Short read sequence pipelines

- Genomic resequencing
- SNP / Polymorphism identification

Introducing more tools

- There are more tools that you can remember…
- Focusing on some basics

Pipelines

- Multiple steps to achieve analyses
- Often steps that have dependancies
- Want to restart and not have to re-run everything
- Use cluster queueing system (sbatch)

NGS sequence data

- Quality control
- Alignment
- Variant calling
  - SNPs
  - Indels

Sequence data sources

- Sanger
  - Long reads, high quality, expensive
- Illumina
  - Short reads 50-150bp (HiSeq) and up to 250bp (MiSeq)
  - Cheap and Dense read total (HiSeq 200-300M paired-reads for ~$2k)
- PacBio
  - Long reads, but smaller amount (10k)
  - Low seq quality and not cheap. But getting better
  - Can help improve assemblies, probably not sufficient for an assembly alone (too expensive to get deep enough coverage)
- Nanopore
  - Long reads. Fast running time
  - Quality improving
  - Lower density than Illumina
  - relatively cheap

File formats

FASTQ

```
@SRR527545.1 1 length=76
GTCGATGATGCCTGCTAAACTGCAGCTTGACGTACTGCGGACCCTGCAGTCCAGCGCTCGTCATGGAACGCAAACG
+
HHHHHHHHHHHHFGHHHHHHFHHGHHHGHGHEEHHHHHEFFHHHFHHHHBHHHEHFHAH?CEDCBFEFFFFAFDF9
```

FASTA format

```
>SRR527545.1 1 length=76
GTCGATGATGCCTGCTAAACTGCAGCTTGACGTACTGCGGACCCTGCAGTCCAGCGCTCGTCATGGAACGCAAACG
```

SFF - Standard Flowgram Format - binary format for 454 reads

Read naming

ID is usually the machine ID followed by flowcell number column, row, cell of the read.

Paired-End naming can exist because data are in two file, first read in file 1 is paired with first read in file 2, etc. This is how data come from the sequence base calling pipeline. The trailing /1 and /2 indicate they are the read-pair 1 or 2.

In this case #CTTGTA indicates the barcode sequence since this was part of a multiplexed run.

File: Project1_lane6_1_sequence.txt

```
@HWI-ST397_0000:2:1:2248:2126#CTTGTA/1
TTGGATCTGAAAGATGAATGTGAGAGACACAATCCAAGTCATCTCTCATG
+HWI-ST397_0000:2:1:2248:2126#CTTGTA/1
eeee\dZddadddddeeeeeeedaed_ec_ab_\NSRNRcdddc[_c^d
```

File: Project1_lane6_2_sequence.txt

```
@HWI-ST397_0000:2:1:2248:2126#CTTGTA/2
CTGGCATTTTCACCCAAATTGCTTTTAACCCTTGGGATCGTGATTCACAA
+HWI-ST397_0000:2:1:2248:2126#CTTGTA/2
]YYY_\[[][da_da_aa_a_a_b_Y]Z]ZS[]L[\ddccbdYc\ecacX
```

Paired-end reads

These files can be interleaved, several simple tools exist, see velvet package for shuffleSequences scripts which can interleave them for you.

Interleaved was requried for some assemblers, but now many support keeping them separate. However the order of the reads must be the same for the pairing to work since many tools ignore the IDs (since this requires additional memory to track these) and instead assume in same order in both files.

Orientation of the reads depends on the library type. Whether they are

```
---->   <----   Paired End (Forward Reverse)
<----   ---->   Mate Pair  (Reverse Forward)
```

Data QC

- Trimming
- sickle, FASTX_toolkit, SeqPrep
- Additional considerations for Paired-end data
- Evaluating quality info with reports

FASTX toolkit

- Useful for trimming, converting and filtering FASTQ and FASTA data
- One gotcha - Illumina quality score changes from 64 to 33 offset
- Default offset is 64, so to read with offset 33 data you need to use -Q 33 option
- fastx_quality_trimmer
- fastx_splitter - to split out barcodes
- fastq_quality_formatter - reformat quality scores (from 33 to 64 or)
- fastq_to_fasta - to strip off quality and return a fasta file
- fastx_collapser - to collapse identical reads. Header includes count of number in the bin

FASTX - fastx_quality_trimmer

- Filter so that X% of the reads have quality of at least quality of N
- Trim reads by quality from the end so that low quality bases are removed (since that is where errors tend to be)
- Typically we use Phred of 20 as a cutoff and 70% of the read, but you may want other settings
- This is adaptive trimming as it starts from end and removes bases
- Can also require a minimum length read after the trimming is complete

Trimming paired data

- When trimming and filtering data that is paired, we want the data to remain paired.
- This means when removing one sequence from a paired-file, store the other in a separate file
- When finished will have new File_1 and File_2 (filtered & trimmed) and a separate file File_unpaired.
- Usually so much data, not a bad thing to have agressive filtering

Trimming adaptors

- A little more tricky, for smallRNA data will have an adaptor on 3' end (usually)
- To trim needs to be a matched against the adaptor library - some nuances to make this work for all cases.
- What if adaptor has low quality base? Indel? Must be able to tolerate mismatch
- Important to get right as the length of the smallRNAs will be calculated from these data
- Similar approach to matching for vector sequence so a library of adaptors and vector could be used to match against
- Sometimes will have adaptors in genomic NGS sequence if the library prep did not have a tight size distribution.

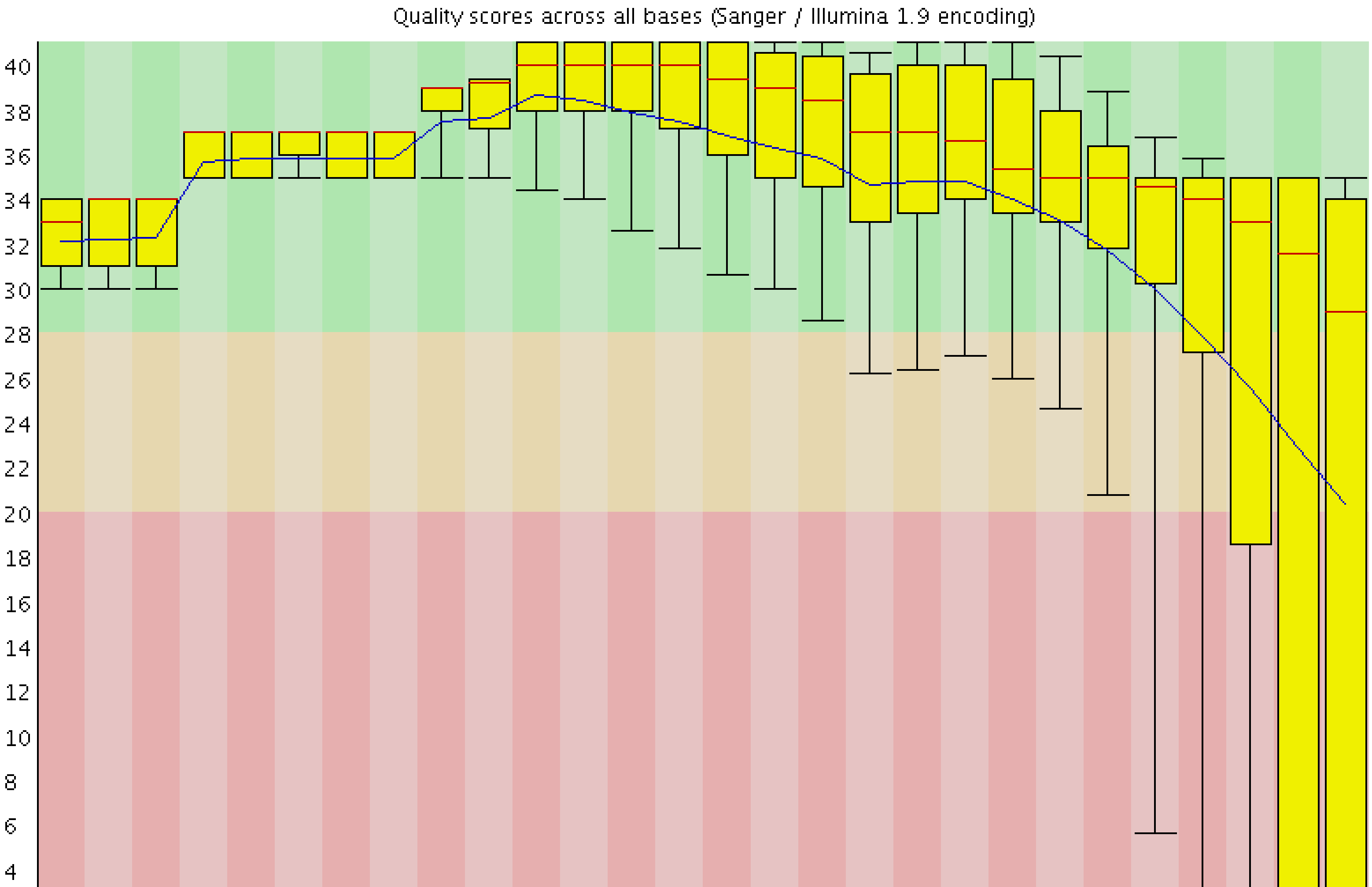
Trimming adaptors - tools

- cutadapt - Too to matching with alignment. Can search with multiple adaptors but is pipelining each one so will take 5X as long if you match for 5 adaptors.
- SeqPrep - Preserves paired-end data and also quality filtering along with adaptor matching

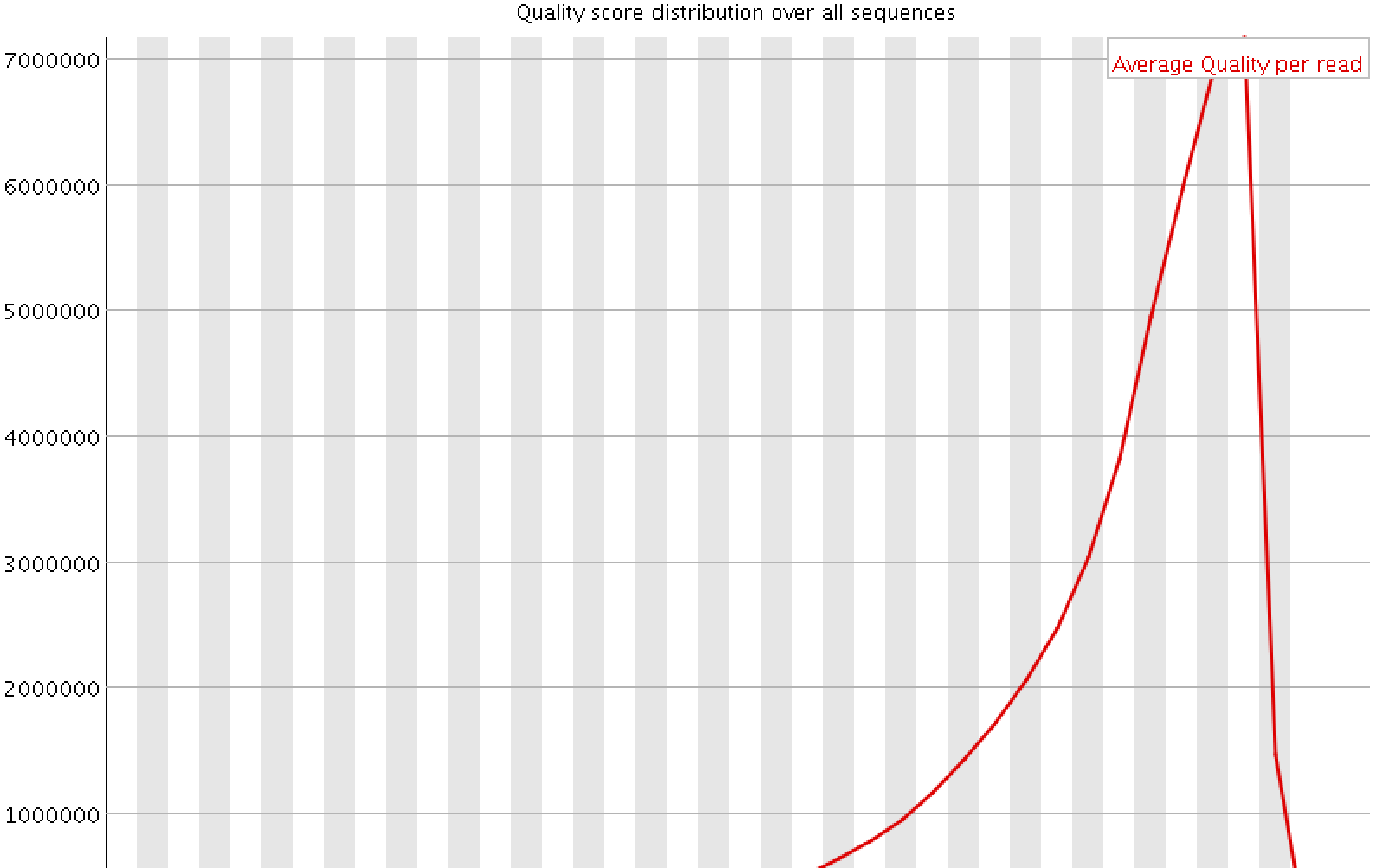
FASTQC for quality control

- Looking at distribution of quality scores across all sequences helpful to judge quality of run
- Overrepresented Kmers also helpful to examine for bias in sequence
- Overrepresented sequences can often identify untrimmed primers/adaptors

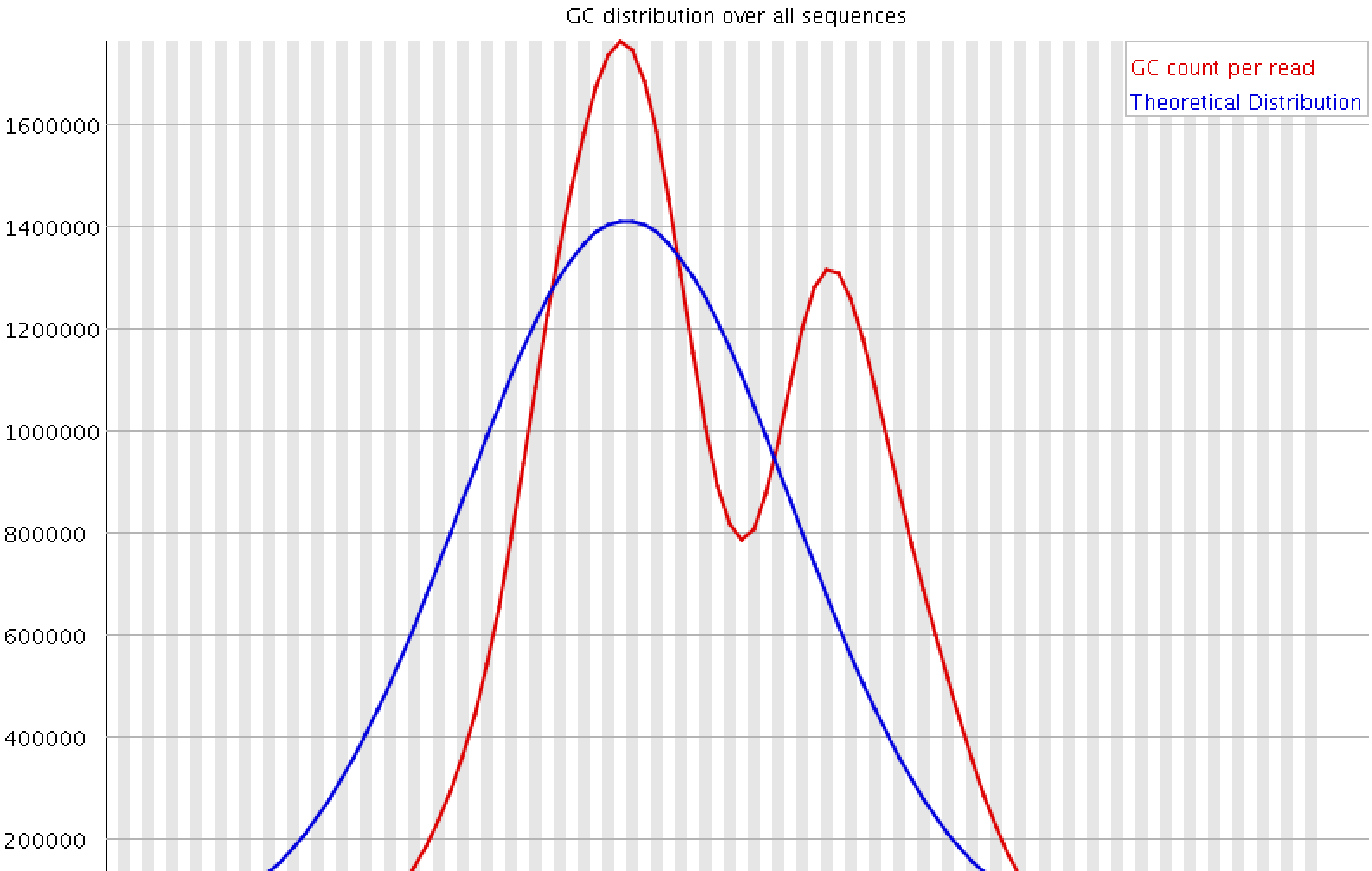
FASTQC - per base quality



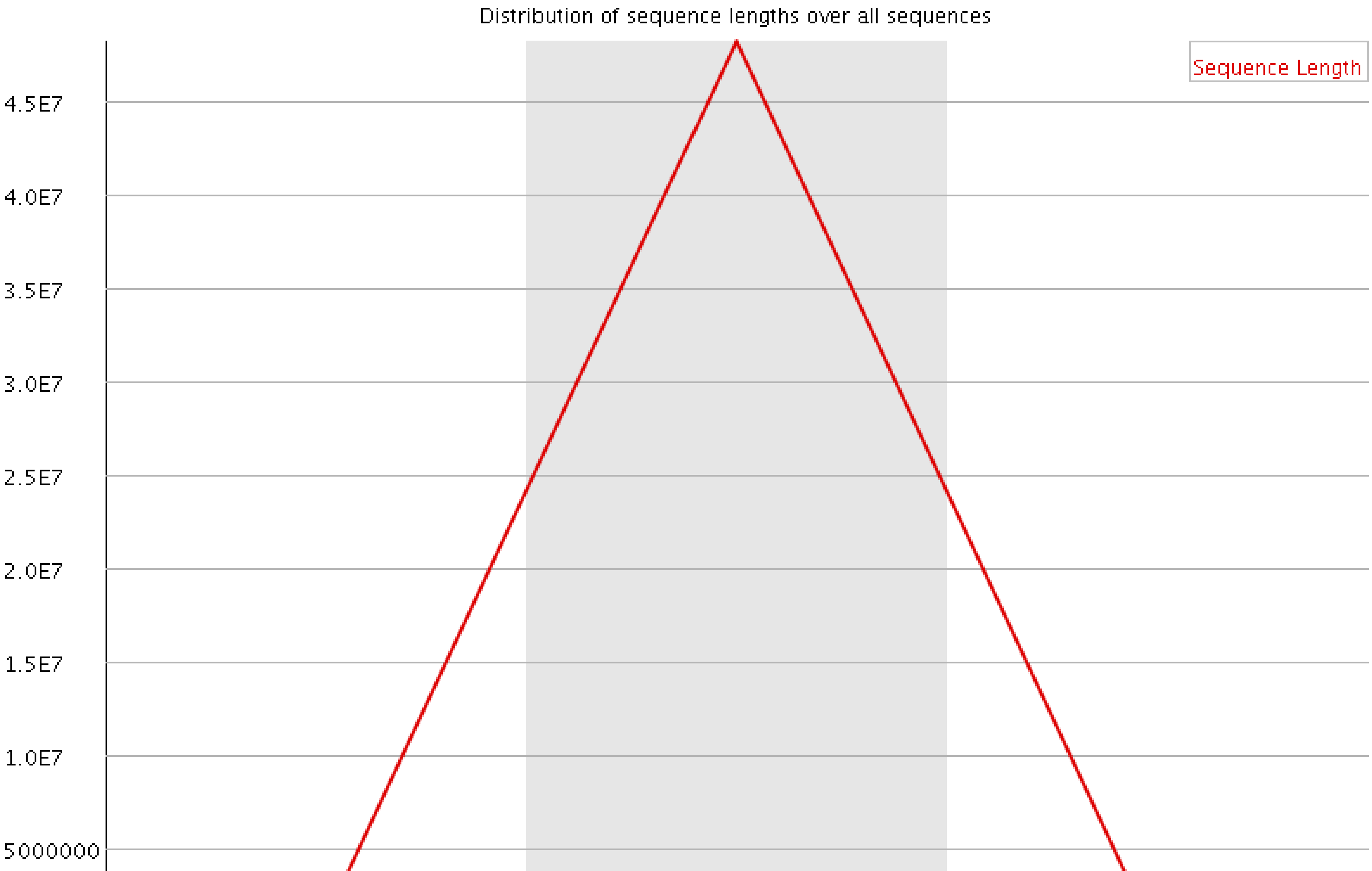
FASTQC - per seq quality



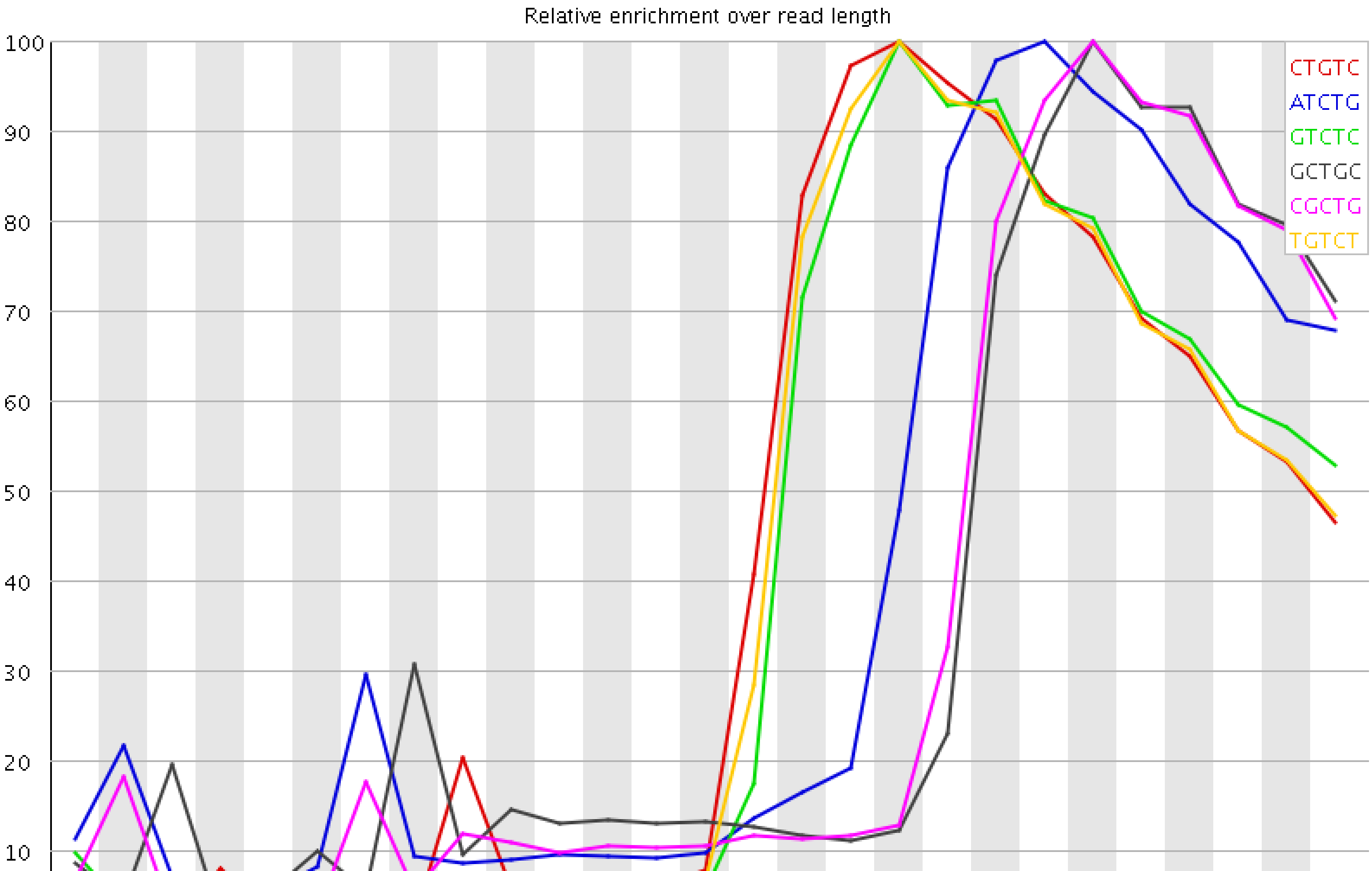
FASTQC - per seq GC content



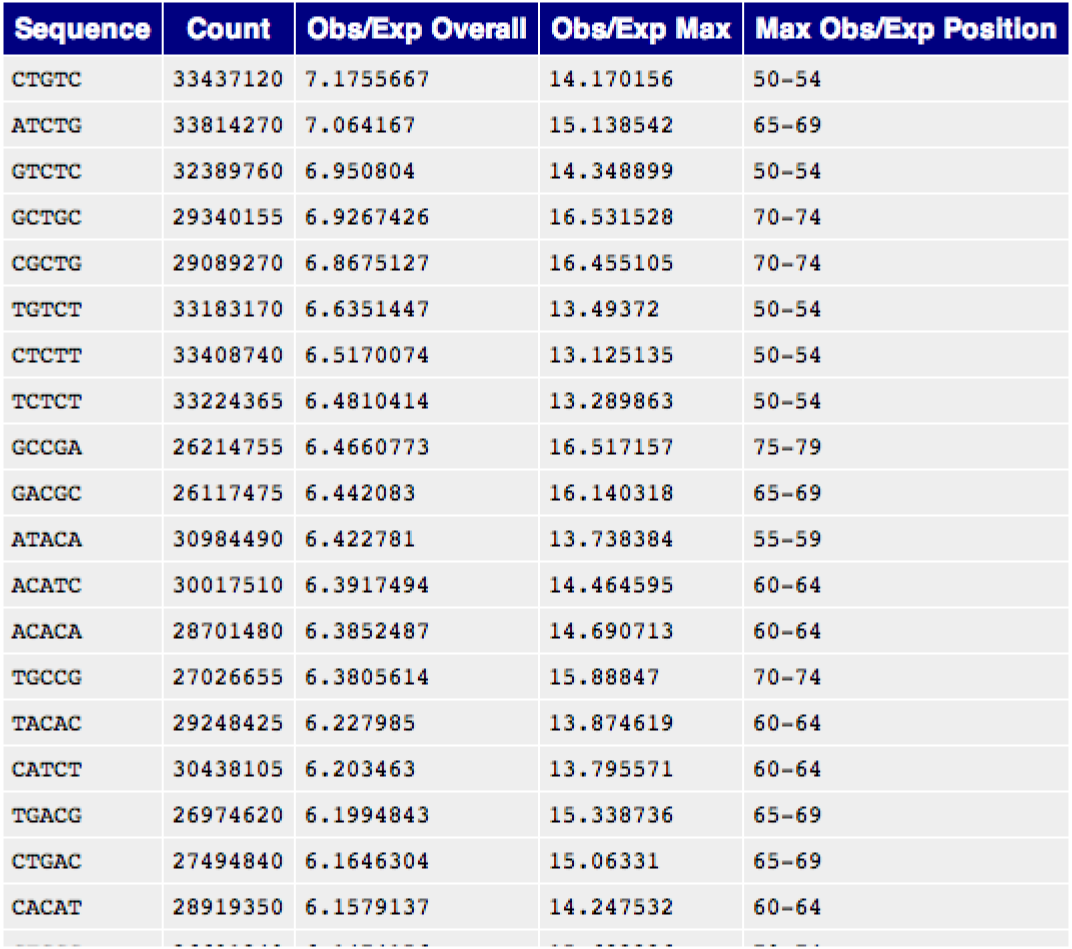
FASTQC - Sequence Length



FASTQC - kmer distribution



FASTQC - kmer table



Sequence	Count	Obs/Exp Overall	Obs/Exp Max	Max Obs/Exp Position
CTGTC	33437120	7.1755667	14.170156	50-54
ATCTG	33814270	7.064167	15.138542	65-69
GTCTC	32389760	6.950804	14.348899	50-54
GCTGC	29340155	6.9267426	16.531528	70-74
CGCTG	29089270	6.8675127	16.455105	70-74
TGTCT	33183170	6.6351447	13.49372	50-54
CTCTT	33408740	6.5170074	13.125135	50-54
TCTCT	33224365	6.4810414	13.289863	50-54
GCCGA	26214755	6.4660773	16.517157	75-79
GACGC	26117475	6.442083	16.140318	65-69
ATACA	30984490	6.422781	13.738384	55-59
ACATC	30017510	6.3917494	14.464595	60-64
ACACA	28701480	6.3852487	14.690713	60-64
TGCCG	27026655	6.3805614	15.88847	70-74
TACAC	29248425	6.227985	13.874619	60-64
CATCT	30438105	6.203463	13.795571	60-64
TGACG	26974620	6.1994843	15.338736	65-69
CTGAC	27494840	6.1646304	15.06331	65-69
CACAT	28919350	6.1579137	14.247532	60-64

Short read aligners

Strategy requires faster searching than BLAST or FASTA approach. Some approaches have been developed to make this fast enough for Millions of sequences.

- maq - one of the first aligners
- Burrows-Wheeler Transform is a speed up that is accomplished through a transformation of the data. Require indexing of the search database (typically the genome). BWA, Bowtie
- ? LASTZ
- ? BFAST

Workflow for variant detection

- Trim
- Check quality
- Re-trim if needed
- Align
- Possible realign around variants
- Call variants - SNPs or Indels
- Possibly calibrate or optimize with gold standard (possible in some species like Human)

Working a test example

What's in there?

- Downloaded Yeast genome from https://www.yeastgenome.org/
- Seqeuence Read Archive - https://www.ncbi.nlm.nih.gov/sra/
- One project https://www.ncbi.nlm.nih.gov/sra/SRX2627316[accn]
- BioProject https://www.ncbi.nlm.nih.gov/bioproject/PRJNA359887
- How to get Sequence files?

Fastq

FastA like format. Also encoded quality scores for bases

https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR5328284

```
@SRR5328284.1 1/1
TAGTANGCCTATGATGGCTGAGACACCTTGACGATGGAATCGCATTCTCTATCTAAACAGTGGTCAA
TGGATTCTACGAGAGCTAAGACCATAGATCTGTCTCTTATACACATCTCCGAGCCCACGAGACGCTA
CGCTATCTCGTATGCC
+
AAAAA#AE//AEAEE66EEA/EEEEEE/EEEEEAEEEEAEEEAE/EEAA/EEEEEEAEEEE/AEEEEE
/<A/EEE/AE////EEEE/EAE/EE/6E<E/EEEAEEEAEEAEA/EEA<E/E/E<A
A/AEAEEA<E/<EAA/AEE<<EEEAE
```

Let's walk through a few steps

First get an interactive session on the cluster

```
$ srun --pty bash -l
# go into a folder for testing, eg bigdata
$ cd bigdata
$ git clone https://github.com/biodataprog/datasets.git
$ cd datasets/SNP
$ ls -l fastq
$ ls -l genome
```

Using tools for short read alignment: BWA

BWA is one tools for short read alignment

See the BWA manual page: http://bio-bwa.sourceforge.net/bwa.shtml

Step 1: Index the Genome

```
$ module load bwa
$ bwa index genome/yeast_genome.fasta
```

Align reads

Align short reads into an alignment file

```
$ bwa mem genome/yeast_genome.fasta \
fastq/SRR5328284_1.trunc100k.fastq.gz \
fastq/SRR5328284_2.trunc100k.fastq.gz > yeast.sam
```

SAM File

```
$ more SRR5328284.100k_reads.sam
@SQ    SN:NC_001133    LN:230218
@SQ    SN:NC_001134    LN:813184
@SQ    SN:NC_001135    LN:316620
@SQ    SN:NC_001136    LN:1531933
@SQ    SN:NC_001137    LN:576874
@SQ    SN:NC_001138    LN:270161
@SQ    SN:NC_001139    LN:1090940
@SQ    SN:NC_001140    LN:562643
@SQ    SN:NC_001141    LN:439888
@SQ    SN:NC_001142    LN:745751
@SQ    SN:NC_001143    LN:666816
@SQ    SN:NC_001144    LN:1078177
@SQ    SN:NC_001145    LN:924431
@SQ    SN:NC_001146    LN:784333
@SQ    SN:NC_001147    LN:1091291
@SQ    SN:NC_001148    LN:948066
@SQ    SN:NC_001224    LN:85779
@PG    ID:bwa    PN:bwa    VN:0.7.12-r1039    CL:bwa mem genome/yeast_genome.fasta f
SRR5328284.1    99    NC_001148    16824    60    96M54S    =    16822    96    TAGT
GATGGAATCGCATTCTCTATCTAAACAGTGGTCAATGGATTCTACGAGAGCTAAGACCATAGATCTGTCTCTTATACACATCTC
AAAAA#AE//AEAEE66EEA/EEEEEE/EEEEEAEEEEAEEEAE/EEAA/EEEEEEAEEEE/AEEEEE/<A/EEE/AE////EE
```

Samtools and BCFTools

- samtools http://samtools.sourceforge.net/
- http://www.htslib.org/workflow/
- bcftools http://www.htslib.org/doc/bcftools.html

See http://www.htslib.org/workflow/

```
$ module load samtools
$ samtools sort -O bam -l 0 -o yeast.bam yeast.sam
$ samtools view -T yeast.fasta -C -o yeast.cram yeast.bam
$ module load bcftools
$ samtools mpileup -uf genome/yeast_genome.fasta
SRR5328284.100k_reads.bam | bcftools view -bvcg - > var.raw.bcf

$ bcftools stats -F <ref.fa> -s - <study.vcf.gz> > <study.vcf.gz.stats>
$ mkdir plots
$ plot-vcfstats -p plots/ <study.vcf.gz.stats>
```

VCF file

Variant call format

http://www.internationalgenome.org/wiki/Analysis/vcf4.0/