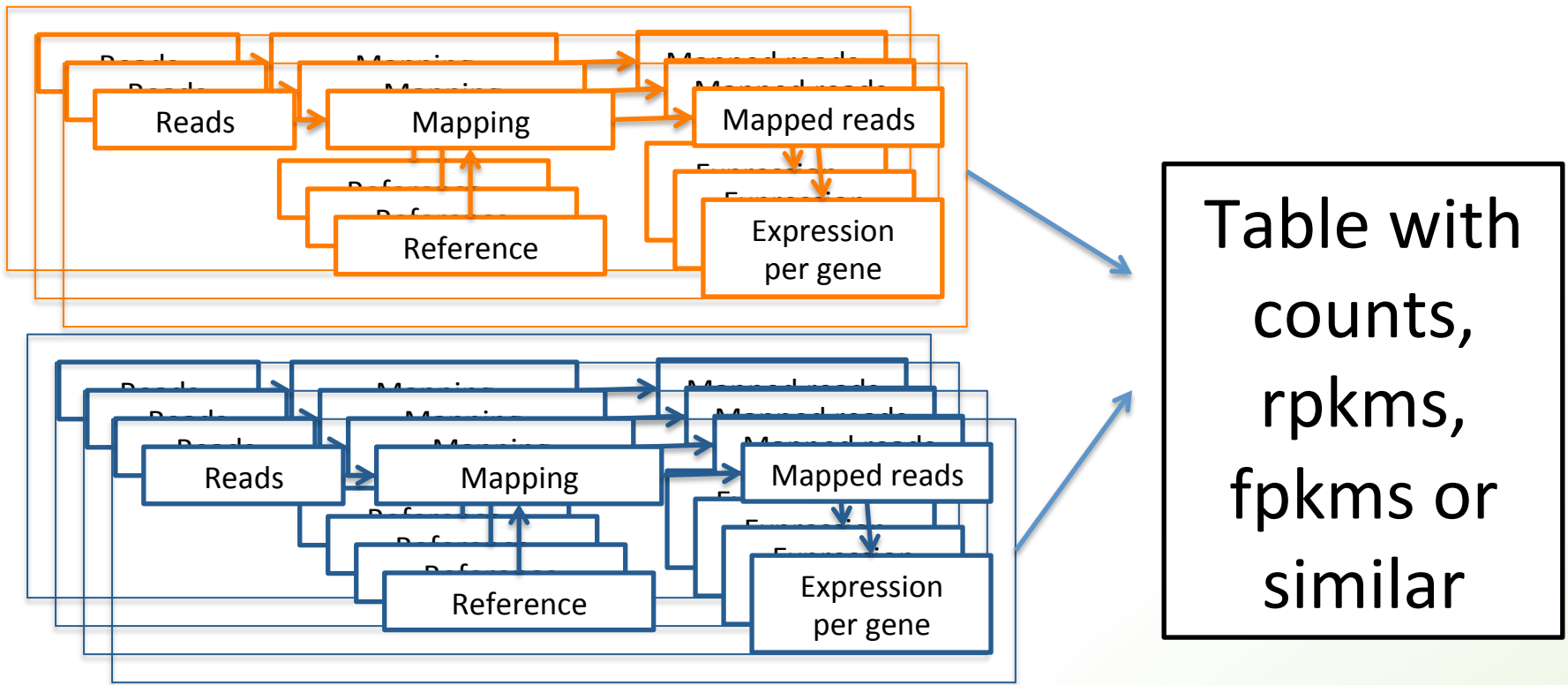
## How deep do you need to sequence?



# Bad RseQC output – what to do?

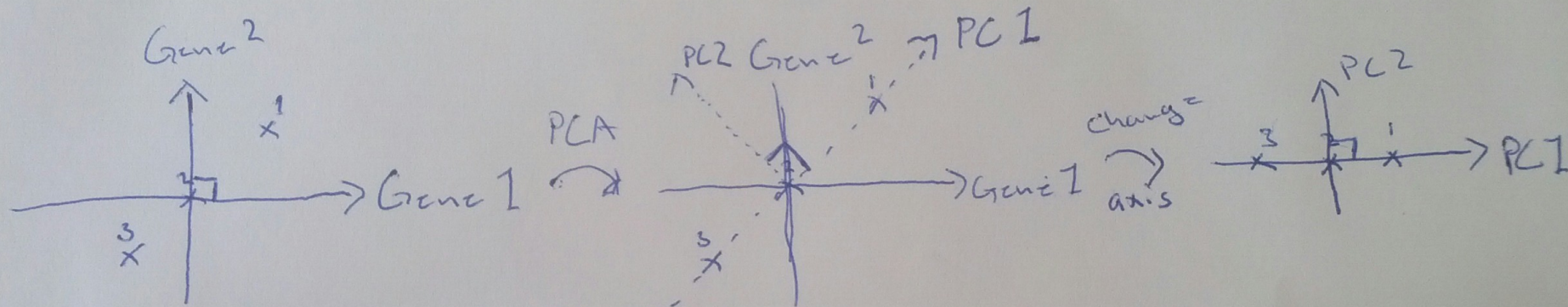
- Try to figure out what went wrong.
  - Redo library prep controlling for possible errors
  - Is it equal for all samples?
  - Could it introduce any bias in the data?
- RNA-degradation in some samples
  - Possible to use a region at 3' end for expression estimates.





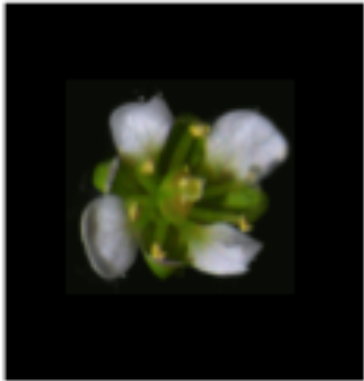
**Sample swaps and outliers can be identified using PCA**

# Differences in read distribution between samples can be identified using Principal Component Analysis (PCA)





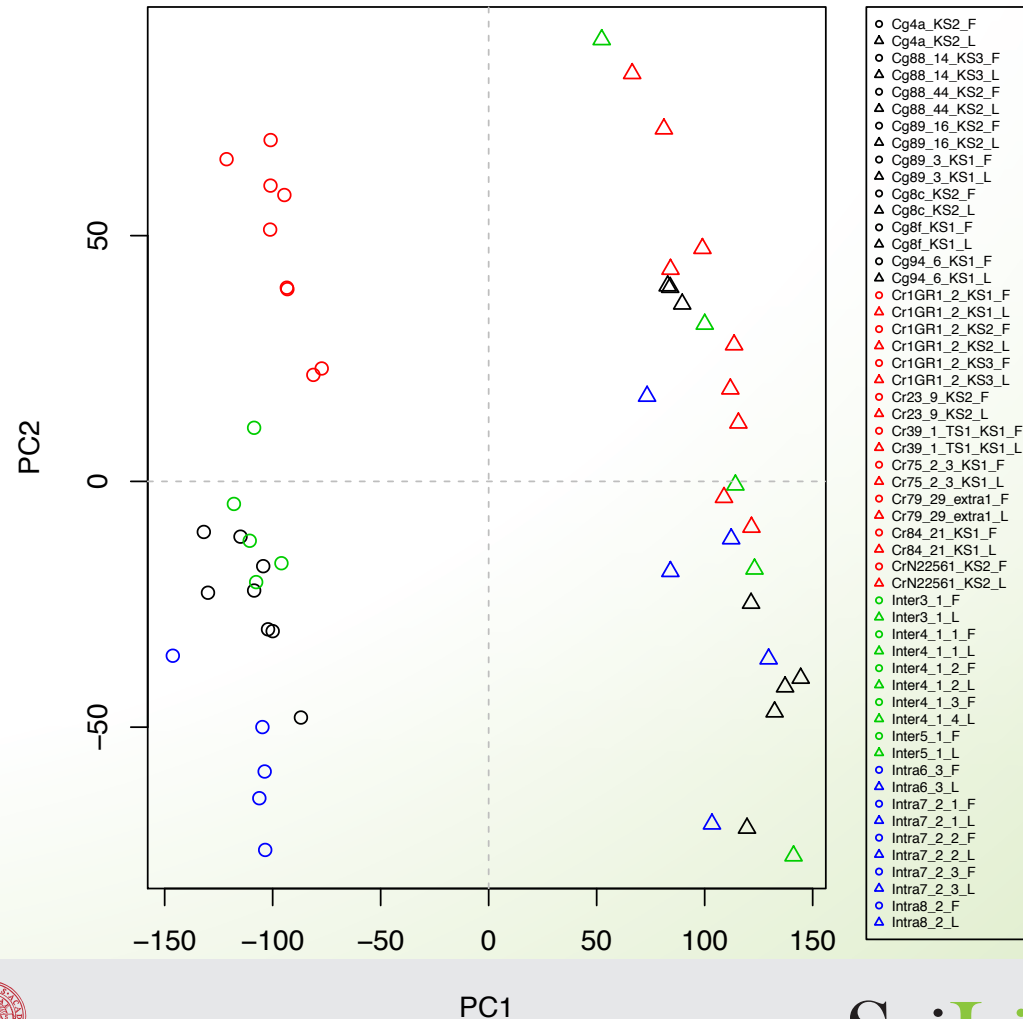
# QC test case 1



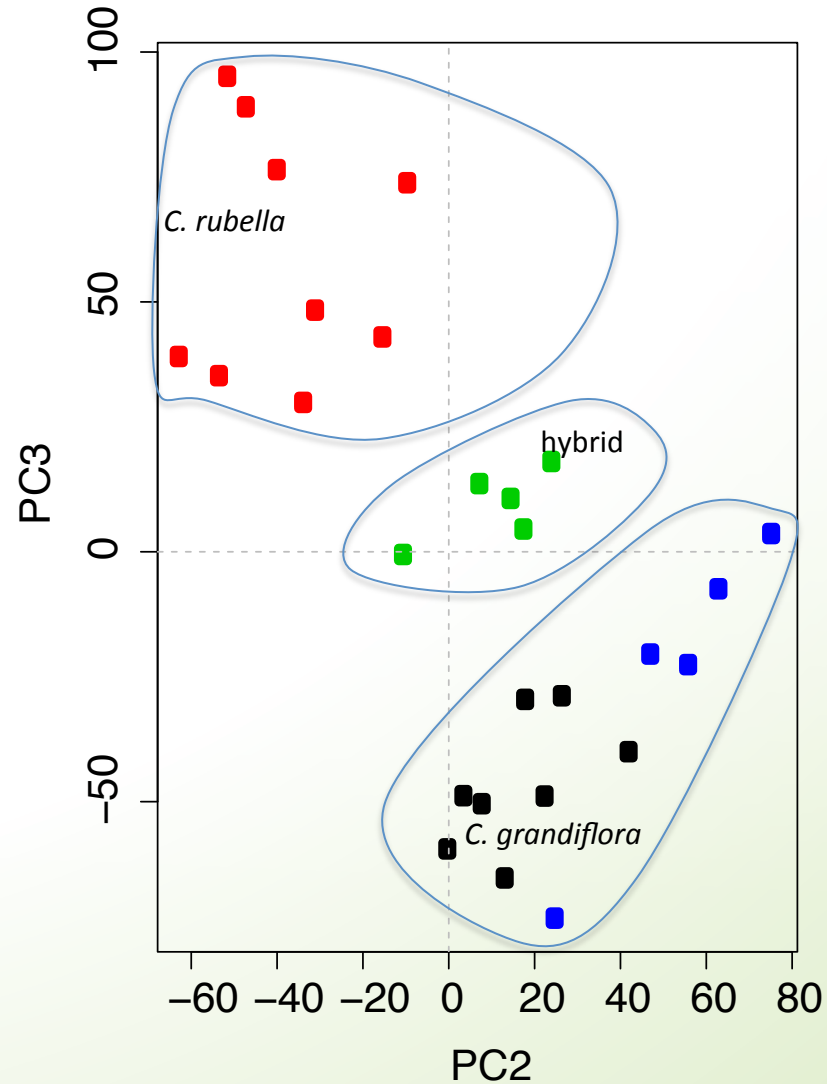
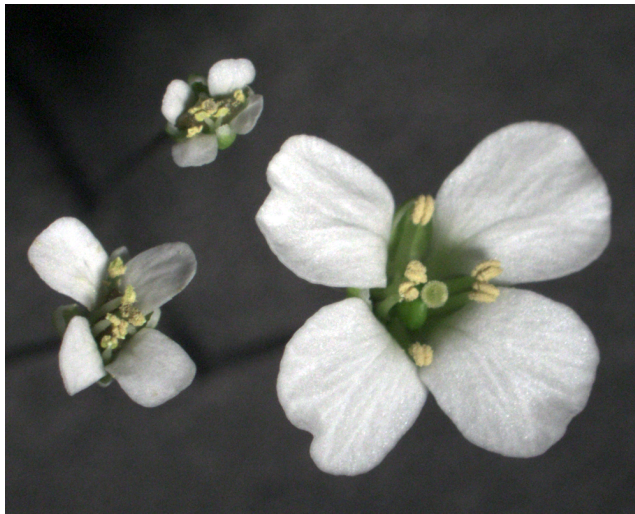
Samples from three different species

1. *C. rubella*
  - Small flowers
  - Normal leaves
  - Genome is sequenced
2. *C. grandiflora*
  - Large flowers
  - Normal leaves
3. Hybrid
  - Intermediate flowers
  - Normal leaves

# Principal component 1 separates samples from flowers and leaves



# Principal component 2 and 3 separates the different species



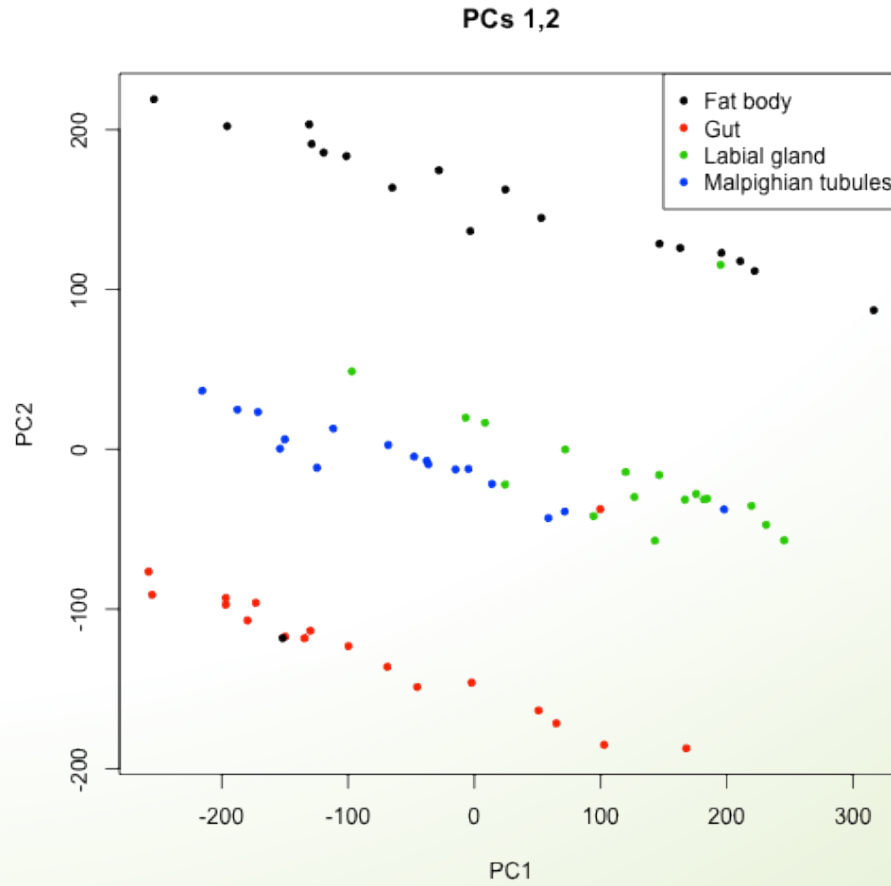
# QC test case 2

- 4 Tissues
  - Fat body
  - Gut
  - Labial gland
  - Malpighian tubules
- 3 Phylogenetic groups
- >70 samples



ButterflyUtopia.com

# PCA analysis detected potential sample swaps



# My PCA looks strange – what to do?

- Clear sample swaps
  - Check sequence indices, lab logs etc. to verify new classification.
  - If you have enough replicates, remove instead of changing labels if you are uncertain.
- Outliers
  - Figure out why they are outliers
  - Do not remove samples only because they do not fit your expectation – Bad science!
- PCA does not group my sample sets
  - Try different methods of dimensionality reduction / clustering
  - Perhaps technical/biological variation is higher than your expected effect -> Batch normalization

# Sources of variation

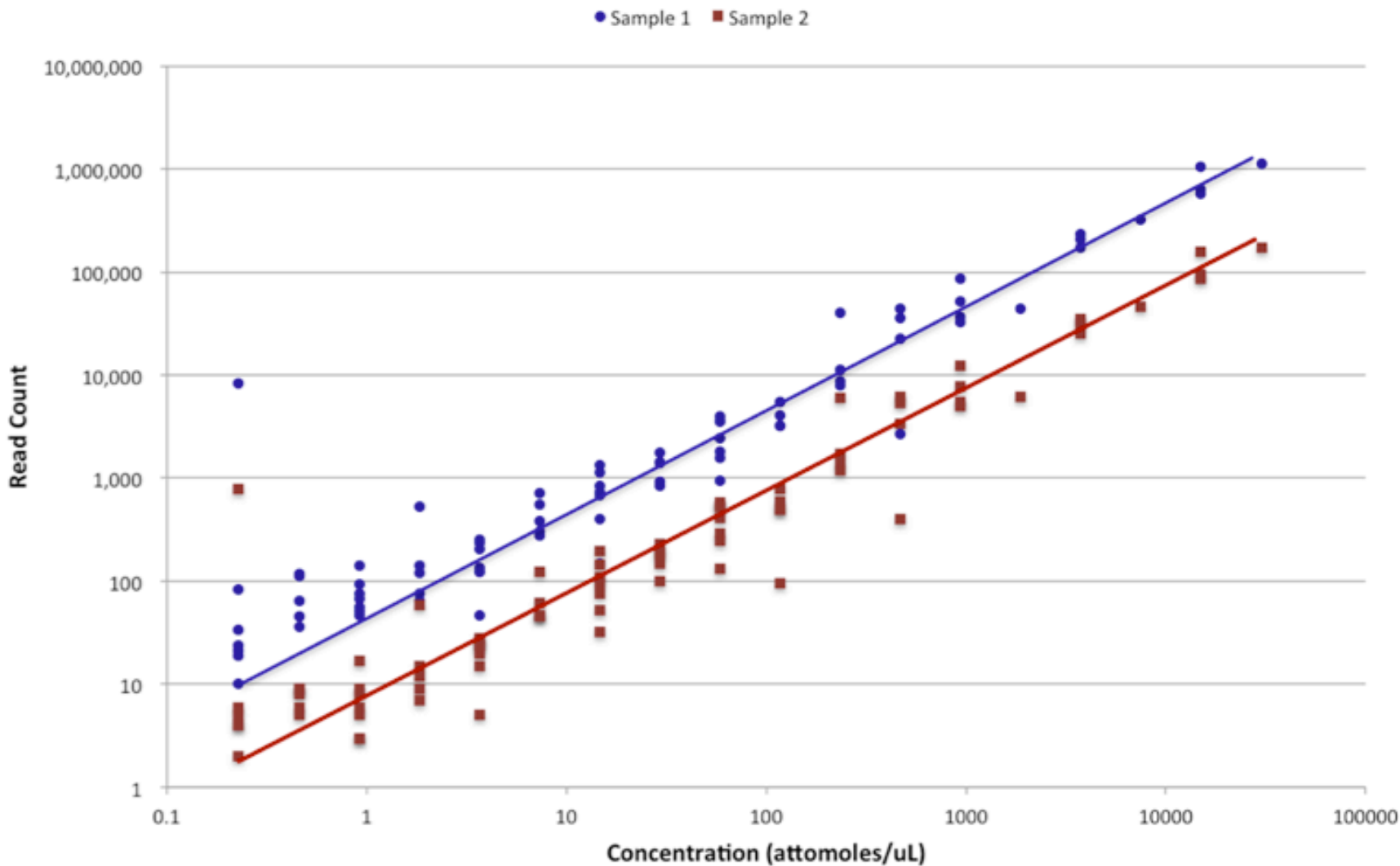
- Biological variation
  - Patient to patient variation
  - Sex
  - Time points of samples taken
  - Etc.....
- Technical variation
  - At each step of RNA extraction and library preparation

# Spike-in control RNA

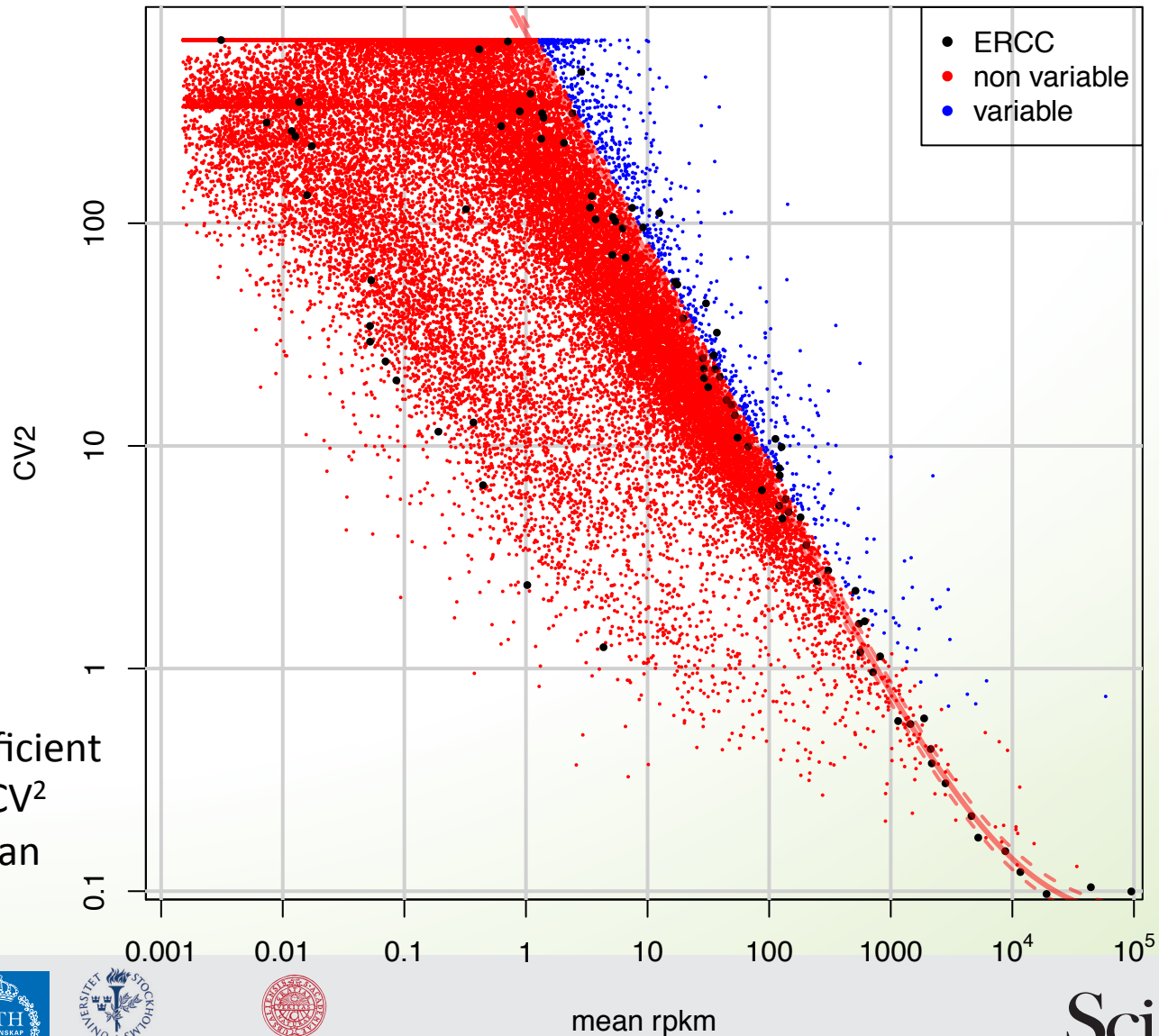
- Addition of external RNA molecules into the samples before library prep
- Will give estimate of technical variation:
  - Sensitivity / detection
  - Accuracy
  - Specific biases
- Also used to estimate amount of RNA in the samples
- Most commonly ERCC - pool of 48 or 96 synthetic mRNAs with various lengths and GC content, at 17 different concentrations
- Allows for cross comparison of datasets



# Read Count vs. ERCC Concentration



# Technical noise / Biological variation

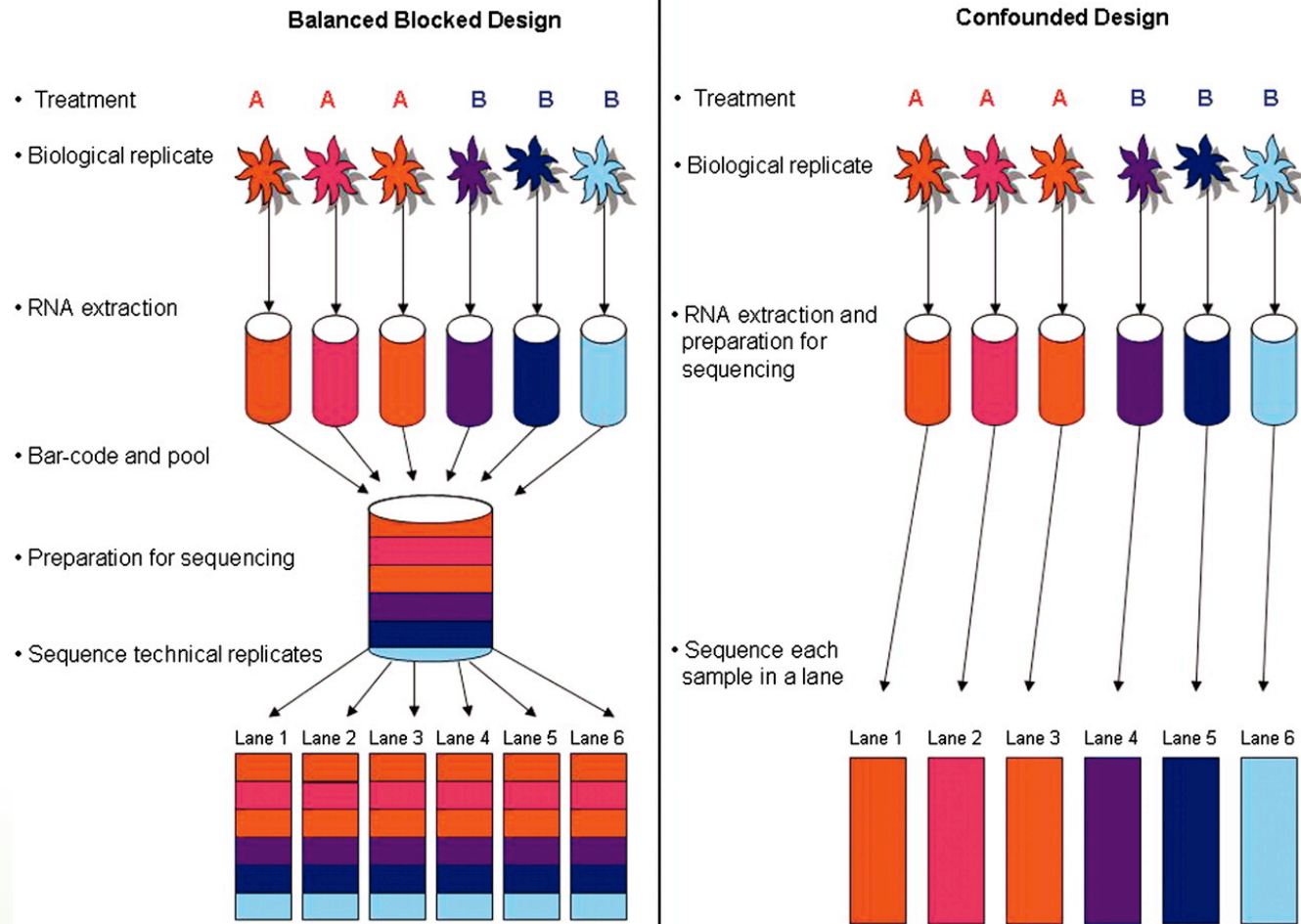


Squared coefficient of variation:  $CV^2$   
 $CV = \text{std} / \text{mean}$

# Replicates, replicates, replicates

- Technical replicates
- Biological replicates
- If you have enough material, always do extra replicates in case you want to remove low quality samples.

# Experimental Design



Copyright © 2010 by the Genetics Society of America  
DOI: 10.1534/genetics.110.114983

Statistical Design and Analysis of RNA Sequencing Data

Paul L. Auer and R. W. Doerge<sup>1</sup>

<sup>1</sup>Department of Statistics, Purdue University, West Lafayette, Indiana 47907

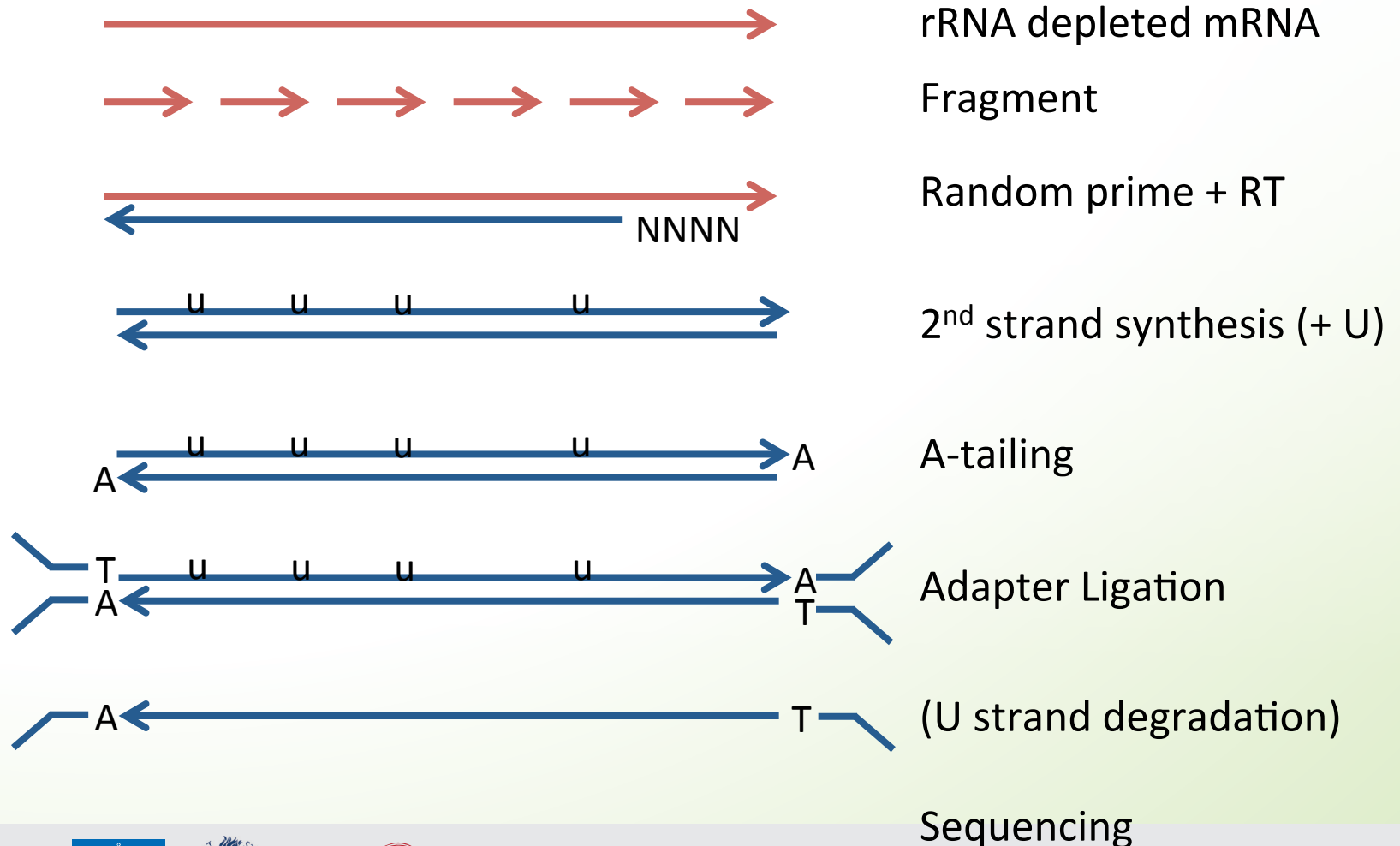
Manuscript received January 31, 2010  
Accepted for publication March 15, 2010

# Conclusions

- Good quality data is the first step in any RNA-seq experiment
- The reason for low quality samples may require some detective work
- More replicates allows you to filter out low quality libraries without losing statistical power
- Depending on where you sequence, some of the QC steps will be performed at the platform.

# Questions?

# RNA-seq libraries



# PCR duplicates – read\_duplication.py

