Before we start

Unix commands (after $)

purple highlighted (exercise at class)

Lines starting with # are comments

- In a laptop browser:

  Go to http://jura.wi.mit.edu/bio/education/hot_topics/ -> "Short Read Sequencing" -> Feb 2014

  Open your WI web mail to view results from cluster nodes

- Connect to your lab folder through laptop (see handout)

- Finish "step 0" in the Exercise sheet.

Next-Generation Sequencing: Quality Control and Mapping Reads

BaRC Hot Topics – February 2014

Bingbing Yuan, Ph.D.

http://jura.wi.mit.edu/bio/education/hot_topics

Questions I received

- Why do only a small amount of my reads map to genome?

- There is a lot of information in the output from the quality control program. What information is important?

Outline

1. Check quality control and clean up reads
2. Map reads
    1. Non-spliced alignment
    2. Spliced alignment
3. Check mapped reads:
    1. View reads in genome browser
    2. Calculate mapping statistics

Illumina data format

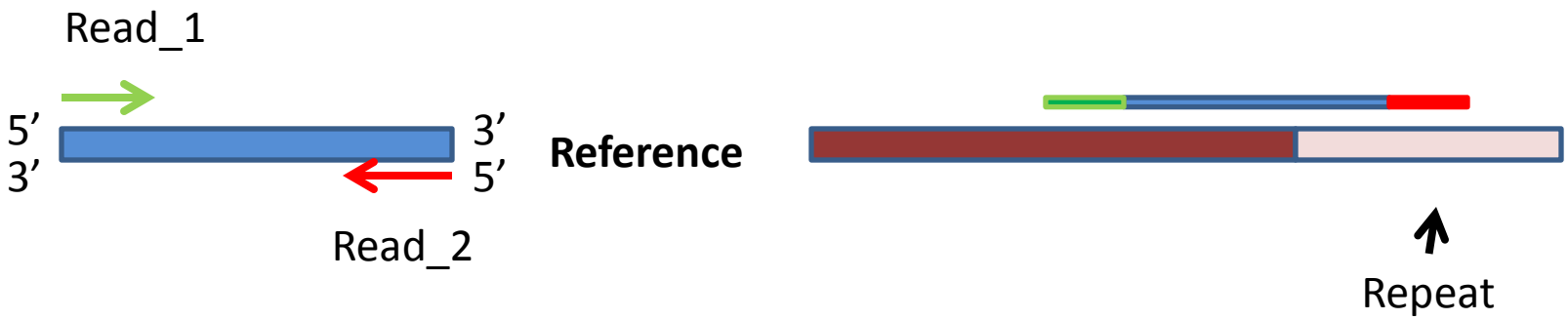
- Fastq format: (WI local file: QualityScore/s_1_sequence.txt)

```
@ILLUMINA-F6C19_0048_FC:5:1:12440:1460#0/1
GTAGAACTGGTACGGACAAGGGGAATCTGACTGTAG
+ILLUMINA-F6C19_0048_FC:5:1:12440:1460#0/1
hhhhhhhhhhghhhhhhhehhhedhhhhfhhhhhh
```

→ @seq identifier
→ seq
→ +any description
→ seq quality values

What does a fastq file look like?
# display the top 10 lines of sample_reads.fastq
$ head sample.fastq
Or
# view the contents of sample_reads.fastq one screen at a time
$ more sample.fastq

Paired-end reads



- Local filename format:

  s_7_1_sequence.txt, s_7_2_sequence.txt

Check read quality

1. Run fastqc to check read quality

   $ bsub fastqc sample.fastq

2. check job status

   $ bjobs

3. You will be notified by email on the job status: done or exit

4. Look at fastqc_report.html in output folder suffixed with _fastqc

   Use your browser on your laptop to look at fastqc_report.html under _fastqc folder

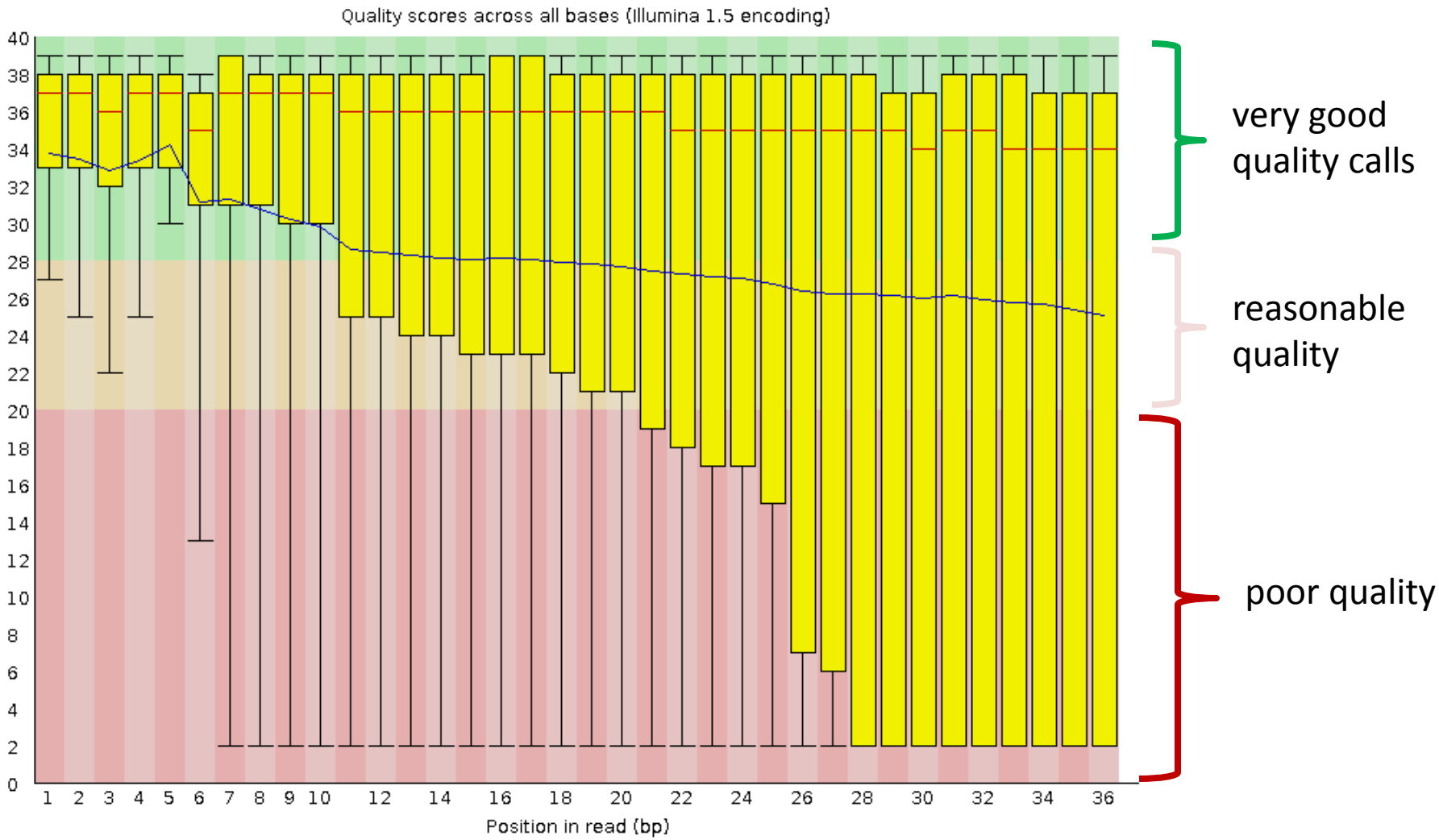
Output from fastqc

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

Note: sample.fastq is 0.05% of original fastq

Output from fastqc



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

Remove reads with lower quality

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

$ fastq_quality_filter –h # usage information

$ bsub fastq_quality_filter -v -q 20 -p 75 -i sample.fastq -o sample_good.fastq

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

Check job status:

$ bjobs

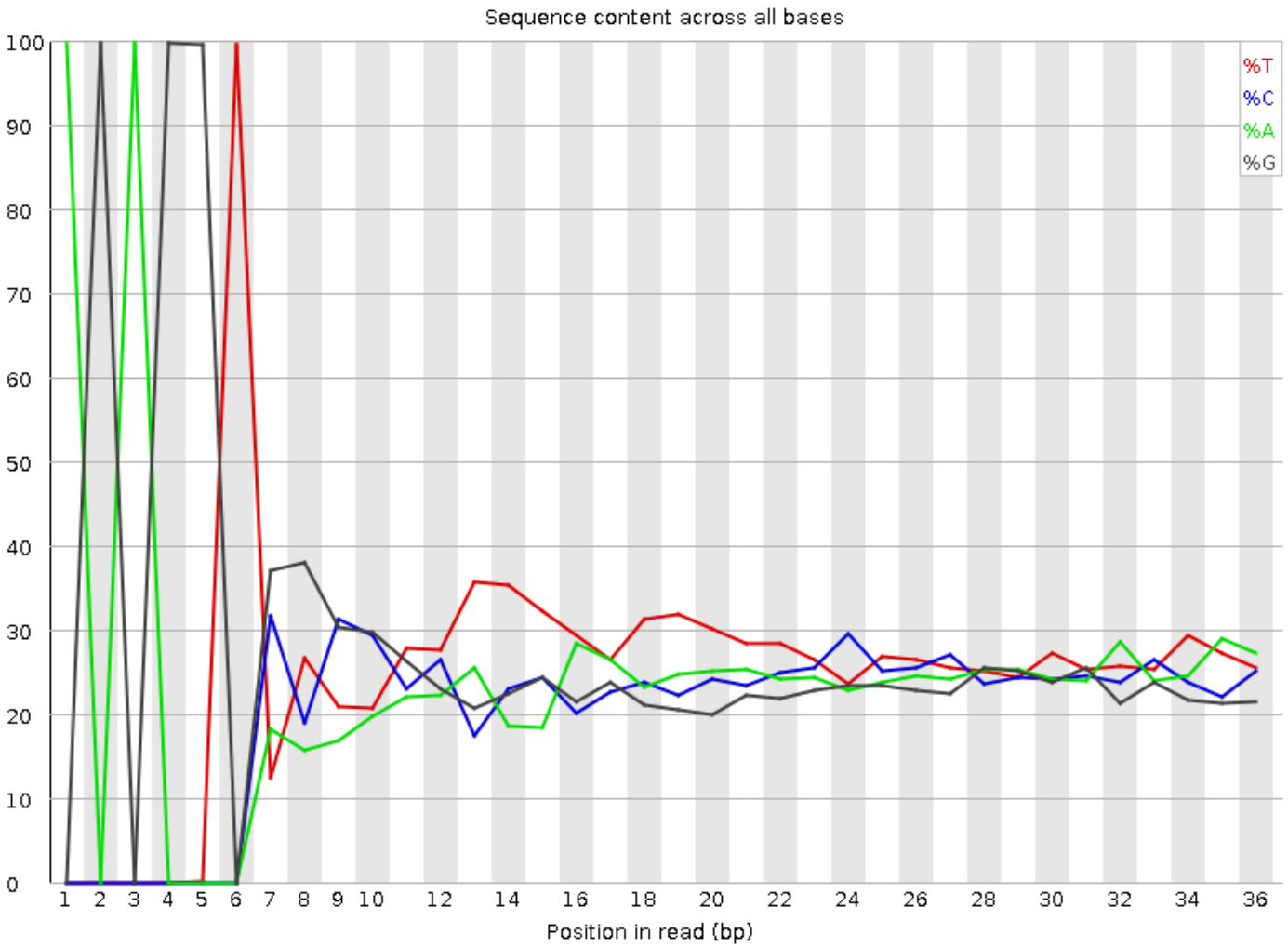
Look at your email to see the number of discarded reads

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

$ bsub fastqc sample_good.fastq

Use your browser to look at the fastqc_report.html under sample_good_fastqc folder

Output from fastqc



About 100% of the first six bases are AGAGGT

Output from fastqc

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AGAGGTCTGGTTAGTTTCTTTTCCTCCGCTGACTAA	92	1.3878413033640065	No Hit
AGAGGTCGGGCGGTGTGTACAAAGGGCAGGGACTTA	77	1.1615628299894403	No Hit
AGAGGTGTTAGTTTCTTTTCCTCCGCTGACTAATAT	65	0.9805400512897873	No Hit
AGAGGTGTTTCTTTTCCTCCGCTGACTAATATGCTT	40	0.6034092623321768	No Hit

Kmer Content

Sequence	Count	Obs/Exp Overall	Obs/Exp Max	Max Obs/Exp Position
AGAGG	6800	21.456266	666.39294	
GAGGT	6675	20.988543	663.04395	
AGGTC	2160	9.3512945	279.39185	
AGGTG	2615	8.222478	254.1584	

Basic Statistics

Measure	Value
Filename	sample_good.fastq
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	6629

Trim the read sequence

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

Delete the first 6nt from 5'

```
$  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 fastqc_report.html under sample_good_trimmed_fastqc folder

Output from fastqc

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 –o
no_adapter.fastq exp.fastq
```

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

-o: output file name

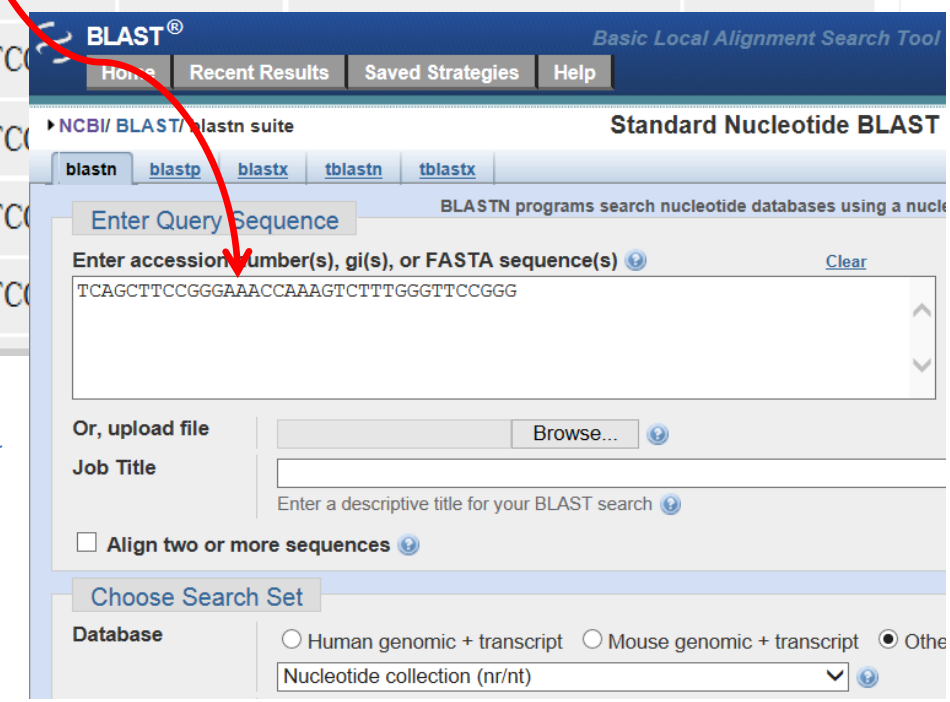
-e : max. error rate (default =0.1)

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

How to find more information on overrepresented reads

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
TCAGCTTCCGGGAAACCAAAGTCTTTGGGTTCCGGG	1224636	5.575549525595965	No Hit
TAAGCTTCCGGGAAACCAAAGTCTTTGGGTTCCGGG	267329	1.217101309391561	No Hit
TTAGCTTCCGGGAAACCAAAGTCTTTGGGTTCCGGG	114681	0.5221221613155834	No Hit
CCAGCTTCCGGGAAACCAAAGTCTTTGGGTTC			
CAAGCTTCCGGGAAACCAAAGTCTTTGGGTTC			
AAAGCTTCCGGGAAACCAAAGTCTTTGGGTTC			
ACAGCTTCCGGGAAACCAAAGTCTTTGGGTTC			



Ribosomal RNAs?

Check quality and clean up reads Summary

- Quality control
  - fastQC: fastqc
- Remove reads with low quality:
  - fastx tool kits: fastq_quality_filter
- Trim reads
  - fastx tool kits: fastx_trimmer
- Remove adapter/linker from reads
  - Cutadapt: cutadapt

2. Map reads

1. Non-spliced alignment
2. Spliced alignment

Local genomic files

tak: /nfs/genomes/

- Human, mouse, zebrafish, C.elegans, fly, yeast, etc.
- Different genome builds
  - mm9: mouse_gp_jul_07
  - mm10: mouse_mm10_dec_11
- human_gp_feb_09 vs human_gp_feb_09_no_random?
  - human_gp_feb_09 includes *_random.fa, *hap*.fa, etc.
- Sub directories:
  - bowtie
    - Bowtie1: *.ebwt
    - Bowtie2: *.bt2
  - fasta:
  - fasta_whole_genome: all sequences in one file
  - gtf: gene models from Refseq, Ensembl, etc.

Non-spliced alignment software

- Bowtie: ultrafast, easy to run, good mapping results
  - bowtie 1 vs bowtie 2 (from bowtie manual)
    - For reads >50 bp Bowtie 2 is generally faster, more sensitive, and uses less memory than Bowtie 1.
    - bowtie 2 supports gapped alignment. Bowtie 1 only finds ungapped alignments.
    - Bowtie 2 supports a "local" alignment mode, in addition to the “end-to-end" alignment mode supported by bowtie1
- BWA
  - refer to the BaRC SOP for detailed information

mapping reads to genome with bowtie2

single end reads

$ **bsub bowtie2 –phred64 –x /nfs/genomes/mouse_mm10_dec_11_no_random/bowtie/mm10 DNA.fastq –S DNA.sam**

paired-end reads

$ bsub bowtie2 –phred64 –x /nfs/genomes/mouse_mm10_dec_11_no_random/bowtie/mm10 **-1 Reads1.fastq -2 Reads2.fastq** –S DNA.sam

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

- **-S** name of SAM output file
- **-x** bt2 index

check your email to see percentage of reads mapped.

By default, bowtie reports one alignment if a read mapped to multiple genomic regions.

Aligned file format

- Bam vs Sam:
  1. Bam: binary format
  2. Bam is much smaller than sam.

- Convert .sam to .bam format, etc.
- $ **bsub /nfs/BaRC_Public/BaRC_code/Perl/SAM_to_BAM_sort_index/SAM_to_BAM_sort_index.pl DNA.sam**
  1. Convert .sam to .bam
  2. Sort bam file
  3. Index bam file, created a .bai file
- Delete the .sam file

Spliced alignment with tophat
tophat2 used bowtie2

signal end reads
$ **bsub tophat --solexa1.3-quals --segment-length 15 -G /nfs/genomes/mouse_mm10_dec_11_no_random/gtf/mm10_no_random.refseq.gtf /nfs/genomes/mouse_mm10_dec_11_no_random/bowtie/mm10 sample_good_trimmed.fastq**
# paired-end reads
Add additional fastq file to the end of above command.

Input qualities	Illumina version
--solexa-quals	<= 1.2
--phred64 or --solexa1.3-quals	1.3-1.7
--phred33	>= 1.8

-o/--output-dir	default = tophat_out
--segment-length	Shortest length of a spliced read that can map to one side of the junction. default:25
-N	max. number of mismatches in a read
-G <GTF file>	Map reads to virtual transcriptome (from gtf file) first.

Systematic evaluation of spliced alignment programs for RNA-seq data
Engstrom et.al Nature Methods 10, 1185–1191 (2013)

Gene model files

- Gene model:
  - Genomic location of transcripts: exons, UTRs, CDS
- Refseq vs Ensembl:
  - The number of genes in Refseq is much smaller than Ensembl: mm9: 24k vs 38k
    - Refseq: known genes from NCBI
    - Ensembl: multiple resources. Automatic + manual curation
  - Ensembl also includes gene categories:
    - protein_coding, lincRNA, miRNA, rRNA, etc.

What to look for when few reads mapped?

- Reads are not perfectly paired. *
  - Usually occurs after QC'ing step. Removing low quality reads or adapters creates uneven distribution of reads

  $ bsub "/nfs/BaRC_Public/BaRC_code/Perl/cmpfastq/cmpfastq.pl s_8_1_filtered.fastq s_8_2_filtered.fastq"
- Too many reads mapped to ribosome?
  - Count reads with Ensembl rRNAs gene models
  - Blast top overrepresented sequences in fastQC output
- Mapping parameters are too stringent. *
  - Increase number of mismatches
  - Adjust the insert size of paired-end reads?

* Refer to *BaRC SOP* for more information

Optimize mapping across introns

- Tophat default parameters are designed for mammalian RNA-seq data.
- Reduce “maximum intron length” for non-mammalian organisms

  -l: default is 500,000

Species	Max_intron_length
yeast	2,484
arabidopsis	11,603
c.elegans	100,913
fly	141,628

1. Check quality and clean up reads
2. Map Reads:
   - bowtie
   - tophat
3. Check mapped reads:
   - Look at the reads in genome browser
   - Get mapping statistics

Index the .bam file in tophat output folder

1. Bam index file (.bai) is needed for visualization
2. Go to the directory with mapped RNA-seq results:

   $ cd tophat_out

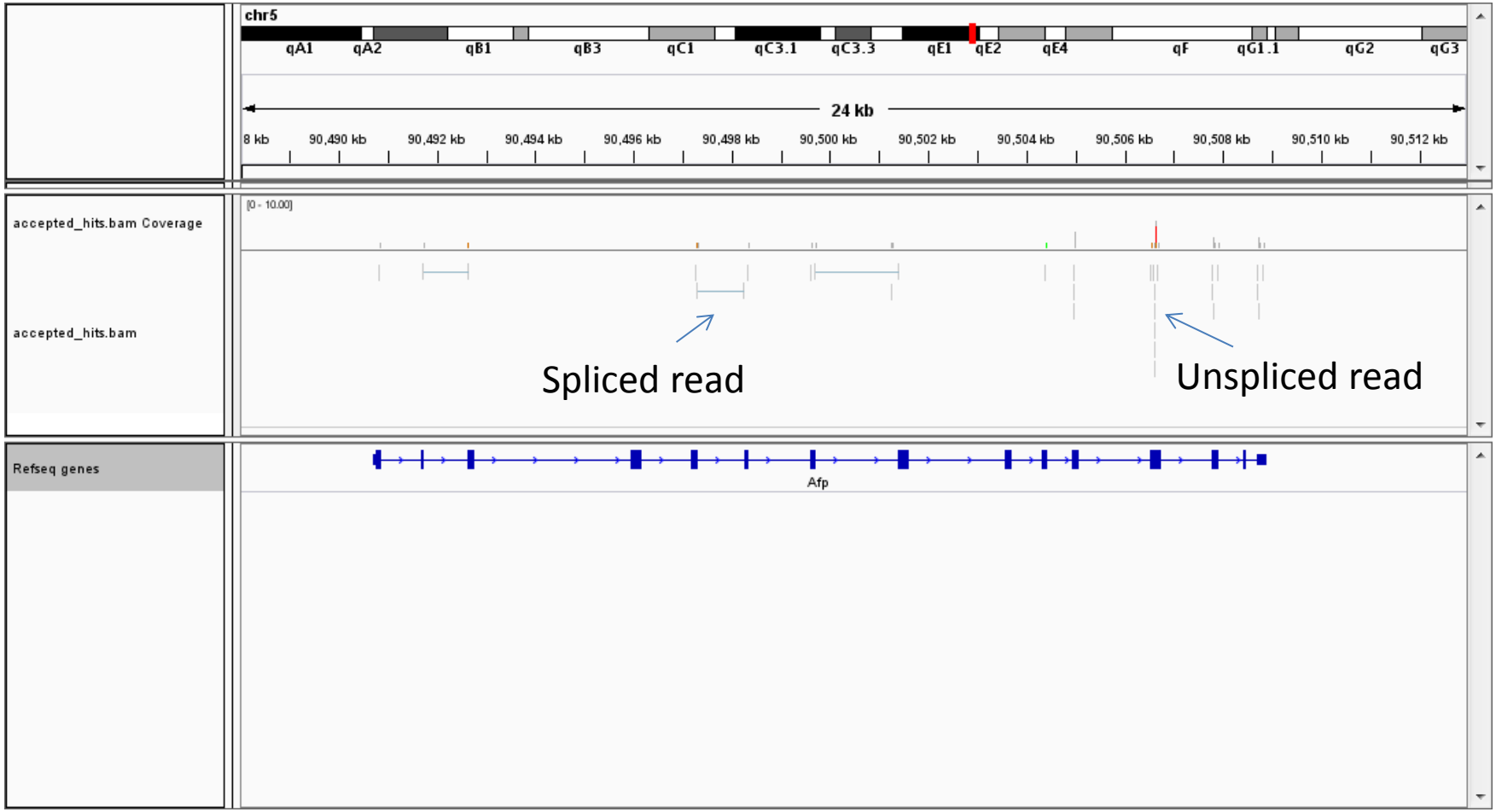
3. Index bam file

   $ samtools index accepted_hits.bam

Note: tophat puts aligned reads in accepteed_hits.bam, and not-aligned reads in unmapped.bam

IGV view of the RNA-seq data:

sample_good_trimmed.fastq



Note: We used only 0.05% of original reads for mapping.

How many reads mapped?

$ bam_stat.py -i accepted_hits.bam

Total Records	10640
QC failed	0
Optical/PCR duplicate	0
Non Primary Hits	4730
Unmapped reads	0
Multiple mapped reads	**1652**
Uniquely mapped	**4258**
Read-1	0
Read-2	0
Reads map to '+'	1659
Reads map to '-'	2599
Non-splice reads	4038
Splice reads	220
Reads mapped in proper pairs	0

The total number of reads used for mapping: can be found in the fastQC output

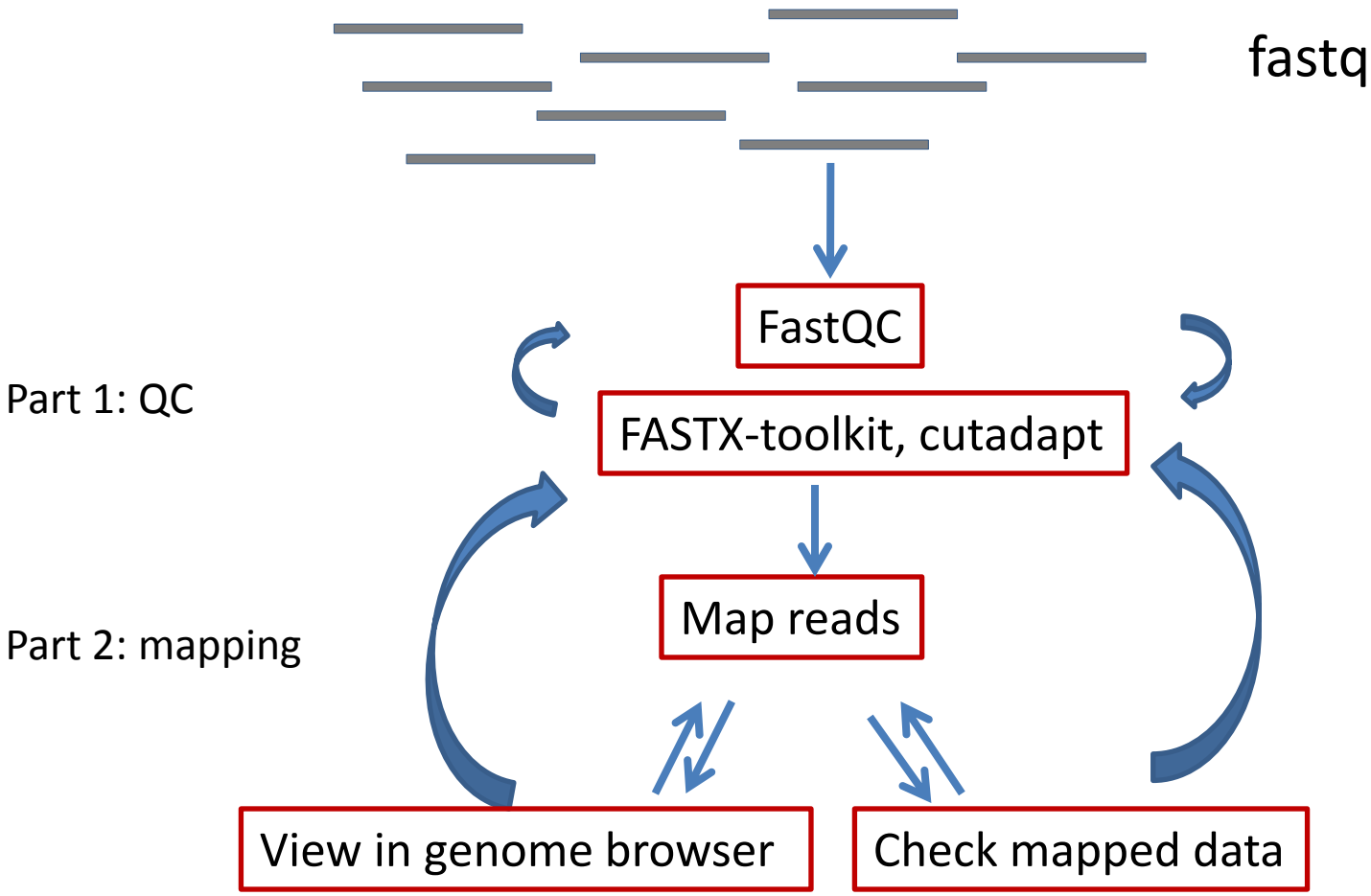
Percent of reads mapped:
(1652+4258)/6629

89%

Note: If a read mapped to multiple regions, tophat reports up to 20 best alignments (records) by default.

RSeQC package from Baylor College of Medicine

Summary


fastq
FastQC
Part 1: QC
FASTX-toolkit, cutadapt
Map reads
Part 2: mapping
View in genome browser
Check mapped data

Summary

- Quality control
  - fastQC: fastqc
- Clean up reads:
  - fastx tool kits: fastq_quality_filter fastx_trimmer
  - Cutadapt: cutadapt
- Map reads:
  - Bowtie: bowtie2
  - Tophat: tophat
- Understand the mapped files, and check mapping quality:
  - Samtools: samtool view, samtool index
  - RSeQC: bam_stat.py

BaRC Standard operating procedures

barcwiki.wi.mit.edu/wiki/SOPs

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

BaRC
Bioinformatics and Research Computing

Search

Login | Preferences | Help/Guide | About Trac

Wiki | Timeline | Search

wiki: SOPs

Start Page | Index | History

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 next

- RNA-seq
- Chip-seq
- Annotation of genomic regions
- Visualizing NGS Data