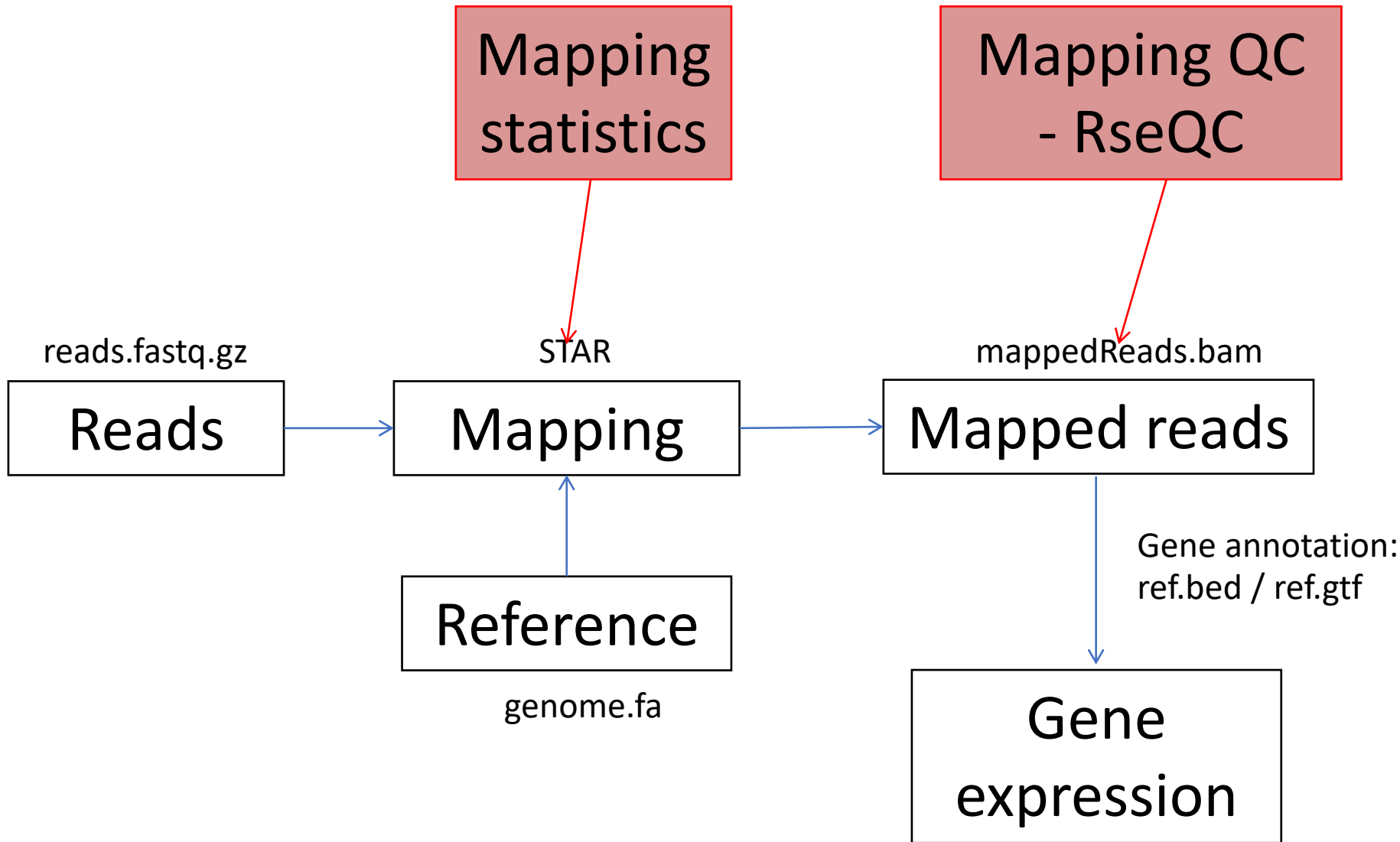
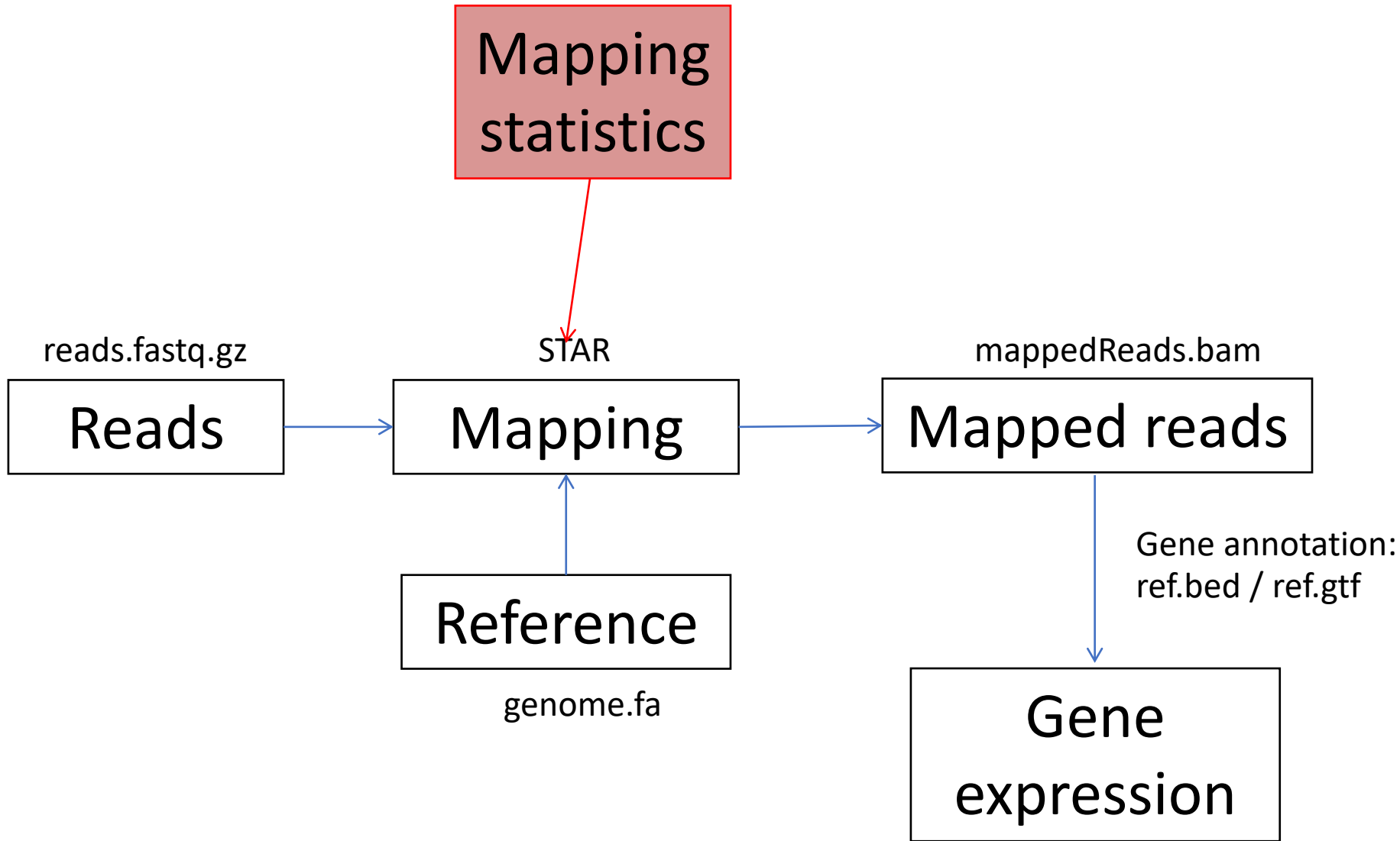


After mapping QC

RNA-seq data analysis

Johan Reimegård





Mapping logs – mapping efficiency

- Program specific how the output will be (STAR, Bowtie, BWA, Tophat...)
- Always gives:
 - % uniquely mapping – ideally around 90% for 100 bp reads
 - % multi-mapping – will depend on read length
 - % unmapped – could indicate contaminations, adaptors
- Also statistics on:
 - Mismatches / indels
 - Splice junctions

Star log example

```
[johanr@rackham3 star]$ more sample12_Log.final.out
    Started job on | May 11 20:01:21
    Started mapping on | May 11 20:02:59
    Finished on | May 11 20:10:30
Mapping speed, Million of reads per hour | 211.40

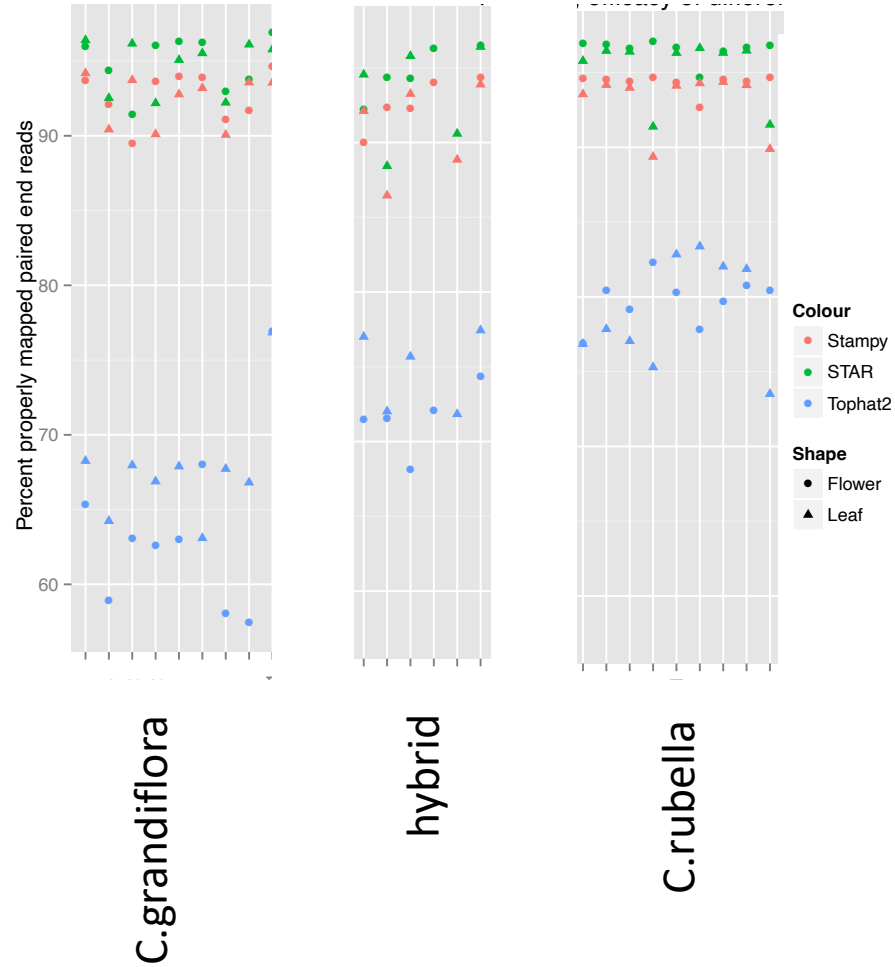
    Number of input reads | 26483380
    Average input read length | 202
    UNIQUE READS:
    Uniquely mapped reads number | 23584867
    Uniquely mapped reads % | 89.06%
    Average mapped length | 198.57
    Number of splices: Total | 15591437
    Number of splices: Annotated (sjdb) | 15442151
    Number of splices: GT/AG | 15453389
    Number of splices: GC/AG | 110331
    Number of splices: AT/AC | 13452
```

```
    Number of splices: Non-canonical | 14265
    Mismatch rate per base, % | 0.33%
    Deletion rate per base | 0.01%
    Deletion average length | 1.97
    Insertion rate per base | 0.01%
    Insertion average length | 1.36
    MULTI-MAPPING READS:
    Number of reads mapped to multiple loci | 838432
    % of reads mapped to multiple loci | 3.17%
    Number of reads mapped to too many loci | 5600
    % of reads mapped to too many loci | 0.02%
    UNMAPPED READS:
    % of reads unmapped: too many mismatches | 0.00%
    % of reads unmapped: too short | 7.73%
    % of reads unmapped: other | 0.03%
```

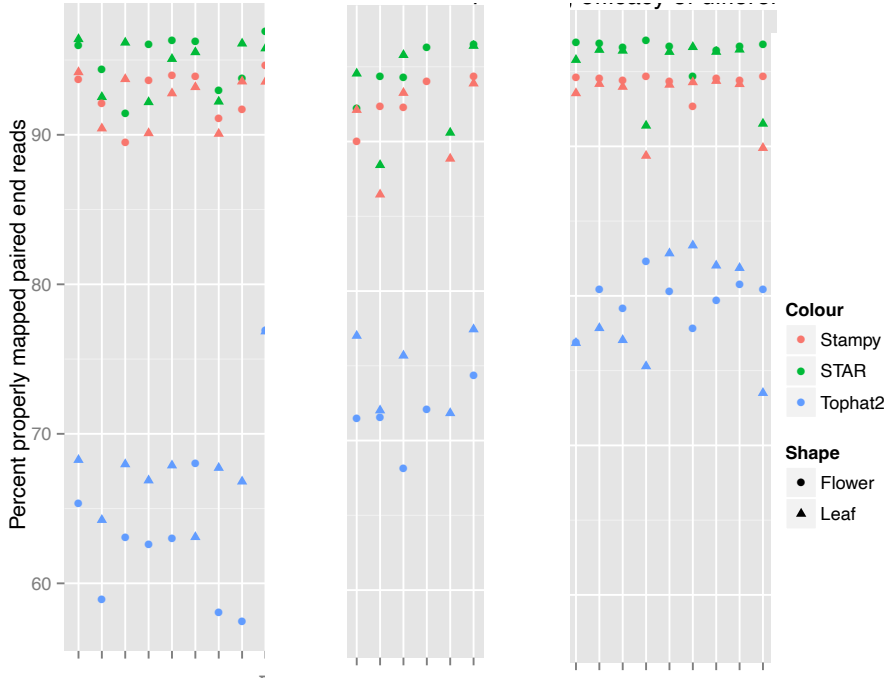
Hisat2 log example

```
13229276 reads; of these:
  13229276 (100.00%) were paired; of these:
    2258930 (17.08%) aligned concordantly 0 times
    10385753 (78.51%) aligned concordantly exactly 1 time
    584593 (4.42%) aligned concordantly >1 times
  ----
  2258930 pairs aligned concordantly 0 times; of these:
    271241 (12.01%) aligned discordantly 1 time
  ----
  1987689 pairs aligned 0 times concordantly or discordantly; of these:
    3975378 mates make up the pairs; of these:
      2915792 (73.35%) aligned 0 times
      963693 (24.24%) aligned exactly 1 time
      95893 (2.41%) aligned >1 times
88.98% overall alignment rate
```

Map log can be used to compare how well different programs work on different samples



More variation when using Tophat2 with default settings than when using STAR or Stampy with default settings



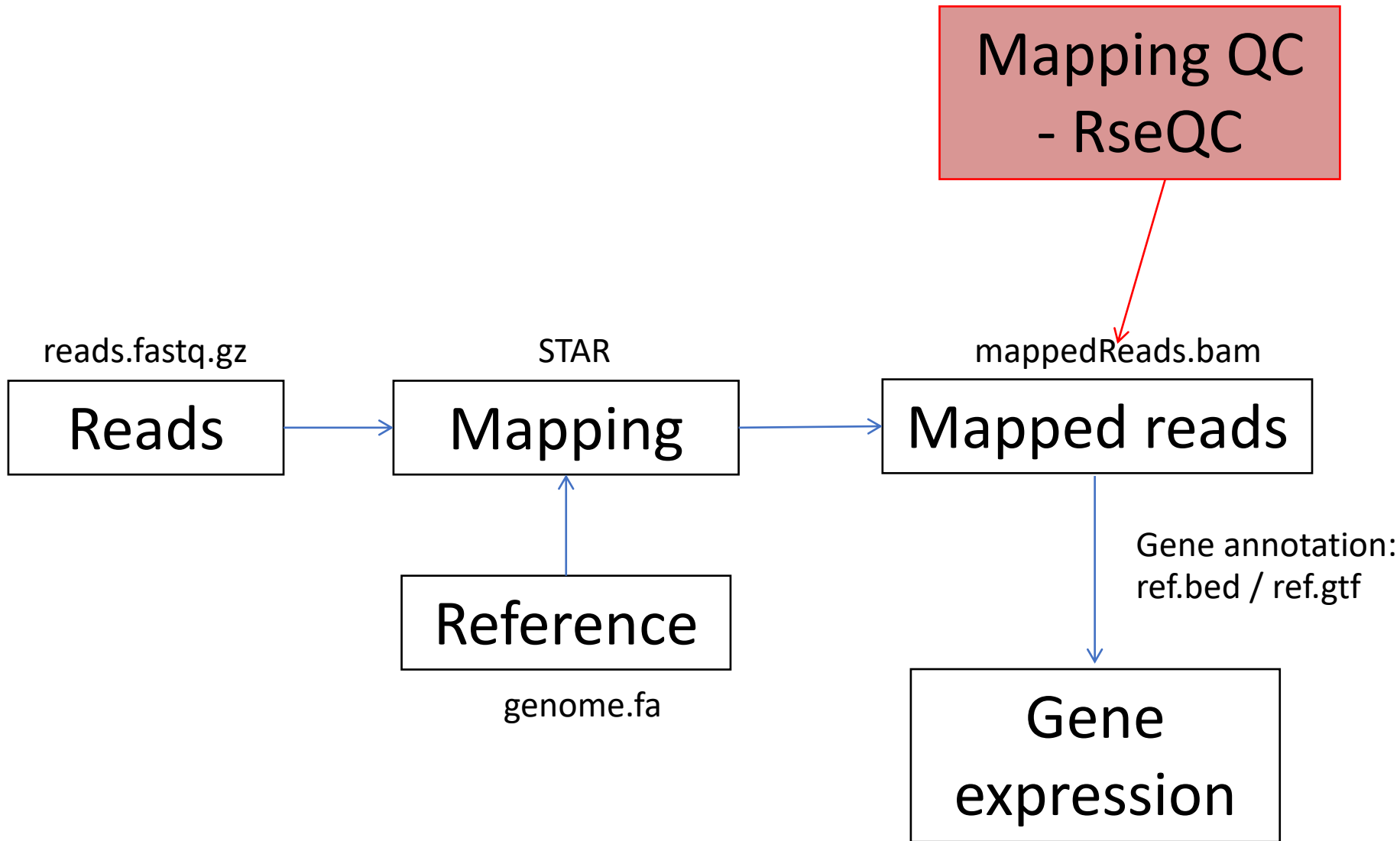
C.grandiflora

hybrid

C.rubella

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?



After mapping - RseQC package

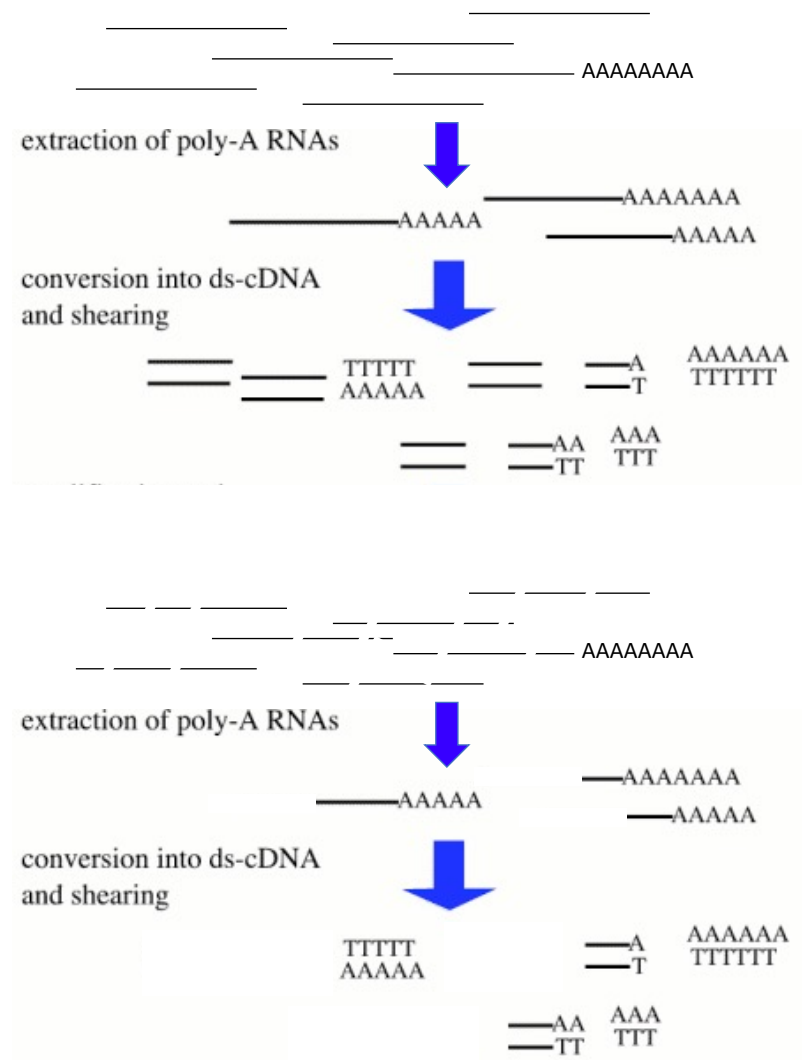
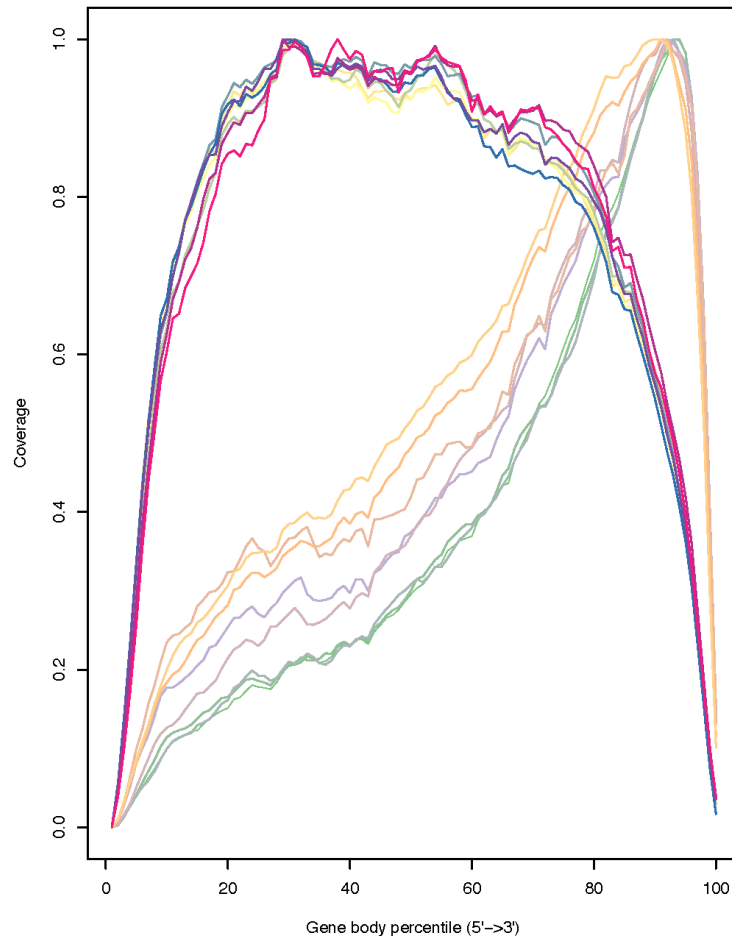
- General sequence QC:
 - sequence quality
 - nucleotide composition bias
 - PCR bias and
 - GC bias
- RNA-seq specific QC:
 - evaluate sequencing saturation
 - mapped reads distribution
 - coverage uniformity
 - strand specificity
 - Etc..
- Some tools for file manipulations

Code

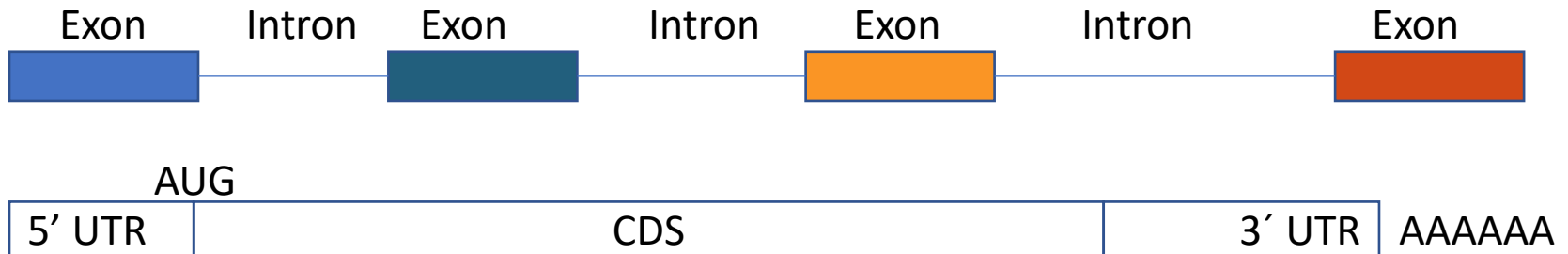
```
$ geneBody_coverage.py -r  
ref.bed12 -i mappedReads.bam -o  
genecoverage
```

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

Gene coverage - geneBody_coverage.py

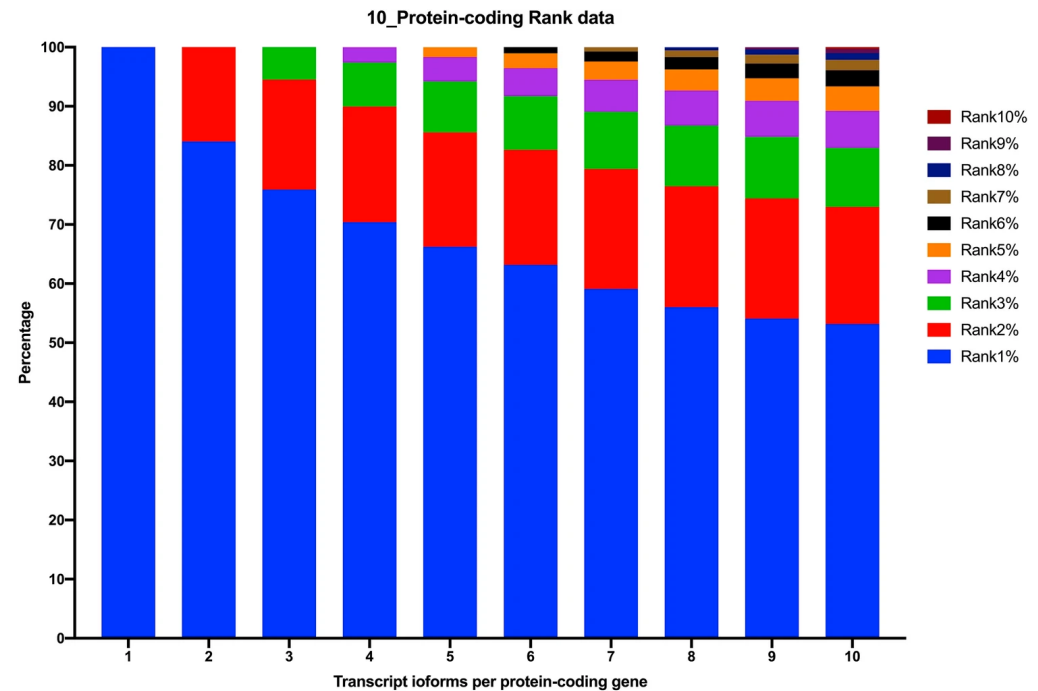
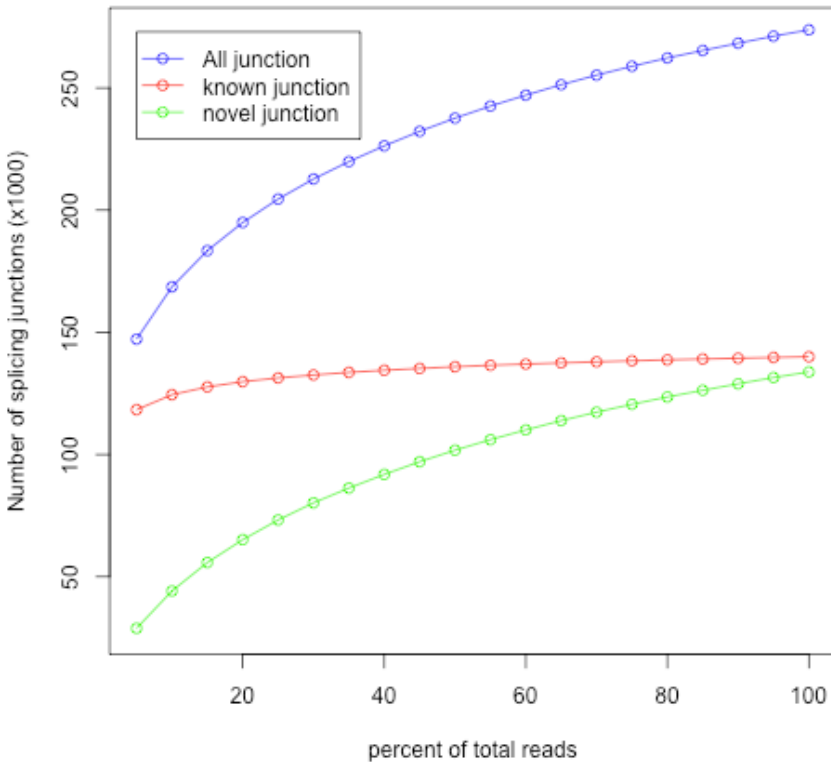


Where in the genome do your reads map? - read_distribution.py

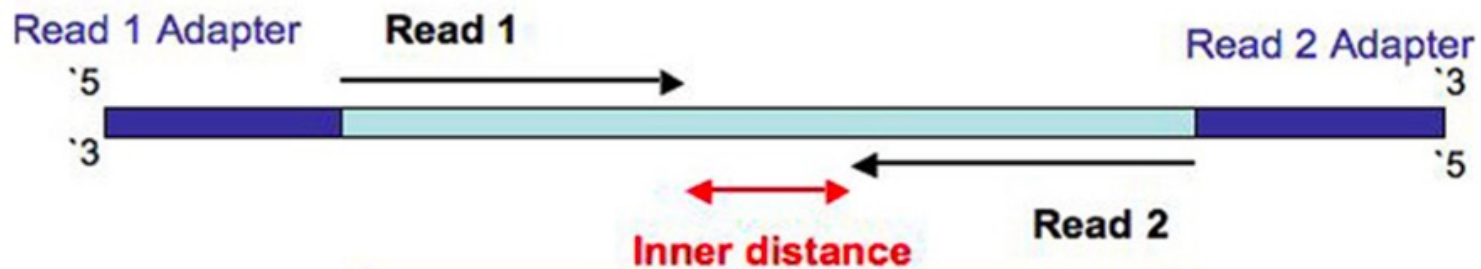
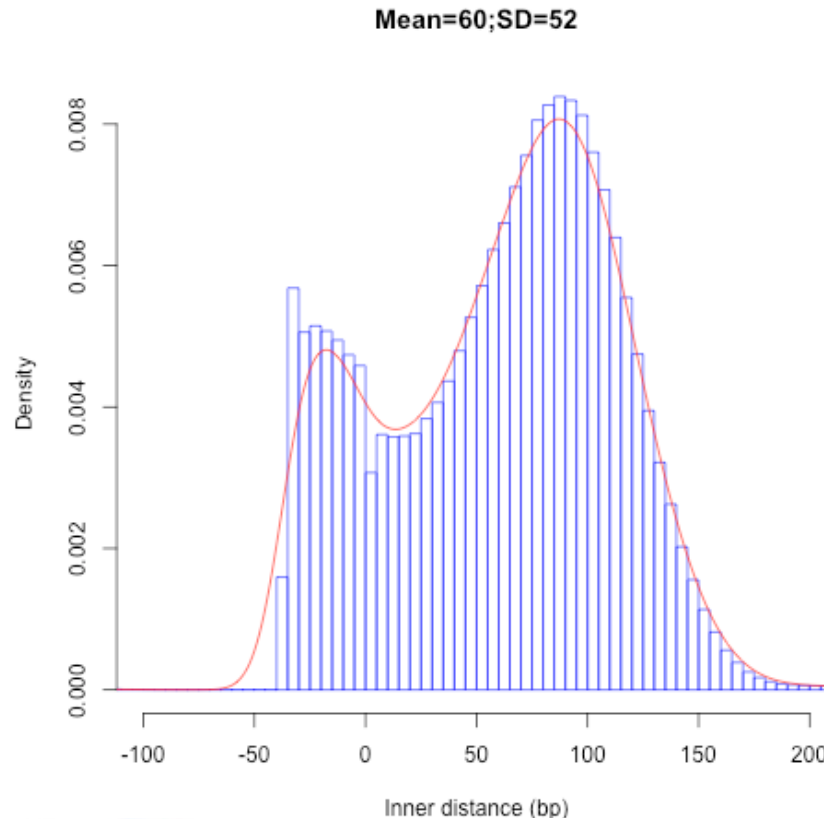


Group	Total_bases	Tag_count	Tags/Kb
CDS_Exons	33302033	20002271	600.63
5'UTR_Exons	21717577	4408991	203.01
3'UTR_Exons	15347845	3643326	237.38
Introns	1132597354	6325392	5.58
TSS_up_1kb	17957047	215331	11.99
TSS_up_5kb	81621382	392296	4.81
TSS_up_10kb	149730983	769231	5.14
TES_down_1kb	18298543	266161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



Distance between PE-reads - inner_distance.py



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.

MultiQC – summary of QC stats

The screenshot shows the MultiQC v0.8 web interface. The top navigation bar includes tabs for RNA-Seq, Whole-Genome Seq, Bisulfite Seq, Hi-C, and MultiQC_NGI. The left sidebar contains a menu of report sections: General Stats, featureCounts, STAR, Cutadapt, FastQC, Sequence Quality Histograms, Per Sequence Quality Scores, Per Base Sequence Content, Per Sequence GC Content, Per Base N Content, Sequence Length Distribution, Sequence Duplication Levels, and Adapter Content. The main content area displays the MultiQC logo, a description of the tool as a modular aggregator of bioinformatics results, and the report generation date and path. A notification banner offers a tutorial video. Below this is the 'General Statistics' section, which includes a table of data for two samples.

MultiQC
v0.8

General Stats
featureCounts
STAR
Cutadapt
FastQC
Sequence Quality Histograms
Per Sequence Quality Scores
Per Base Sequence Content
Per Sequence GC Content
Per Base N Content
Sequence Length Distribution
Sequence Duplication Levels
Adapter Content

MultiQC

A modular tool to aggregate results from bioinformatics analyses across many samples into a single report.

Report generated on 2016-09-26, 17:09 based on data in: /Users/philewels/GitHub/MultiQC_website/public_html/examples/rna-seq/data

Welcome! Not sure where to start? [Watch a tutorial video](#) (6:06) [don't show again](#)

General Statistics

Showing 8/8 rows and 7/10 columns.

Sample Name	% Assigned	M Assigned	% Aligned	M Aligned	% Trimmed
SRR3192396	67.5%	71.9	93.7%	97.8	4.0%
SRR3192397	66.6%	63.0	94.7%	87.1	3.5%
		36.5	88.2%	58.7	5.0%
		42.3	88.2%	65.6	5.0%

Code

```
$ multiqc .
```

(<http://multiqc.info/>)



Thank you.

Johan Reimegård