Next-Generation Sequencing: Quality Control

Bingbing Yuan

BaRC Hot Topics – January 2017
Bioinformatics and Research Computing
Whitehead Institute

http://barc.wi.mit.edu/hot_topics/

Why QC?

- Do you want to include the reads with low quality base calls?
- Why are so little reads mapped? Could it be because of adaptor, other species/vectors contamination?
- How is your library preparation? How to identify samples with low library complexity? For RNA-seq, are the high gene expression levels due to real biological signal or to PCR artefacts ?

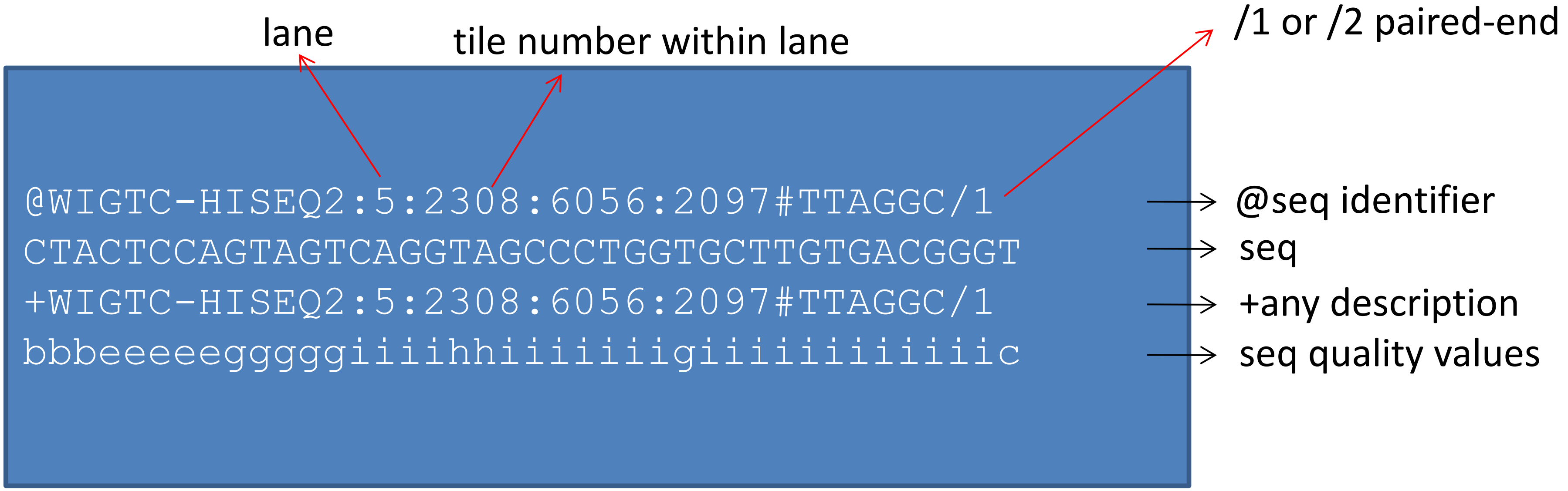
QC

- Before mapping:
  - How to identify and remove reads with low base calls?
  - How to identify and remove reads with linkers/adaptors ?
  - How to screen for potential species/vector/ribosomal contamination?
  - How is your library complexity?
- After Mapping:
  - What is percentage of reads aligned?
  - Is your sequencing library stranded or unstranded?
  - How could I know if the high expression levels are due to real biological signal or to PCR artefacts?

Illumina data format

- Fastq format:

http://jura.wi.mit.edu/genomecorewiki/index.php/SequencingFormats



```
@WIGTC-HISEQ2:5:2308:6056:2097#TTAGGC/1
CTACTCCAGTAGTCAGGTAGCCCTGGTGCTTGTGACGGGT
+WIGTC-HISEQ2:5:2308:6056:2097#TTAGGC/1
bbbeeeeeggggiiiihhiiiiiiigiiiiiiiiiiiic
```

Input qualities	Illumina versions
--solexa-quals	<= 1.2
--phred64	1.3-1.7
--phred33	>= 1.8

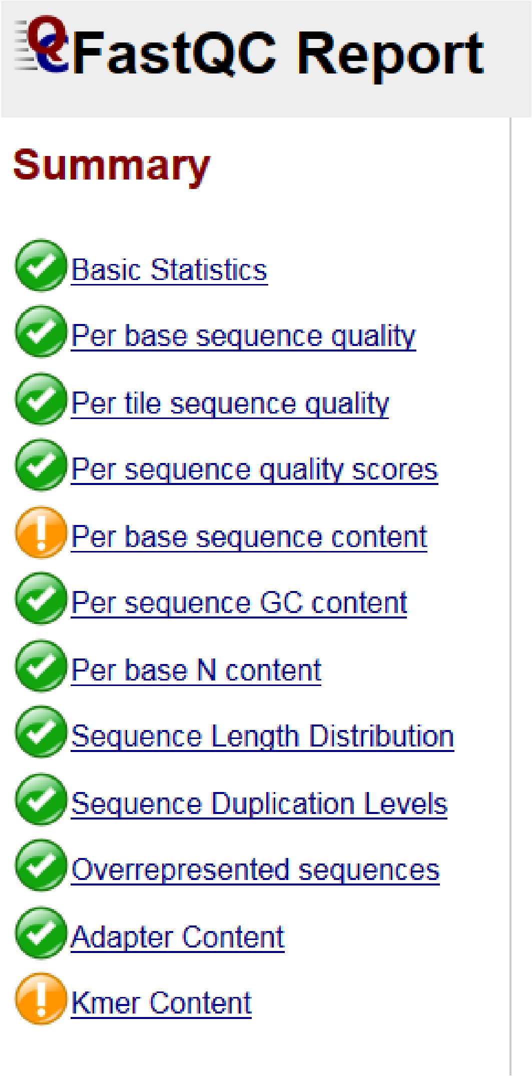
Check read quality with fastqc

(http://www.bioinformatics.babraham.ac.uk/projects/fastqc/)

1. Run fastqc to check read quality

   ```
   $ bsub –q 14 fastqc sample.fastq
   ```

2. Open output file: "fastqc_report.html"



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

Output from fastqc

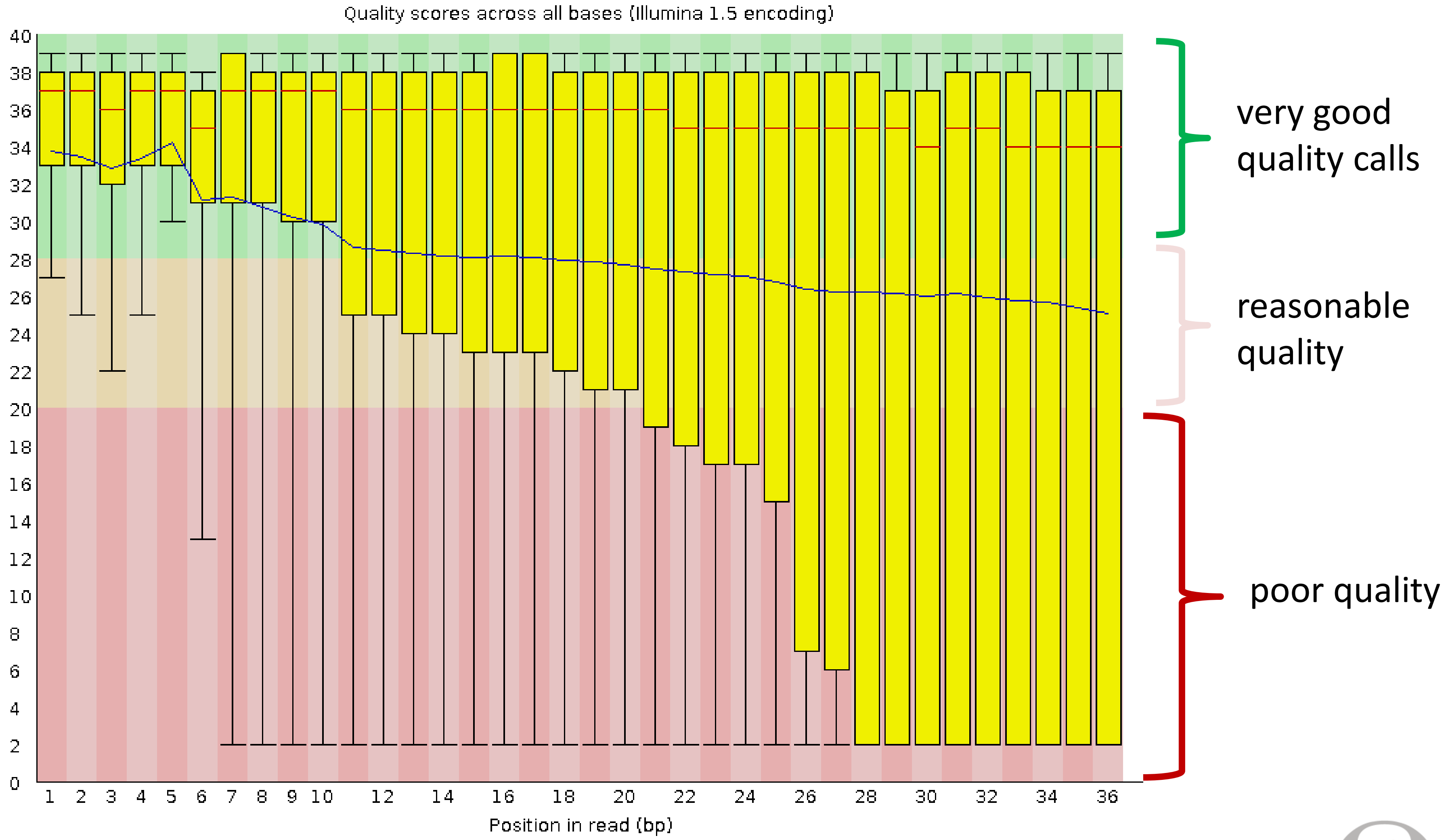
Basic Statistics

Measure	Value
Filename	sample.fastq
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	9053
Filtered Sequences	0
Sequence length	36
%GC	50

We have to know the quality encoding to use the appropriate parameter in the mapping step.

Note: sample.fastq is 0.05% of original fastq

FastQC: per base sequence quality



Red: median blue: mean yellow: 25%, 75% whiskers: 10%, 90%

Whitehead Institute

remove reads with lower quality

```
$ fastq_quality_filter –h  # usage information
$  bsub –q 14 fastq_quality_filter -v -q 20 -p 75 -i sample.fastq -o sample_good.fastq
```

-i: input file
-o: output file
-v: report number of sequences

-q: Minimum quality score
-p: Minimum percent of bases that must have [-q] quality

Check job status:

```
$ bjobs
```

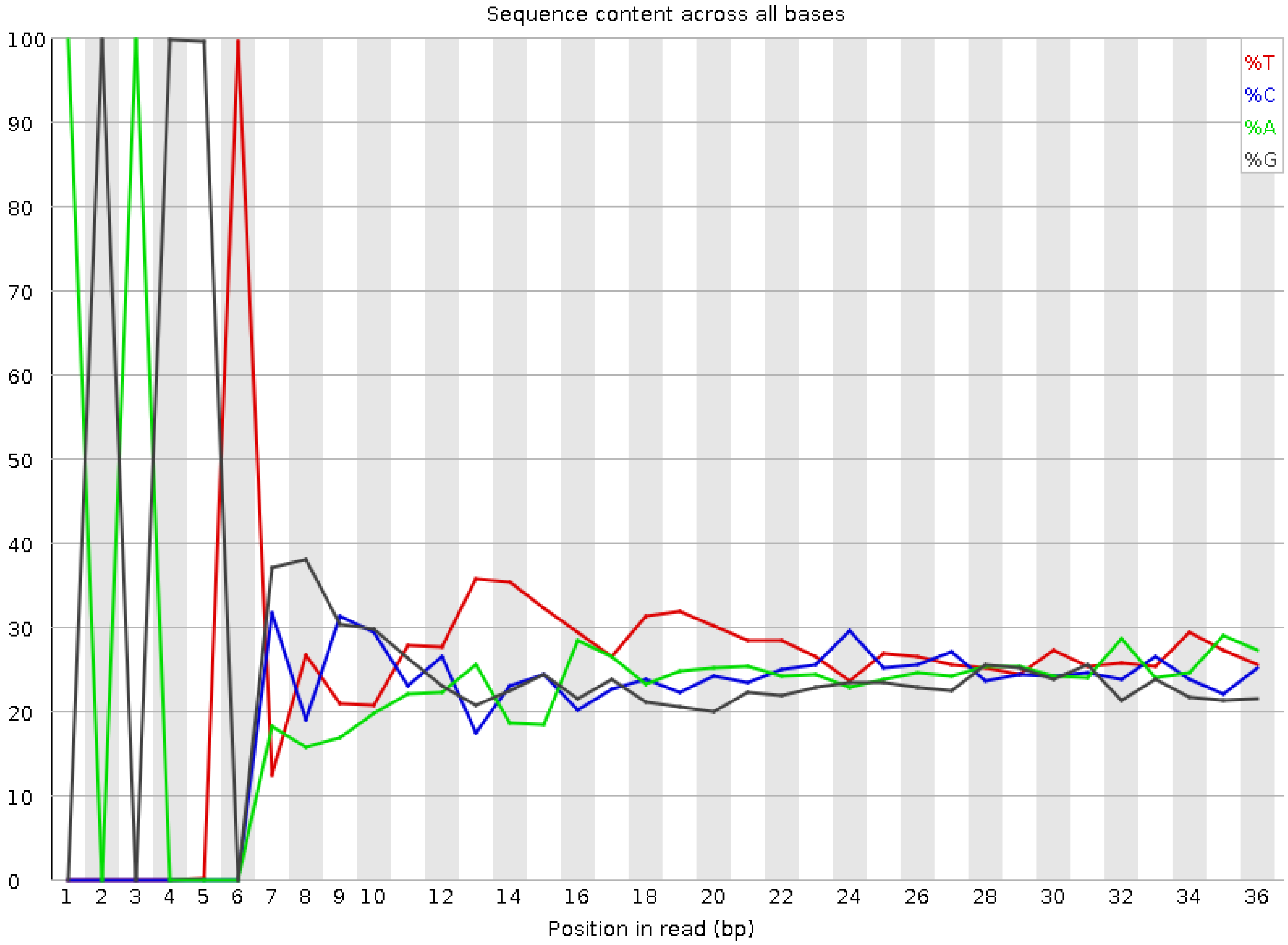
Look at your email to see the number of discarded reads

Problem solved? Re-run quality control on filtered reads:

```
$ bsub fastqc sample_good.fastq
```

Use your browser to look at the sample_good_fastqc.html

Output from fastqc



About 100% of the first six bases are AGAGGT

Trim the read sequence

# delete the first 6nt from 5’

-f: First base to keep
-l: Last base to keep
-i: input file
-o: output file
-v: report number of sequences

```
$  fastx_trimmer –h  # usage information
```

```
$  bsub fastx_trimmer -v -f 7 -l 36 -i sample_good.fastq -o sample_good_trimmed.fastq
```

# Problem solved? Check trimmed reads

```
$ bsub fastqc sample_good_trimmed.fastq
```

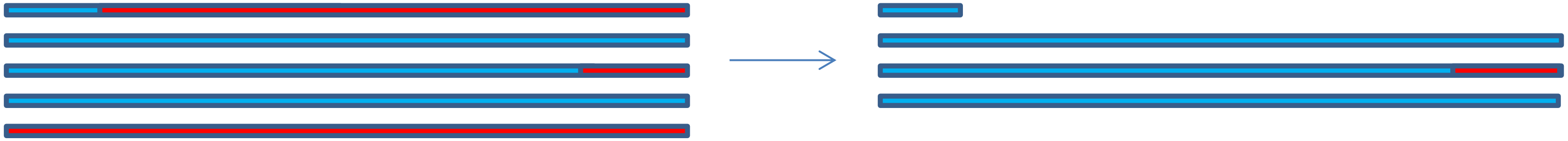
Use your browser on your laptop to look at the sample_good_trimmed_fastqc.html

Output from fastqc can show you the adaptor contamination

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACTTAGGCA	7360116	82.88507591015895	RNA PCR Primer, Index 3 (100% over 40bp)
GCGAGTGCGGTAGAGGGTAGTGGAATTCTCGGGTGCCAAG	541189	6.094535921273932	No Hit
TCGAATTGCCTTTGGGACTGCGAGGCTTTGAGGACGGAAG	291330	3.2807783416601866	No Hit
CCTGGAATTCTCGGGTGCCAAGGAACTCCAGTCACTTAGG	210051	2.365464495397192	RNA PCR Primer, Index 3 (100% over 38bp)

Remove adapter/Linker



```
$ cutadapt  # usage
$ bsub " cutadapt -a TGGAATTCTCGGGTGCCAAGGAACTCCAGTCACTTAGGCA  foo.fastq |
fastx_artifacts_filter  > no_adapter.fastq"
```

-a: Sequence of an adapter that was ligated to the 3' end.

-e : max. error rate (default =0.1)

fastx_artifacts_filter: filter reads with all but 3 identical bases

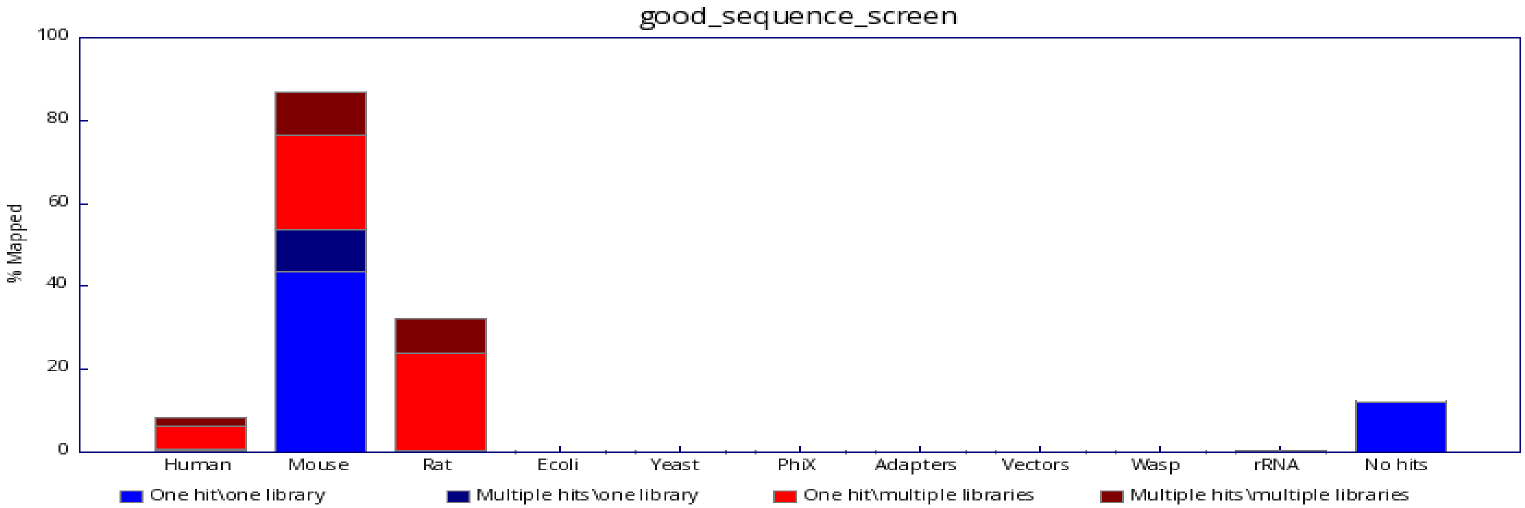
cutadapt: http://code.google.com/p/cutadapt/

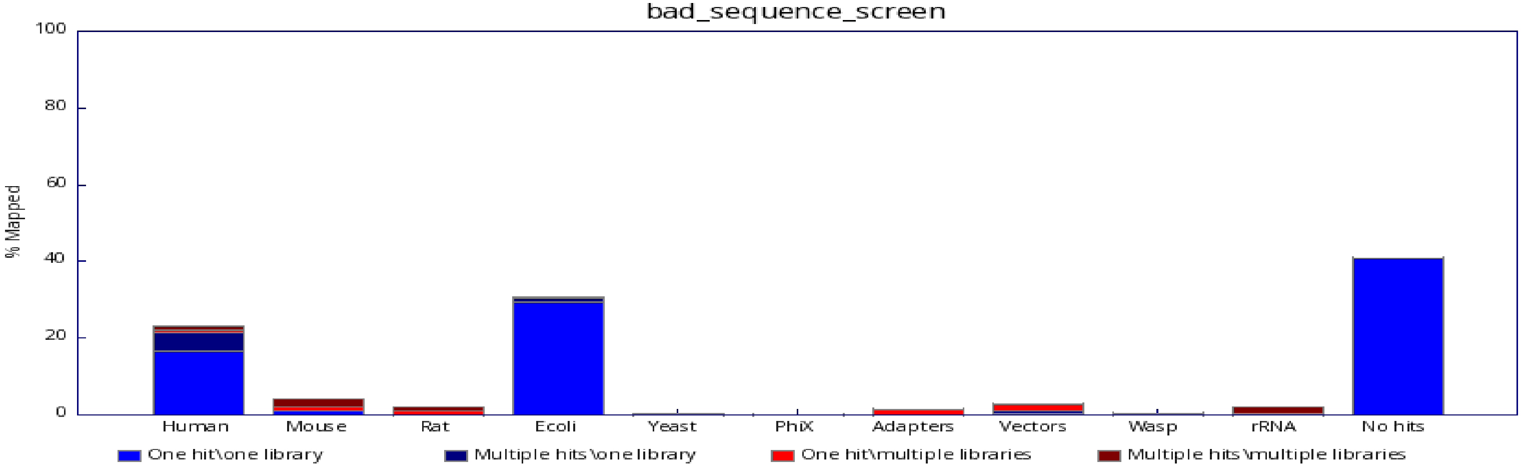
fastx artifacts filter: http://hannonlab.cshl.edu/fastx_toolkit/galaxy.html#fastx_artifacts

Recommendation for preprocessing

- Treat all the samples the same way.
- Watch out for preprocessing that may result in very different read length in the different samples as that can affect mapping.
- If you have paired-end reads, make sure you still have both reads of the pair after the processing is done.
- Run fastqc on the processed samples to see if the problem has been removed.

Identify species/vectors contamination fastq_screen:





http://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/

fastq_screen

- $ bsub fastq_screen --illumina1_3 --aligner bowtie2 sample_good_trimmed.fastq

  --aligner Specify 'bowtie' or bowtie2' to use for the mapping

  --illumina1_3 Assume that the quality values are in encoded in Illumina v1.3 format. Defaults to Sanger.

  --conf Manually specify a location for the configuration file to be used for this run.

  On local tak server: /usr/local/bin/fastq_screen.conf contains the source index files. You can create your own configuration file to include other potential contamination.

```
DATABASE    Human   /nfs/genomes/human_hg38_dec13_no_random/bowtie/hg38
DATABASE    rRNAs_human    /nfs/genomes/human_rRNAs/rRNAs
DATABASE    Mouse   /nfs/genomes/mouse_mm10_dec_11_no_random/bowtie/mm10
DATABASE    Arabidopsis    /nfs/genomes/a.thaliana_TAIR_10/bowtie/tair10
DATABASE    Fly    /nfs/genomes/d.melanogaster_apr_06/bowtie/dm3
DATABASE    Fish   /nfs/genomes/zfish_danRer7/bowtie/danRer7
DATABASE    Yeast   /nfs/genomes/sgd_2010/bowtie/sacCer3
DATABASE    PhiX174 /nfs/genomes/phiX174/phiX174
DATABASE    Ecoli   /nfs/genomes/e_coli/e_coli
DATABASE    adapters_FastQC /nfs/genomes/NGS_adapters_primers/adapters_primers
```

QC

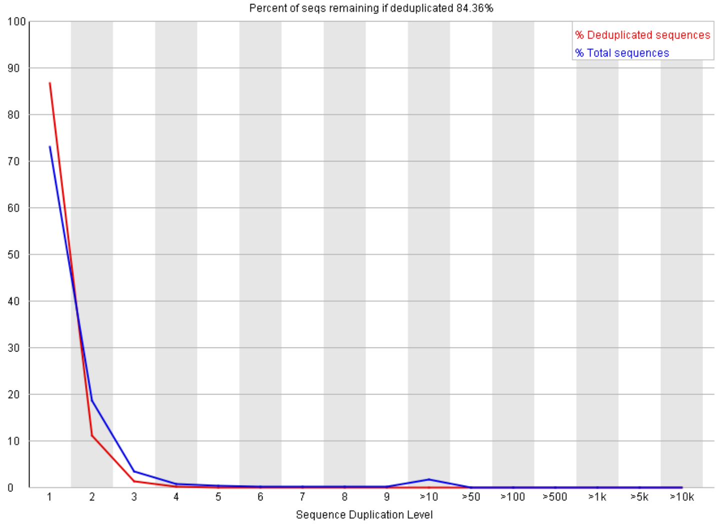
- Before mapping:
  - **How to identify and remove reads with low base calls?**
  - **How to identify and remove reads with linkers/adaptors ?**
  - **How to screen for potential species/vector/ribosomal contamination?**
  - How is your library complexity?
- After Mapping:
  - What is percentage of reads aligned?
  - Is your sequencing library stranded or unstranded?
  - How could I know if the high expression levels are due to real biological signal or to PCR artefacts?

Library Complexity

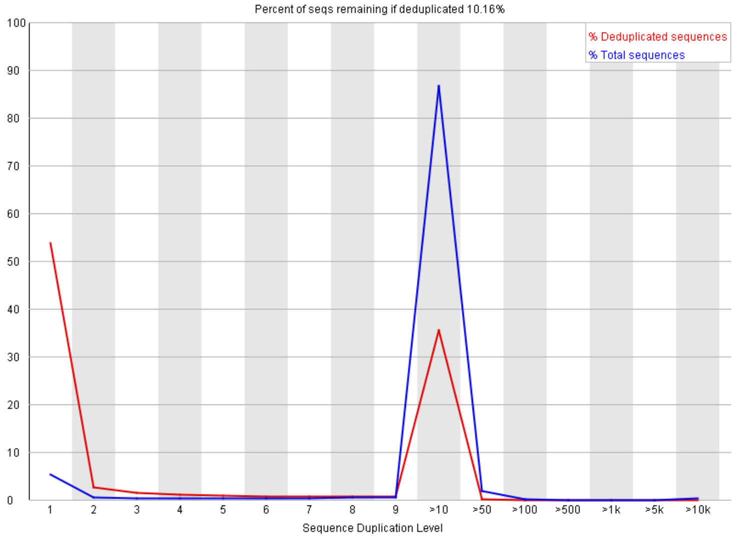
- Reasons: low-input; single cell RNA-seq
- Low library complexity may be an indicator that:
  - A new sample and a new library should be prepared.
  - We can not sequence the same sample anymore because we will not find new sequences.
- Chip-seq:
  - we have to find a better Ab to perform the IP.
  - In certain experimental settings we may expect a low library complexity. *i.e.* We are profiling a protein that binds to a small subset of the genome.

Library Complexity

- Refers to the fraction of unique fragments present in a given library.
- One proxy for library complexity is to look at the sequence duplication levels on the FastQC report:



% Complexity
85.6



% Complexity
4.95

QC

- **Before mapping:**
  - **How to identify and remove reads with low base calls?**
  - **How to identify and remove reads with linkers/adaptors ?**
  - **How to screen for potential species/vector/ribosomal contamination?**
  - **How is your library complexity?**
- After Mapping:
  - What is percentage of reads aligned?
  - Is your sequencing library stranded or unstranded?
  - How could I know if the high expression levels are due to real biological signal or to PCR artefacts?

Mapping Statistics

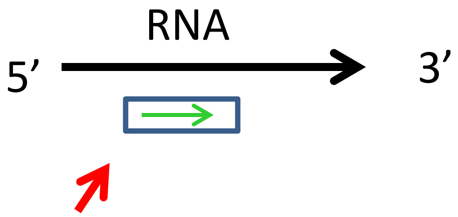
- RSeQC (http://rseqc.sourceforge.net/)
  - $ bam_stat.py –i sample.bam > sample_stat.txt

```
Total records:                              10000

QC failed:                                 0
Optical/PCR duplicate:                     0
Non primary hits                           2121
Unmapped reads:                            0
mapq < mapq_cut (non-unique):               507

mapq >= mapq_cut (unique):                  7372
Read-1:                                    3845
Read-2:                                    3527
Reads map to '+':                          5936
Reads map to '-':                          1436
Non-splice reads:                          7067
Splice reads:                              305
Reads mapped in proper pairs:              4566
Proper-paired reads map to different chrom:0
```

Is your library stranded or not stranded?

- Checking library preparation protocol
- Mapping your RNA-seq reads as if they were non-strand specific, “guess” how RNA-seq reads were stranded.
  - RSeQC (http://rseqc.sourceforge.net/)
  - $ infer_experiment.py -i sample.bam –r gene_model.bed
    - comparing the “strandness of reads” with the “strandness of transcripts”.
    - The “strandness of reads” is determined from alignment
    - the “strandness of transcripts” is determined from annotation.

Infer_experiment.py single-end RNA-seq



Two different ways to strand reads:

i) ++,--

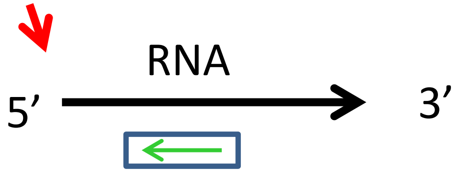
read mapped to '+' strand indicates parental gene on '+' strand

read mapped to '-' strand indicates parental gene on '-' strand

ii) +-,-+

read mapped to '+' strand indicates parental gene on '-' strand

read mapped to '-' strand indicates parental gene on '+' strand



Strand-specific example:

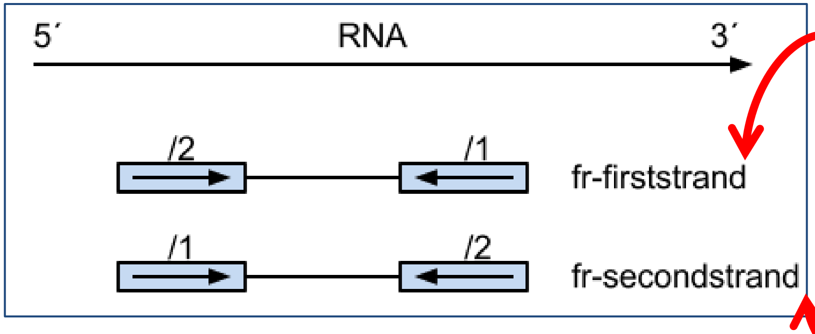
```
Fraction of reads failed to determine: 0.0170
Fraction of reads explained by "++,--": 0.9669
Fraction of reads explained by "+-,-+": 0.0161
```

Infer_experiment.py
pair-end RNA-seq

Ways to strand reads:

1+-,1-+,2++,2–

read1 mapped to ‘+’ strand indicates parental gene on ‘-‘ strand
read1 mapped to ‘-‘ strand indicates parental gene on ‘+’ strand
read2 mapped to ‘+’ strand indicates parental gene on ‘+’ strand
read2 mapped to ‘-‘ strand indicates parental gene on ‘-‘ strand



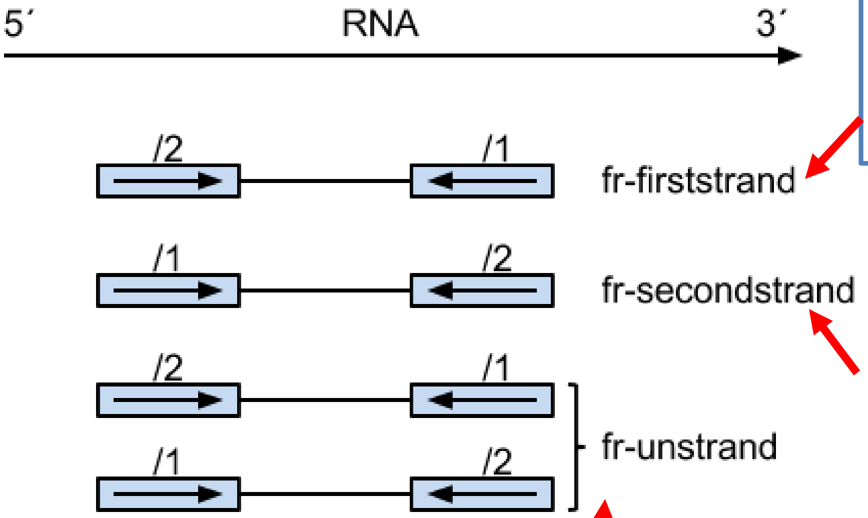
1++,1–,2+-,2-+

read1 mapped to ‘+’ strand indicates parental gene on ‘+’ strand
read1 mapped to ‘-‘ strand indicates parental gene on ‘-‘ strand
read2 mapped to ‘+’ strand indicates parental gene on ‘-‘ strand
read2 mapped to ‘-‘ strand indicates parental gene on ‘+’ strand

http://onetipperday.sterding.com/2012/07/how-to-tell-which-library-type-to-use.html

Infer_experiment.py
pair-end RNA-seq

http://onetipperday.sterding.com/2012/07/how-to-tell-which-library-type-to-use.html



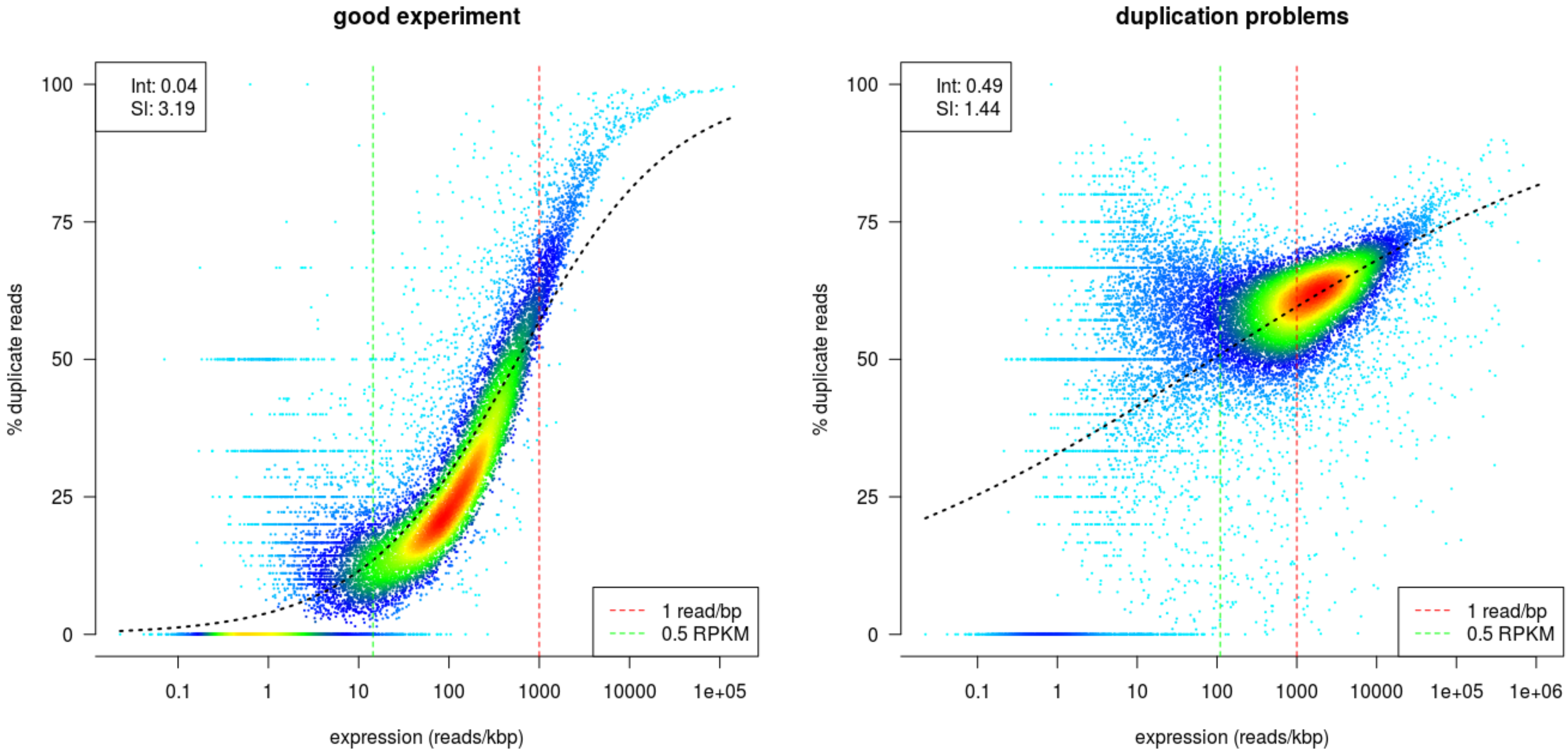
```
Fraction of reads explained by "1++,1--,2+-,2-+": 0.0193
Fraction of reads explained by "1+-,1-+,2++,2--": 0.9807
Fraction of reads explained by other combinations: 0.0000
```

```
Fraction of reads explained by "1++,1--,2+-,2-+": 0.9807
Fraction of reads explained by "1+-,1-+,2++,2--": 0.0193
Fraction of reads explained by other combinations: 0.000
```

```
Fraction of reads failed to determine: 0.0648
Fraction of reads explained by "1++,1--,2+-,2-+": 0.4590
Fraction of reads explained by "1+-,1-+,2++,2--": 0.4763
```

DupRadar

(https://www.bioconductor.org/packages/release/bioc/html/dupRadar.html)



DupRadar requirements:
alignment file: mark duplicates with bamutil or Picard MarkDuplicates
Gene model in gtf format
Strand information
Single or Paired reads

Whitehead Institute

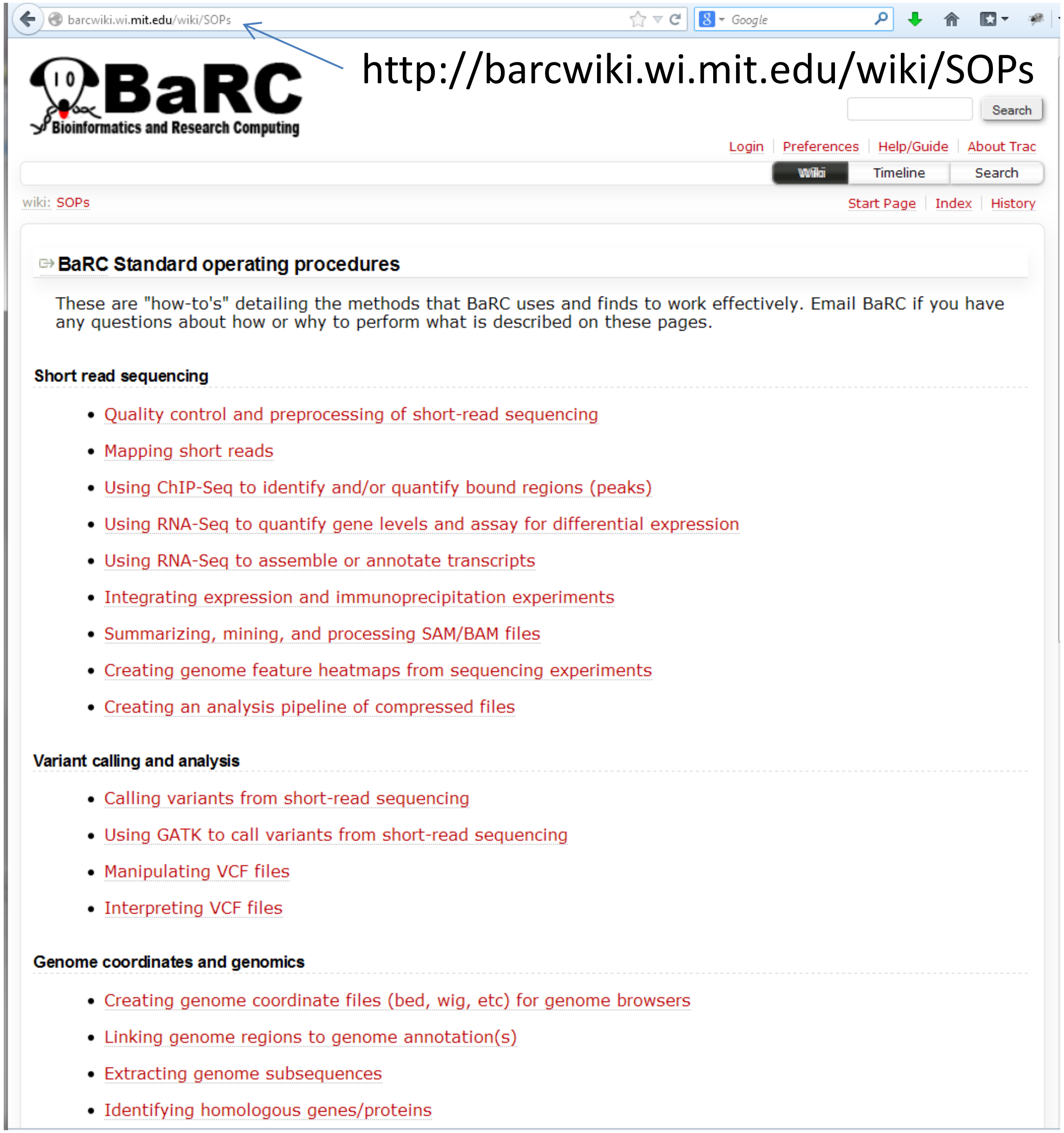
Summary

- Before mapping:
  - Quality control
    - fastqc
  - Clean up reads:
    - fastx tool kit: fastq_quality_filter, fastx_trimmer
    - Cutadapt
  - Check species/vector contamination
    - fastq_screen
- After mapping:
  - Get mapping statistics:
    - bam_stat.py
  - Check library strandness:
    - Infer_experiment.py
  - Check sequence duplication in RNA-seq samples:
    - DupRadar

References

fastqc	do quality control	http://www.bioinformatics.babraham.ac.uk/projects/fastqc
fastx Toolkit	reads editor	http://hannonlab.cshl.edu/fastx_toolkit/
	remove reads with low base call quality	fastq_quality_filter
	remove 5' or 3' reads	fastx_trimmer
	remove artifacts in reads	fastx_artifacts_filter
cutadapt	remove adaptors	https://code.google.com/p/cutadapt
fastq_screen	screen for species/vectors contamination	http://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/
RSeQC	RNAseq quality control packages	http://rseqc.sourceforge.net/
	get mapping summary	bam_stat.py
	check strandness	infer_experiment.py
DupRadar	check reads duplication	http://bioconductor.org/packages/release/bioc/html/dupRadar.html
QualiMap	mapping summary, coverage distribution	http://qualimap.bioinfo.cipf.es/
Picard	package with functions on NGS data	https://broadinstitute.github.io/picard/
	Insert size of PE reads	CollectInsertSizeMetrics
	coverage across transcripts	CollectRnaSeqMetrics
multiQC	merge QC reports	http://multiqc.info/

BaRC Standard operating procedures



barcwiki.wi.mit.edu/wiki/SOPs
Google
BaRC
Bioinformatics and Research Computing
Search
Login | Preferences | Help/Guide | About Trac
Wiki
Timeline
Search
wiki: SOPs
Start Page | Index | History

http://barcwiki.wi.mit.edu/wiki/SOPs

BaRC Standard operating procedures

These are "how-to's" detailing the methods that BaRC uses and finds to work effectively. Email BaRC if you have any questions about how or why to perform what is described on these pages.

Short read sequencing

- Quality control and preprocessing of short-read sequencing
- Mapping short reads
- Using ChIP-Seq to identify and/or quantify bound regions (peaks)
- Using RNA-Seq to quantify gene levels and assay for differential expression
- Using RNA-Seq to assemble or annotate transcripts
- Integrating expression and immunoprecipitation experiments
- Summarizing, mining, and processing SAM/BAM files
- Creating genome feature heatmaps from sequencing experiments
- Creating an analysis pipeline of compressed files

Variant calling and analysis

- Calling variants from short-read sequencing
- Using GATK to call variants from short-read sequencing
- Manipulating VCF files
- Interpreting VCF files

Genome coordinates and genomics

- Creating genome coordinate files (bed, wig, etc) for genome browsers
- Linking genome regions to genome annotation(s)
- Extracting genome subsequences
- Identifying homologous genes/proteins

Coming up

- Introduction to Python:
  - By the authors of “Python For The Life Sciences” book
  - Jan. 24th and 25th @1pm
  - Registration required