Introduction to NGS Data Analysis on the Ada Cluster

Your Login Password

- Both state of Texas law and TAMU regulations prohibit the sharing and/or illegal use of computer passwords and accounts
- Don’t write down passwords
- Don’t choose easy to guess/crack passwords
- Change passwords frequently

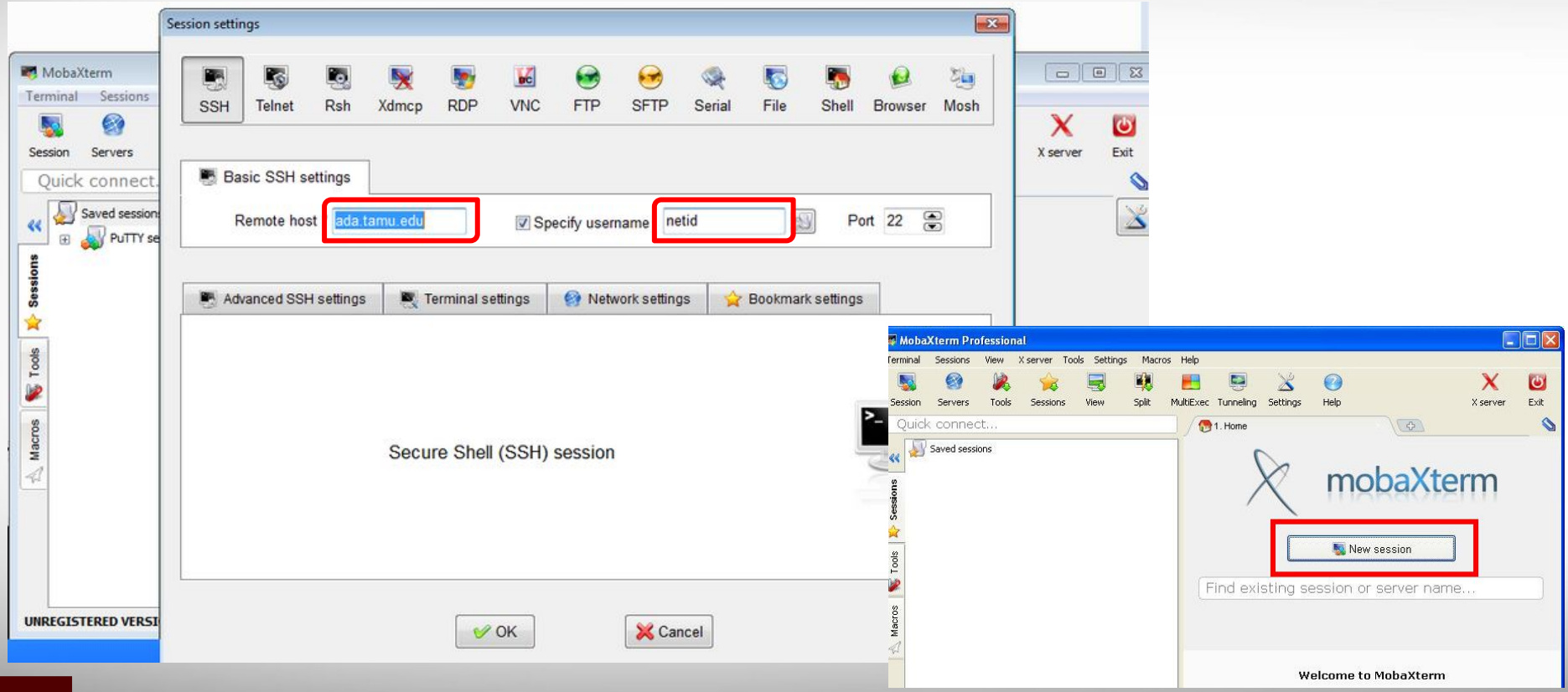
For More Help...

Website:	hprc.tamu.edu
Email:	help@hprc.tamu.edu
Telephone:	(979) 845-0219
Visit us in person:	Henderson Hall, Room 114A

Help us, help you -- we need more info

- Which Cluster
- UserID/NetID
- Job id(s) if any
- Location of your jobfile, input/output files
- Application used if any
- Module(s) loaded if any
- Error messages
- Steps you have taken, so we can reproduce the problem

Using SSH - MobaXterm (on Windows)



Next Generation Sequencing (NGS)

Illumina Sequencing Technology

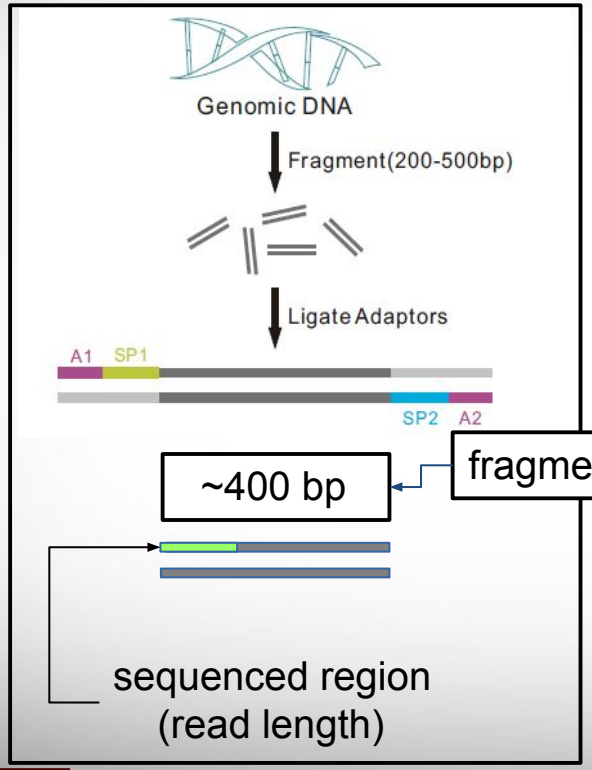
	 MiniSeq System	 MiSeq Series	 NextSeq Series	 HiSeq Series	 HiSeq X Series*	 NovaSeq 5000
Key Methods	Amplicon, targeted RNA, small RNA, and targeted gene panel sequencing.	Small genome, amplicon, and targeted gene panel sequencing.	Everyday exome, transcriptome, and targeted resequencing.	Production-scale genome, exome, transcriptome sequencing, and more.	Population- and production-scale whole-genome sequencing.	Same as HiSeq
Maximum Output	7.5 Gb	15 Gb	120 Gb	1500 Gb	1800 Gb	2000 Gb
Maximum Reads per Run	25 million	25 million†	400 million	5 billion	6 billion	6.6 billion
Maximum Read Length	2 × 150 bp	2 × 300 bp	2 × 150 bp	2 × 150 bp	2 × 150 bp	2 x 150 bp
Run Time	4–24 hours	4–55 hours	12–30 hours	<1–3.5 days (HiSeq 3000/HiSeq 4000) 7 hours–6 days (HiSeq 2500)	<3 days	19 - 40 hrs
Benchtop Sequencer	Yes	Yes	Yes	No	No	no

http://www.illumina.com/systems/sequencing-platforms.html (Oct 2017)

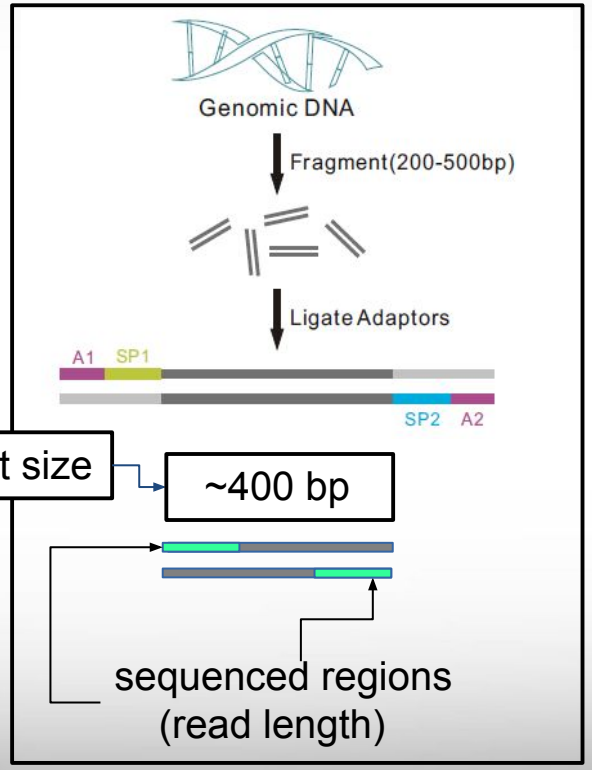
Illumina Sequencing Libraries

illumina.com

single end

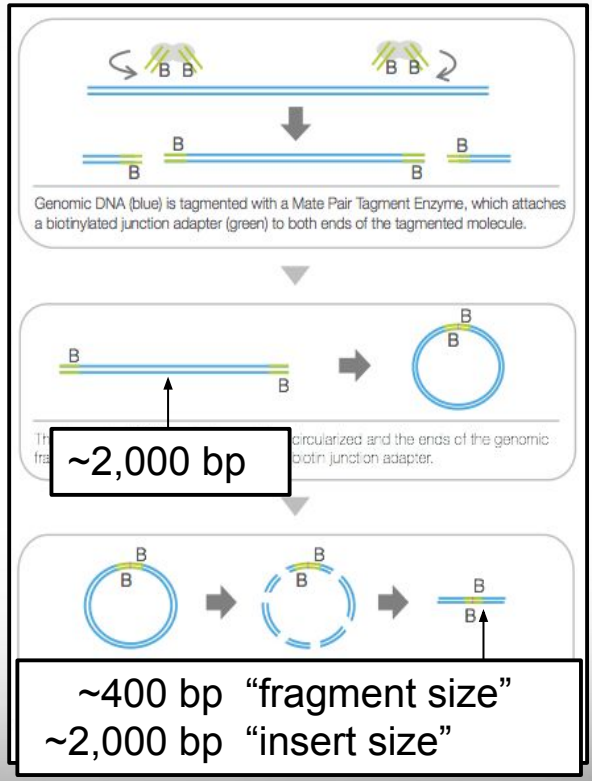


paired ends

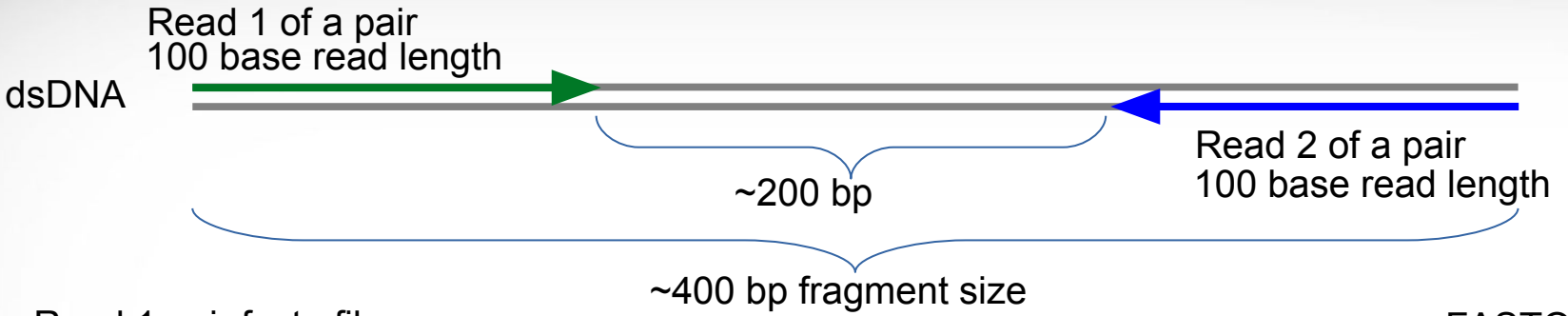


fragment size

mate pairs



Paired End Reads



Read 1 pair fastq file

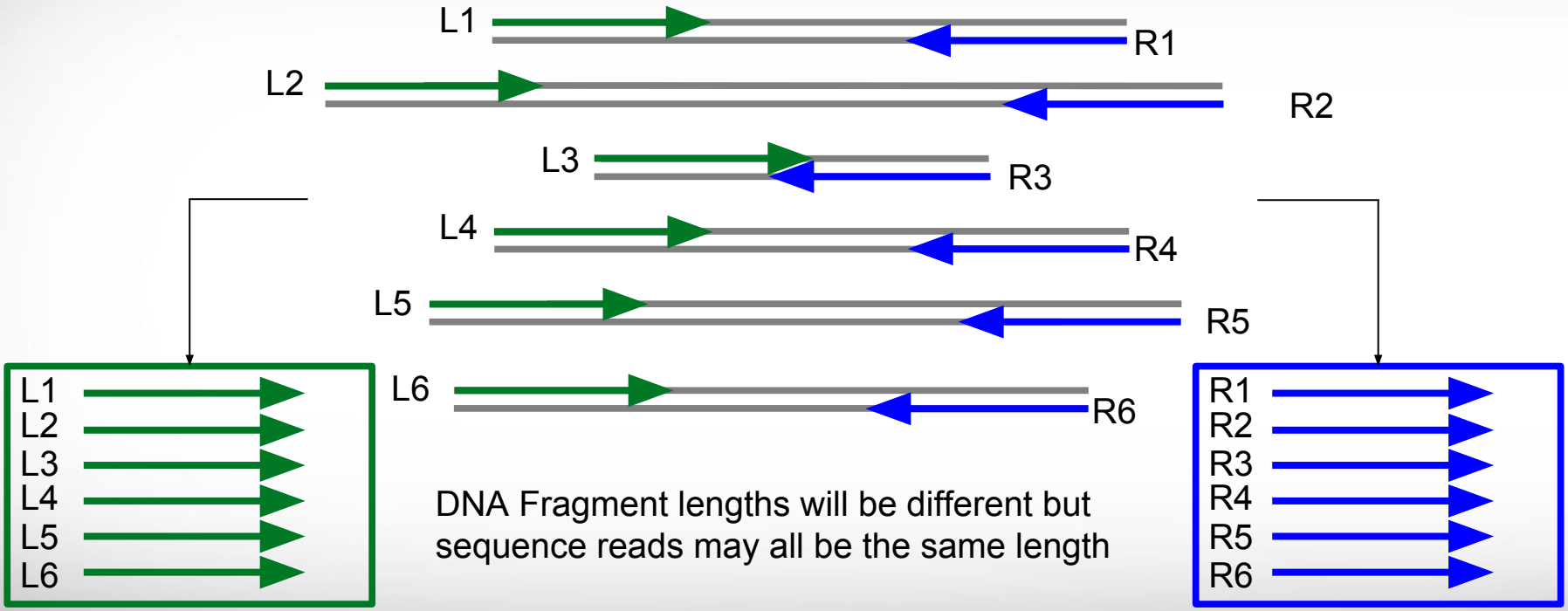
FASTQ format

```
@M00861:1:000000000-A36BE:1:1101:14650:1529 1:N:0:8
TTCTTAAAAATACCATAAAAGGCTTAAACTTGCCATTTACGACGGATTAATTCCAACTCTTTTCGGCTATCTTCATCTTTTAAGGTTAAATGACTCATAACGG
+
FFFHBFFHHIIIIIIHFHHCGEFGHHIHHHIHD/?DGGHHH@DEB,5EGHGHHIIHIF?FGGHHCCBFDGHFHDGHGFFFGDFHH?DFHDFFHHFHFHFFHHH
```

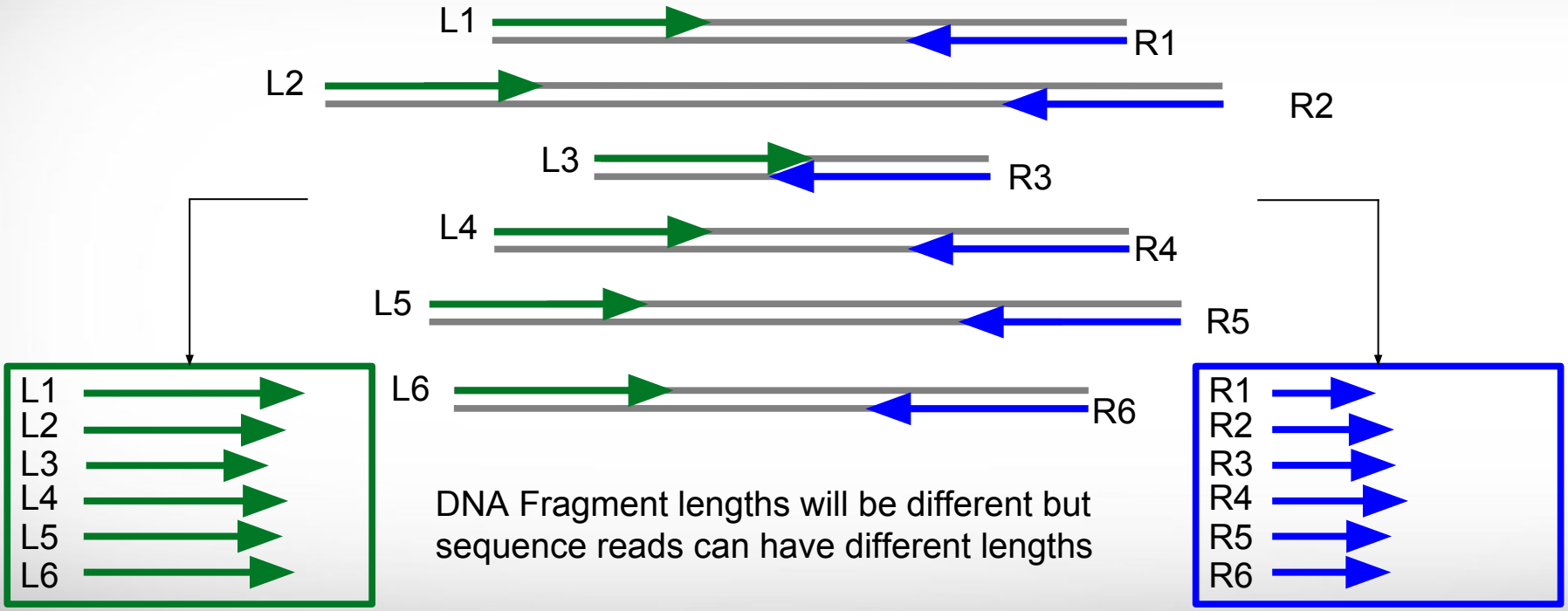
Read 2 pair fastq file

```
@M00861:1:000000000-A36BE:1:1101:14650:1529 2:N:0:8
ACTAAAAATCAATTTTATCAATTTCAAGCTCTACCTTATTTACTCATTATTTTAGTGATGGCCACTTTAATAAAAATATTGGTAGCATATTTTGCAATAGCGG
+
BFFHIHHHFHHDGHIHHIHHHGHHHHHFHHDFFHIHIIIHIHDFHHHIHIIIH=AAFHIIIHFGFHHHHHGGHHIHHFGFFFEGGHHHDGHHH/CGHIFFHHH
```

Maintain Read Pair Order

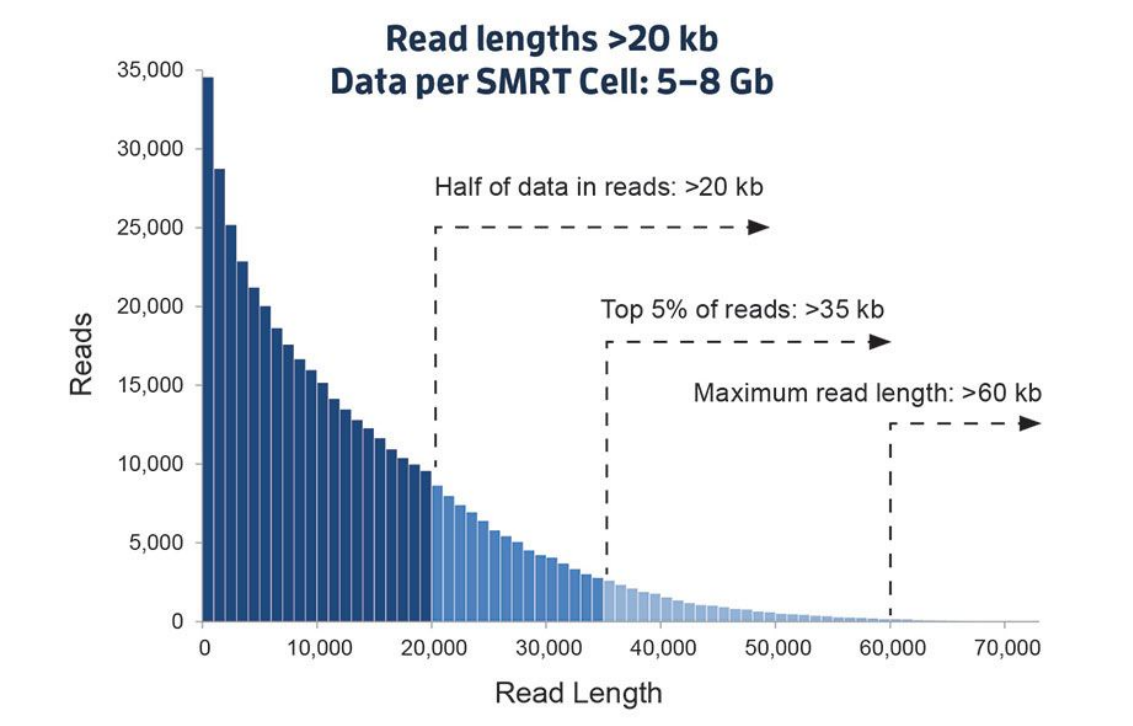


MiSeq Can Perform Initial QC Trimming



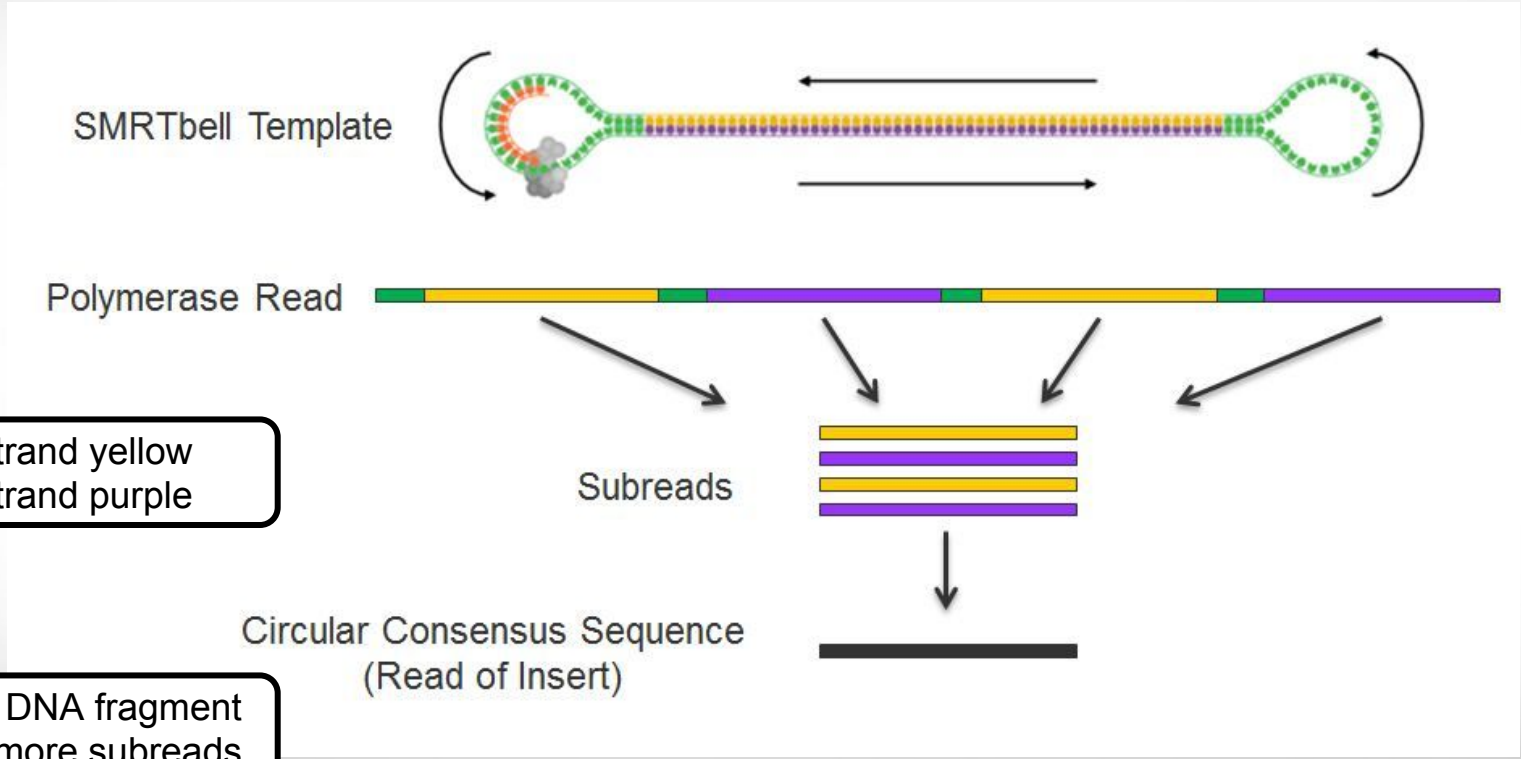
PacBio Long Read Sequencing

Sequel Sequencer



pacb.com

PacBio Long Read Sequencing



\+ Strand yellow
\- Strand purple

Shorter DNA fragment equals more subreads

pacb.com

PacBio Sequencing Tools on Ada

- PBSuite
  - PBJelly aligns PacBio reads to draft assemblies
- Proovread and LSC
  - Correct PacBio reads with Illumina reads
    - Computationally intensive
- Canu
  - PacBio long read assembler

https://hprc.tamu.edu/wiki/Ada:NGS:PacBio_tools

NGS Tools on Ada

Where to Find NGS Tools

- TAMU HPRC Documentation
  - https://hprc.tamu.edu/wiki/index.php/Ada:Bioinformatics
- Type the following UNIX **commands** to see which tools are already installed on Ada
  - `module avail`
  - `module spider toolname`
  - `module key assembly`

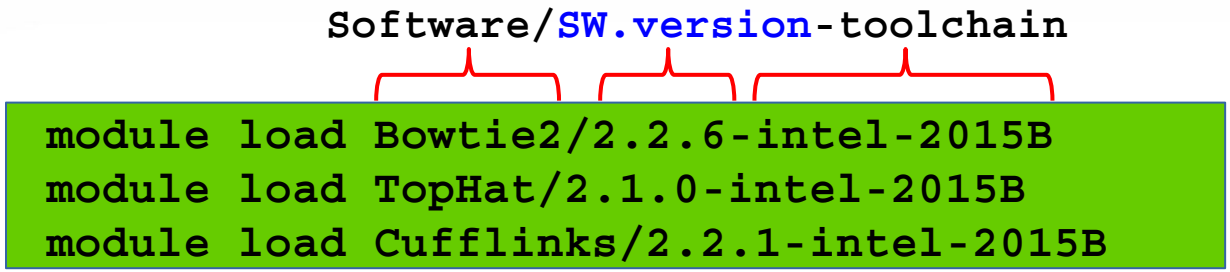
(not case sensitive, but read the entire output)

(some modules may be missed because this searches tool descriptions)

- If you find a tool that you want installed on Ada, send an email with the URL link to: **help@hprc.tamu.edu**
  - SeqAnswers http://seqanswers.com/wiki/Software/list
  - Omictools.com
  - slideshare.net – find shared NGS presentations

Ada Software Toolchains

- Use the same toolchains in your job scripts



- Avoid loading mixed toolchains:

```
module load Bowtie2/2.2.2-ictce-6.3.5
module load TopHat/2.0.14-goolf-1.7.20
module load Cufflinks/2.2.1-intel-2015B
```

- Avoid loading defaults which may have different toolchains

```
module load Bowtie2 TopHat Cufflinks
```

The GCCcore Toolchain

- To minimize the number of software builds, the GCCcore toolchain modules can be loaded alone or with any one of the following 2017A toolchains
  - `intel/2017A`
  - `iomkl/2017A`
  - `foss/2017A`
- Example of loading a GCCcore module with a 2017A module

```
module load Bowtie2/2.3.3.1-GCCcore-6.3.0
module load TopHat/2.1.1-intel-2017A-Python-2.7.12
```

Python-version-bare modules

- You need to load a non '-bare' Python version along with the -bare module
  - If you do not, then the older default OS Python version will be used
- Used in conjunction with GCCcore builds in order to reduce the number of software modules built.

`intel/2017A` `iomkl/2017A` `foss/2017A`

Three Examples of loading GCCcore Python -bare and a Python module with a 2017A toolchain

1.
```
module load Cython/0.25.2-GCCcore-6.3.0-Python-2.7.12-bare
module load Python/2.7.12-foss-2017A
```

2.
```
module load Cython/0.25.2-GCCcore-6.3.0-Python-2.7.12-bare
module load Python/2.7.12-iomkl-2017A
```

3.
```
module load Cython/0.25.2-GCCcore-6.3.0-Python-2.7.12-bare
module load HISAT2/2.1.0-intel-2017A-Python-2.7.12
```

Loads Python indirectly

Use $TMPDIR whenever possible

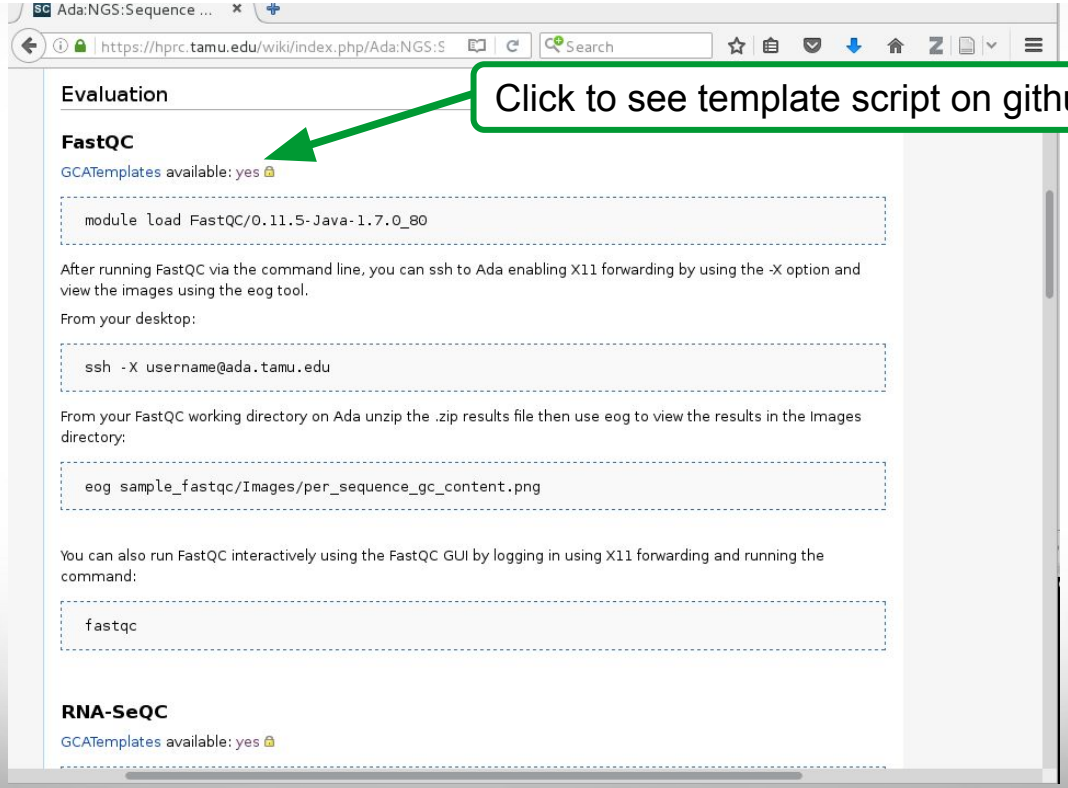
- Use the **$TMPDIR** if the application you are running can utilize a temporary directory for writing temporary files which are deleted when the job ends
- A temp directory (**$TMPDIR**) is automatically assigned for each job which uses the disk(s) on the compute node not the **$SCRATCH** shared file system
  - Especially useful when a computational tool writes tens of thousands of temporary files which are deleted when the job is finished and are not needed for the final results
  - This is useful since files on **$TMPDIR** will not count against your file quota
  - Don't use **$TMPDIR** if your software uses temporary files for restarting where it left off if it should stop before completion
  - Will significantly speed up an mpiBLAST job

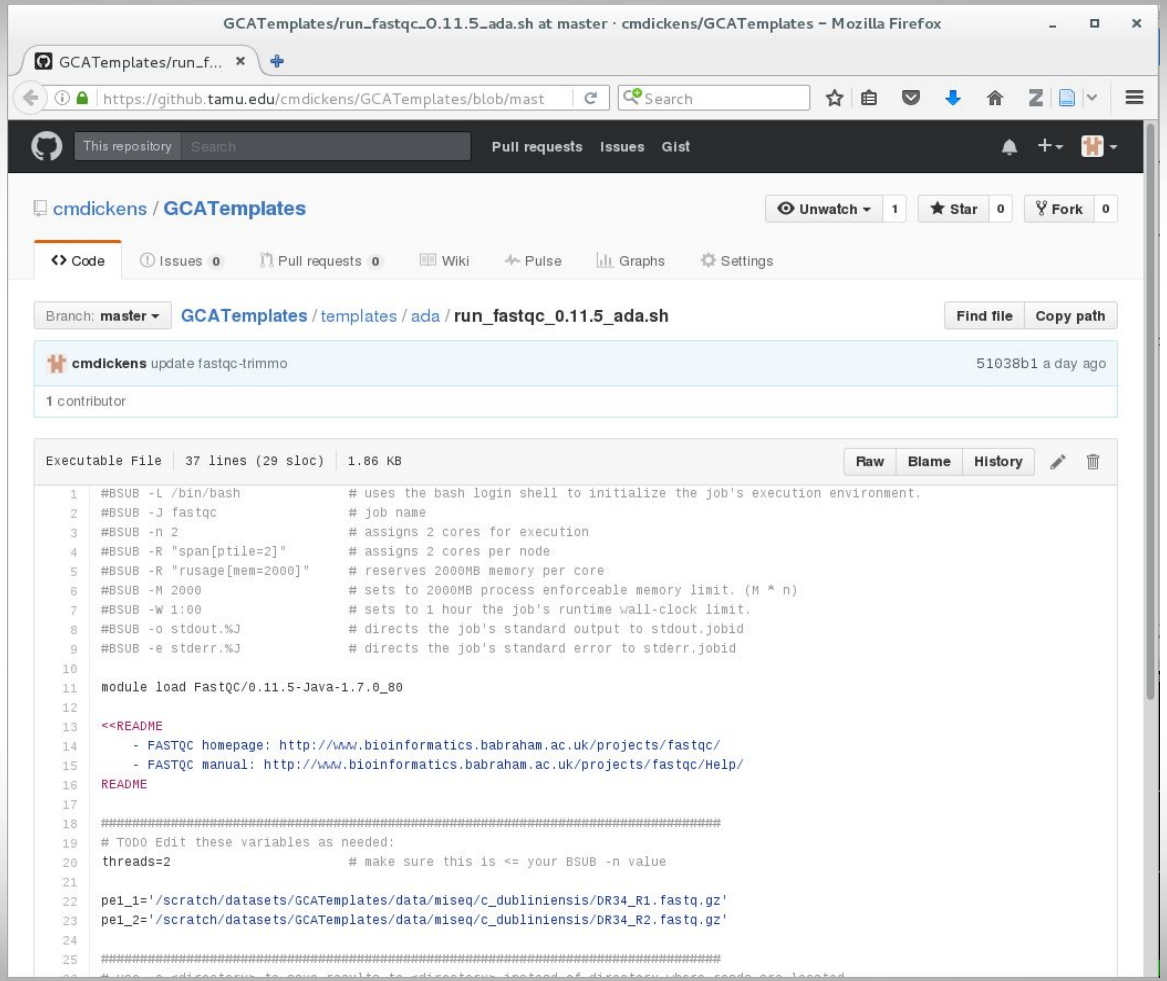
```
java -Xmx53g -jar $EBROOTPICARD/FastqToSam.jar TMP_DIR=$TMPDIR \
FASTQ=$pe1_1 FASTQ2=$pe1_2 OUTPUT=$outfile SAMPLE_NAME=$sample_name \
SORT_ORDER=$sort_order MAX_RECORDS_IN_RAM='null'
```

Template Job Scripts

Access GCATemplate Scripts for Ada

https://hprc.tamu.edu/wiki/index.php/Ada:NGS:Sequence_QC#FastQC




GCATemplates/run_fastqc_0.11.5_ada.sh at master · cmdickens/GCATemplates – Mozilla Firefox
GCATemplates/run_f...
https://github.tamu.edu/cmdickens/GCATemplates/blob/mast
Search
This repository
Search
Pull requests
Issues
Gist
cmdickens / GCATemplates
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings
Branch: master
GCATemplates / templates / ada / run_fastqc_0.11.5_ada.sh
Find file
Copy path
cmdickens update fastqc-trimmo
51038b1 a day ago
1 contributor
Executable File
37 lines (29 sloc)
1.86 KB
Raw
Blame
History
#BSUB -L /bin/bash # uses the bash login shell to initialize the job's execution environment.
#BSUB -J fastqc # job name
#BSUB -n 2 # assigns 2 cores for execution
#BSUB -R "span[ptile=2]" # assigns 2 cores per node
#BSUB -R "rusage[mem=2000]" # reserves 2000MB memory per core
#BSUB -M 2000 # sets to 2000MB process enforceable memory limit. (M * n)
#BSUB -W 1:00 # sets to 1 hour the job's runtime wall-clock limit.
#BSUB -o stdout.%J # directs the job's standard output to stdout.jobid
#BSUB -e stderr.%J # directs the job's standard error to stderr.jobid
module load FastQC/0.11.5-Java-1.7.0_80
<<README
- FASTQC homepage: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
- FASTQC manual: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/
README
# TODO Edit these variables as needed:
threads=2 # make sure this is <= your BSUB -n value
pe1_1='/scratch/datasets/GCATemplates/data/miseq/c_dubliniensis/DR34_R1.fastq.gz'
pe1_2='/scratch/datasets/GCATemplates/data/miseq/c_dubliniensis/DR34_R2.fastq.gz'

Finding NGS job template scripts using GCATemplates on Ada

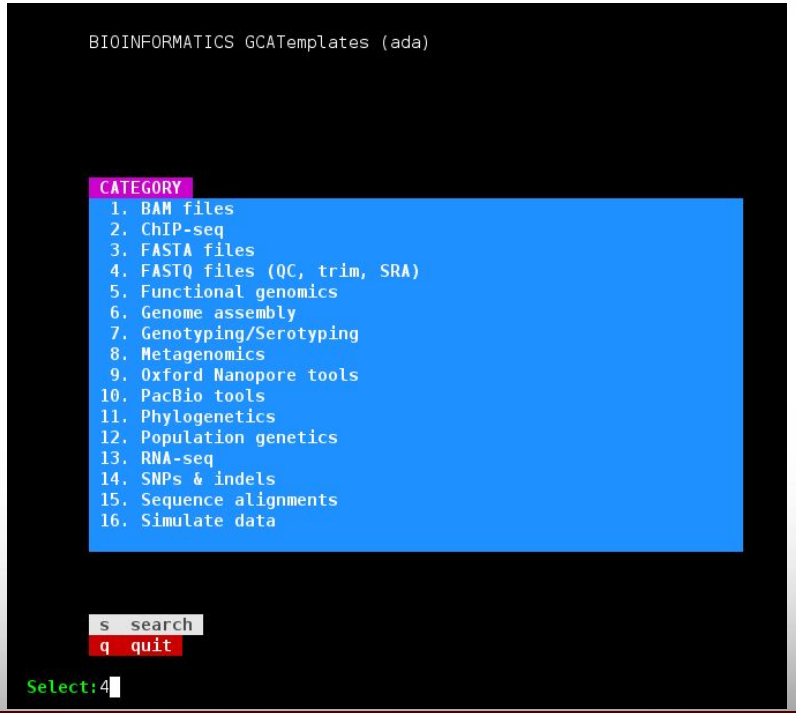
```
mkdir $SCRATCH/ngs_class
cd $SCRATCH/ngs_class
module load GCATemplates
gcatemplates
```

For practice, we will copy a template file

- Select #4 then find the template that contains fastqc
- Final step will save a template job script file to your current working directory
- After you save the template file:

```
module purge
```

Genomic Computational Analysis Templates



Sample GCATemplate Job Script (Ada)

```
#BSUB -L /bin/bash
#BSUB -J blastx
#BSUB -n 1
#BSUB -R "span[ptile=1]"
#BSUB -R "rusage[mem=2500]"
#BSUB -M 2500
#BSUB -W 2:00
#BSUB -o stdout.%J
#BSUB -e stderr.%J

module load BLAST+/2.2.31-intel-2015B-Python-3.4.3

<<README
     BLAST manual: http://www.ncbi.nlm.nih.gov/books/NBK279690/
README

# blastx: search protein databases using a translated nucleotide query

blastx -query mrna_seqs_nt.fasta -db /scratch/datasets/blast/nr \
-outfmt 10 -out mrna_seqs_nt_blastout.csv
```

Sample GCATemplate Job Script (Ada)

```
#BSUB -L /bin/bash
#BSUB -J blastx
#BSUB -n 1
#BSUB -R "span[ptile=1]"
#BSUB -R "rusage[mem=2500]"
#BSUB -M 2500
#BSUB -W 2:00
#BSUB -o stdout.%J
#BSUB -e stderr.%J

module load BLAST+/2.2.31-intel-2015B-Python-3.4.3

<<README
     BLAST manual: http://www.ncbi.nlm.nih.gov/books/NBK279690/
README

# blastx: search protein databases using a translated nucleotide query

blastx -query mrna_seqs_nt.fasta -db /scratch/datasets/blast/nr \
-outfmt 10 -out mrna_seqs_nt_blastout.csv
```

These parameters are read by the job scheduler

Load the required module(s) first

This is a section of comments

This is a single line comment and not run as part of the script

This is the command to run the application

This means the command is continued on the next line;
The space before the \ is required
Do not put a space after the \

Quality Control (QC)

QC Evaluation

- Use FastQC to visualize quality scores
  - Displays quality score distribution of a subset of ~200,000 reads
    - Input is a fastq file or files
    - Can disable grouping (binning) of sequence regions
  - Will alert you of poor read characteristics
  - Can be run as a GUI or a command line interface

```
module load FastQC/0.11.5-Java-1.7.0_80
```

- FastQC will process using one CPU core per file
  - If there are 10 fastq files to analyze and 4 cores used
    - 4 files will start processing and 6 will wait in a queue
  - If there is only one fastq file to process then using 10 cores does not speed up the process

FastQC Exercise

- Use the GCATemplate for FastQC to submit a job evaluating the two sequence files
  - `gedit run_fastqc_0.11.5_ada.sh &`
  - `bsub < run_fastqc_0.11.5_ada.sh`
- After your fastqc job is complete, unzip the results file and you can view the results files with **`lynx`** and **`eog`** (eog requires X11 login)
  - `unzip DR34_R1_fastqc.zip`

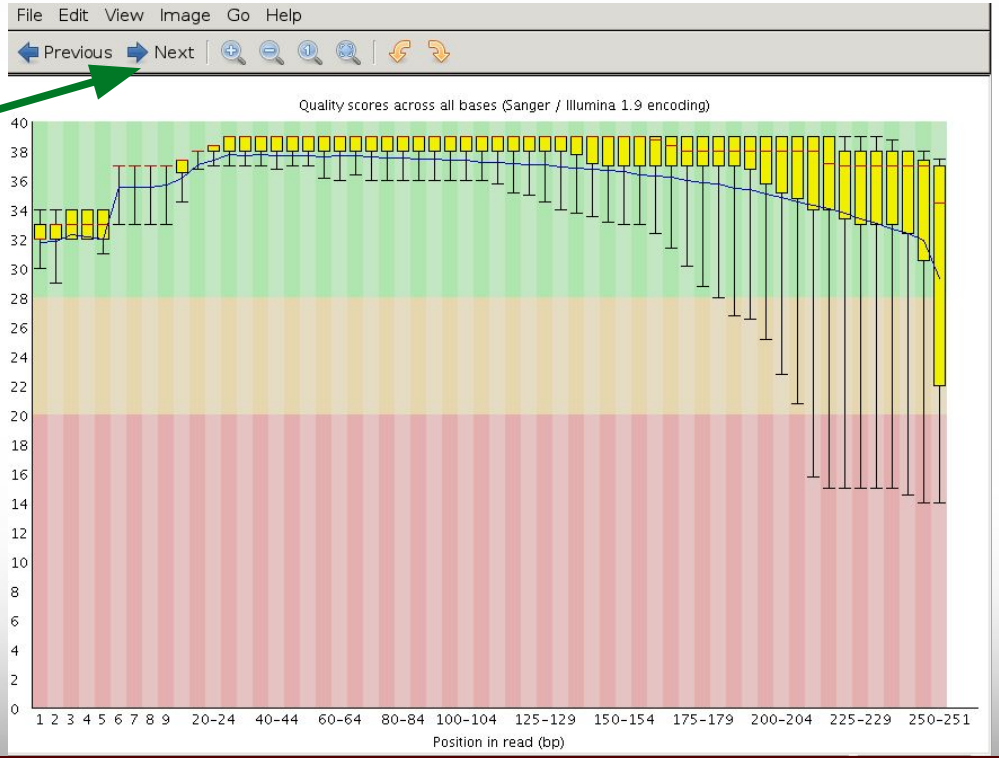
FastQC Report using lynx

```
lynx DR34_R1_fastqc.html
```

FastQC Output Image Quality Distribution

```
eog DR34_R1_fastqc/Images/per_base_quality.png
```

click for the next image in the same directory, or use the left/right arrow keys

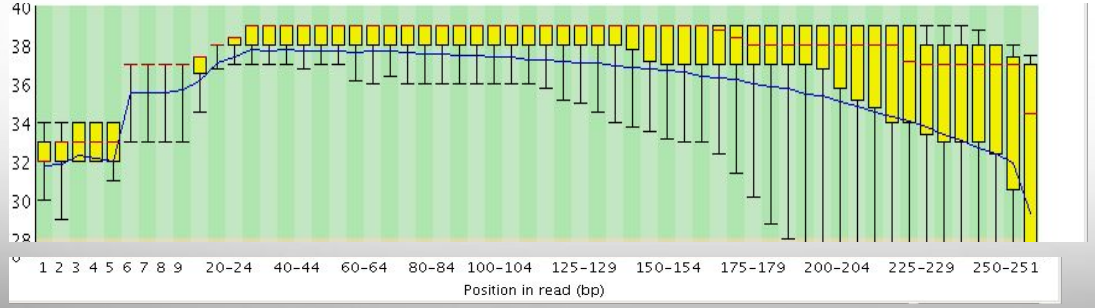


Prior to QC trimming

FastQC Output Image Quality Distribution

FASTQ format

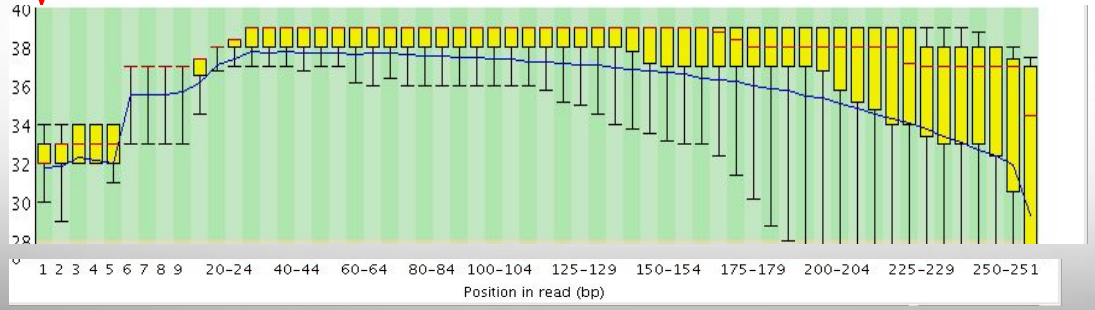
```
@ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGATCAGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAAACAAAAAACATAATGCCGTAAA
+ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
=:=4AD=B8A:+<A::1<:AE<C3*?F<B???<?:8:6?B*9BD;/638.=-'-.@7=).=A:6?DDDCBBBB9555&&)+((+2&&+(((((((()&&&+
@ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
GATGTTTTGTTACTGATTGGAACCATGATTGGTGCTTTACTTGGTTTCTTCCTATTTAACCACAAGCCTGCCAAAGTATTTATGGGAGATGTAGGTAGTT
+ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
BCCFDEFFHHHHHJJJJIJJJJJJJJJJIJJJFHIJJJIJJJJJIJJJJJJJIJJJIJIIJJJJJFHJJJJIJJHH=CHHFFFFFFEDDEDEEEDCCDCE
@ERR504787.5.1 M00368:15:000000000-A0HKH:1:2:16161:12630-1 length=100
TATTTTAAGTGACCAAGGAATGACTCCCCAATCATGGCTGTATCAACTCCAAAATTTTCTGCAACAGTCGCTGAAATATCTGCAAAATGCCCTTGTGGAA
+ERR504787.5.1 M00368:15:000000000-A0HKH:1:2:16161:12630-1 length=100
CCCFFFFFHHHHHJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJIJJJHIJHHGFFFFFFFEEEEEEDDDDDDDDDCDDDD
```

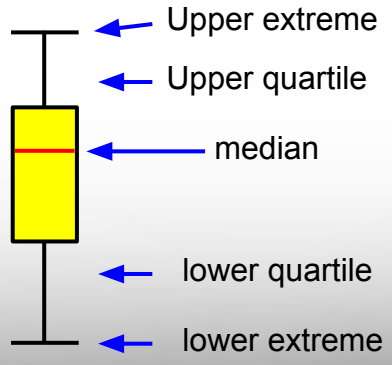


FastQC Output Image Quality Distribution

```
@ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGATCAGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAACAAAAAACATAATGCCGTAAA
+ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
=:=4AD=B8A:+<A::1<:AE<C3*?F<B???<?:8:6?B*9BD;/638.=-'-.@7=).=A:6?DDDCBBBB9555&&)+((+2&&+(((((((()&&&+
@ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
GATGTTTTGTTACTGATTGGAACCATGATTGGTGCTTTACTTGGTTTCTTCCTATTTAACCACAAGCCTGCCAAAGTATTTATGGGAGATGTAGGTAGTT
+ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
BCCFDEFFHHHHHJJJJIJJJJJJJJJJIJJJFHIJJJIJJJJJIJJJJJJJIJJJIJIIJJJJJFHJJJJIJJHH=CHHFFFFFFEDDEDEEEDCCDCE
@ERR504787.5.1 M00368:15:000000000-A0HKH:1:2:16161:12630-1 length=100
TATTTTAAGTGACCAAGGAATGACTCCCCAATCATGGCTGTATCAACTCCAAAATTTTCTGCAACAGTCGCTGAAATATCTGCAAAATGCCCTTGTGGAA
+ERR504787.5.1 M00368:15:000000000-A0HKH:1:2:16161:12630-1 length=100
CCCFFFFFHHHHHJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJIJJJHIJHHGFFFFFFFEEEEEEDDDDDDDDDCDDDD
```

Average quality score distribution at position one

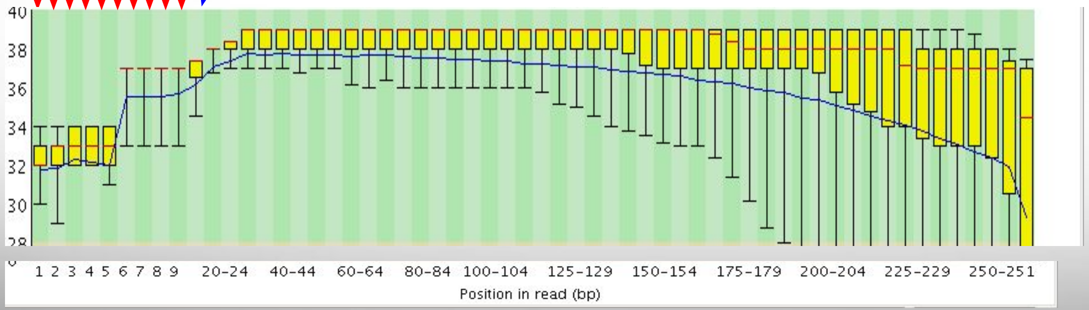




FastQC Output Image Quality Distribution

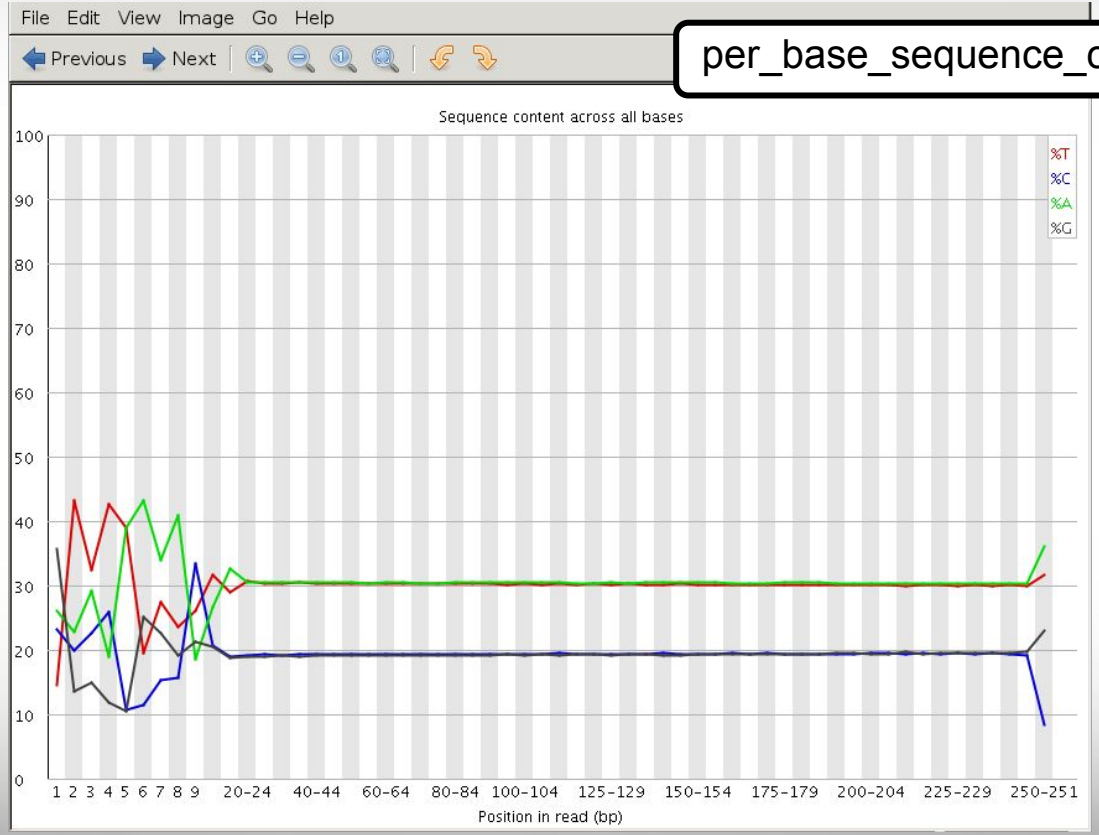
```
@ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGATCAGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAAAAAAAACAAAAAACATAATGCCGTAAA
+ERR504787.2.1 M00368:15:000000000-A0HKH:1:5:21261:10968-1 length=100
=:=4AD=B8A:+<A::1<:AE<C3*?F<B???<?:8:6?B*9BD;/638.=-'-.@7=).=A:6?DDDCBBBB9555&&)+((+2&&+((((((()&&&+
@ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
GATGTTTTGTTACTGATTGGAACCATGATTGGTGCTTTACTTGGTTTCTTCCTATTTAACCACAAGCCTGCCAAAGTATTTATGGGAGATGTAGGTAGTT
+ERR504787.3.1 M00368:15:000000000-A0HKH:1:3:12724:25677-1 length=100
BCCFDEFFHHHHHJJJJIJJJJJJJJJJIJJJFHIJJJIJJJJJIJJJJJJJIJJJIJIIJJJJJFHJJJJIJJHH=CHHFFFFFFEDDEDEEEDCCDCE
@ERR504787.5.1 M00368:15:000000000-A0HKH:1:2:16161:12630-1 length=100
TATTTTAAGTGACCAAGGAATGACTCCCCAATCATGGCTGTATCAACTCCAAAATTTTCTGCAACAGTCGCTGAAATATCTGCAAAATGCCCTTGTGGAA
+ERR504787.5.1 M00368:15:000000000-A0HKH:1:2:16161:12630-1 length=100
CCCFFFFFHHHHHJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJIJJJHIJHHGFFFFFFFEEEEEEDDDDDDDDDCDDDD
```

Positions are ‘binned’ after the first few positions



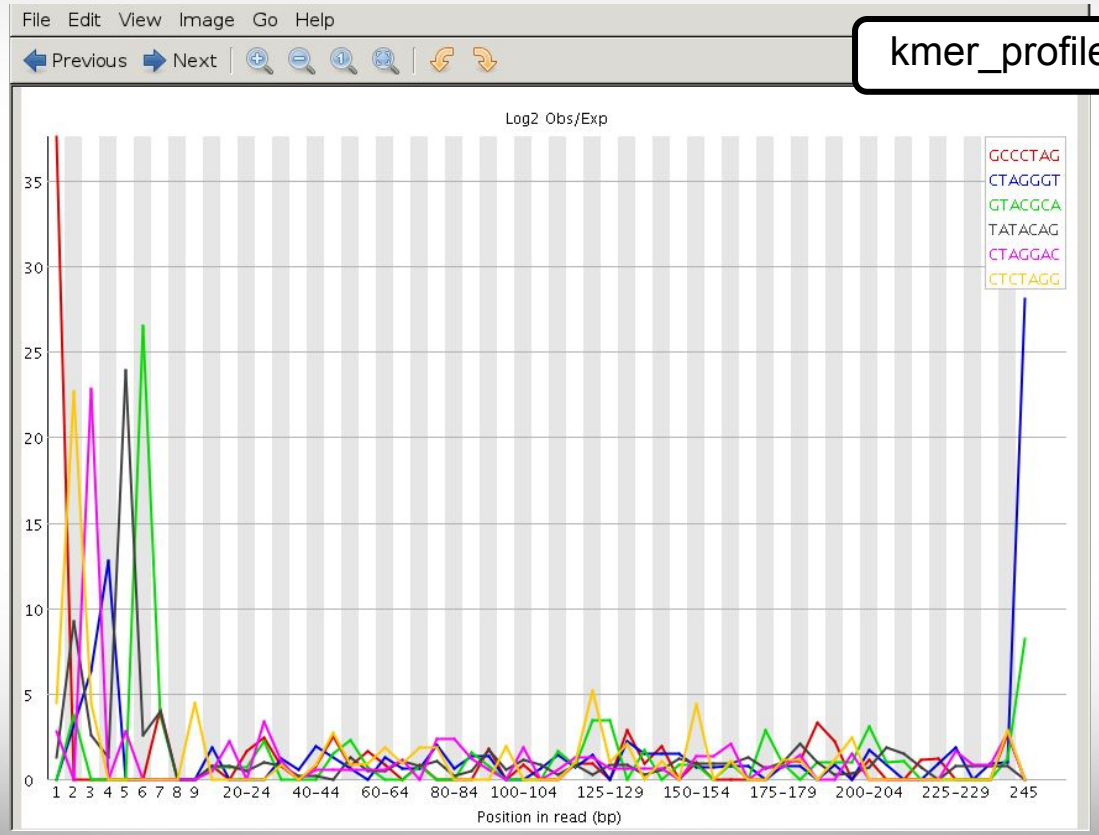
Illumina Transposon Insertion Site

per_base_sequence_content.png



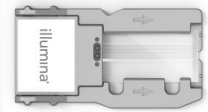
Illumina Transposon Insertion Site

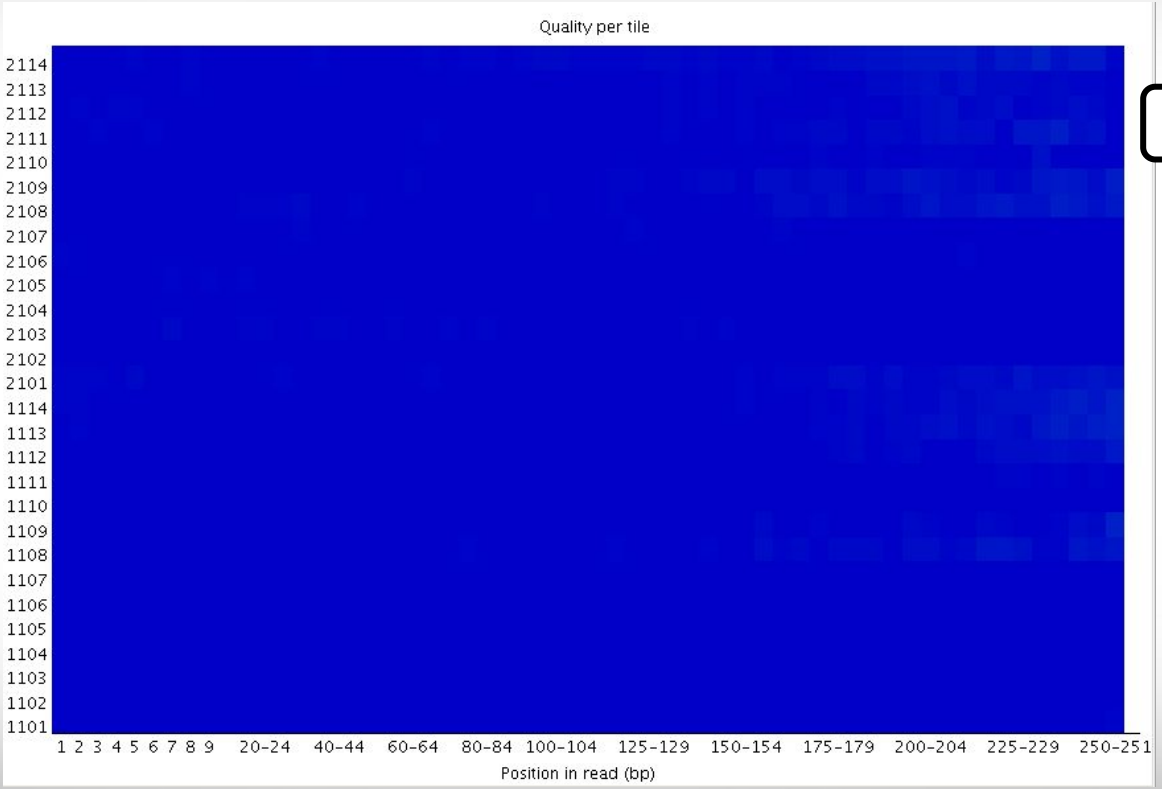
kmer_profiles.png



FastQC Flowcell Quality Image

MiSeq flowcell





per_tile_quality.png

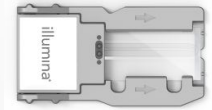
Flowcell quality mapping
Good per_tile quality

good quality  poor quality

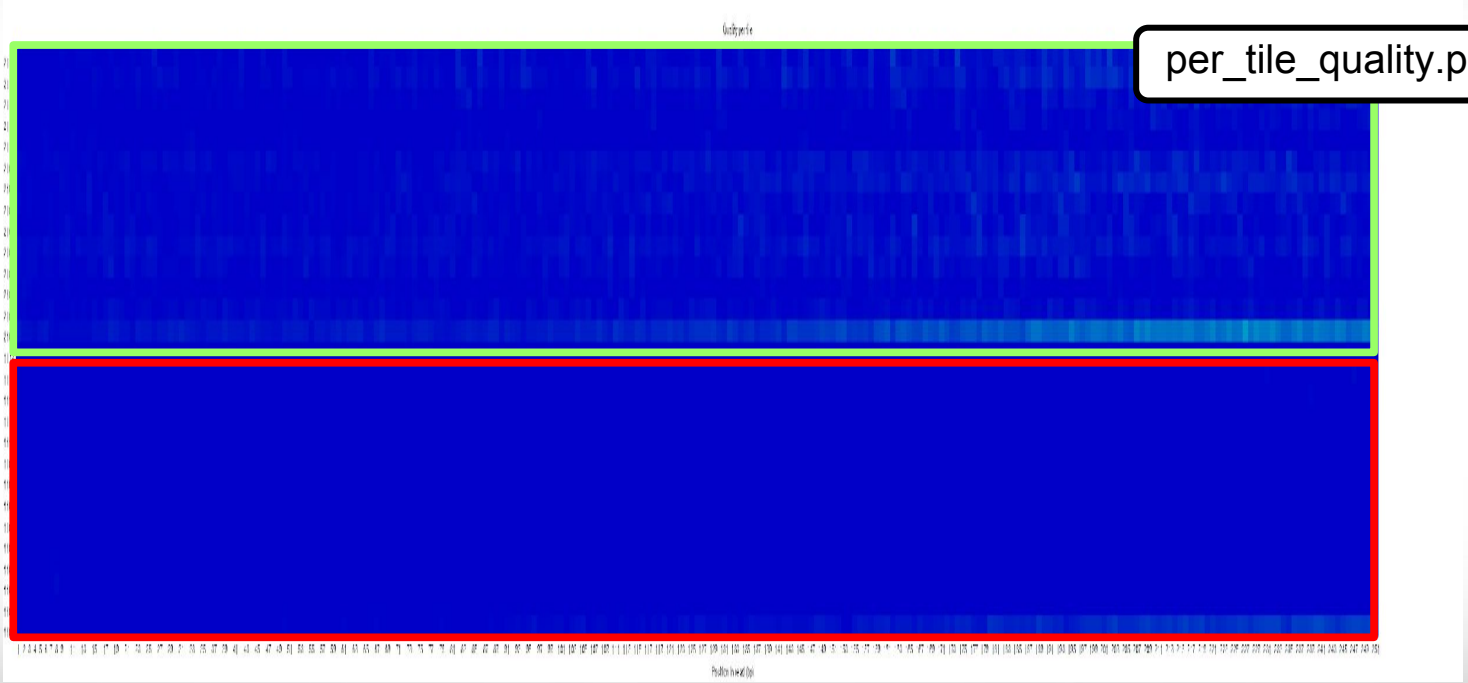
FastQC Flowcell Quality Image

bottom of flowcell

MiSeq flowcell



top of flowcell

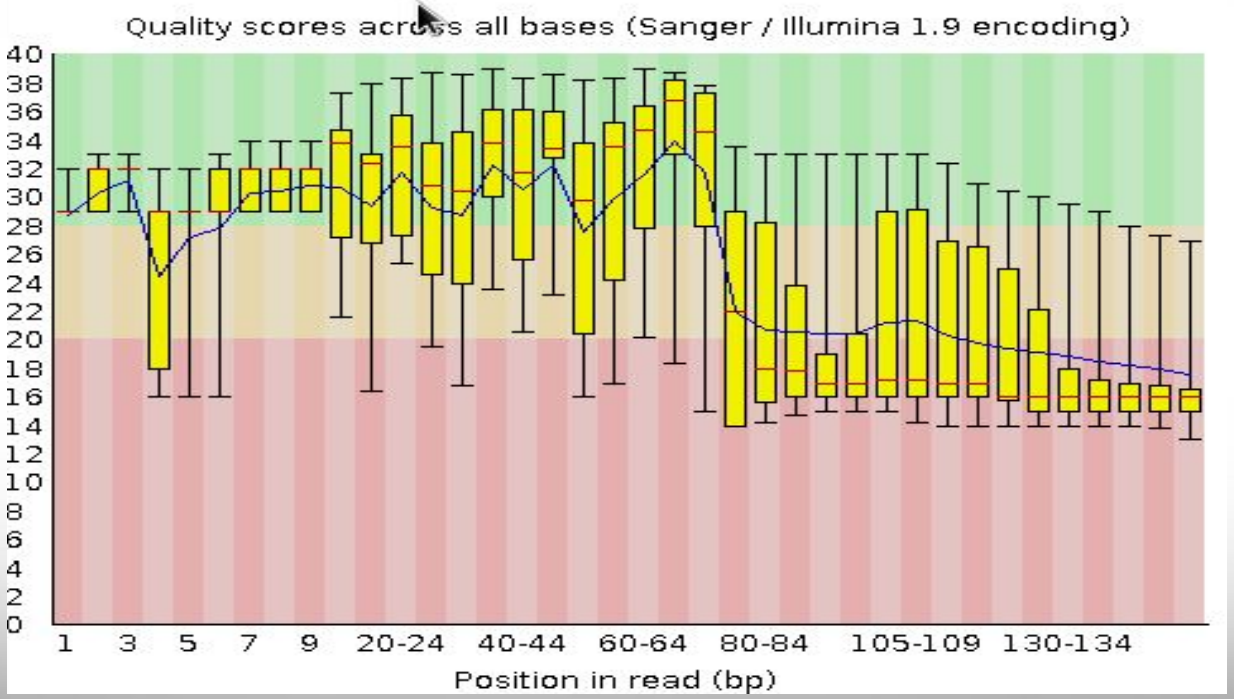


good quality — poor quality

Failed QC Examples

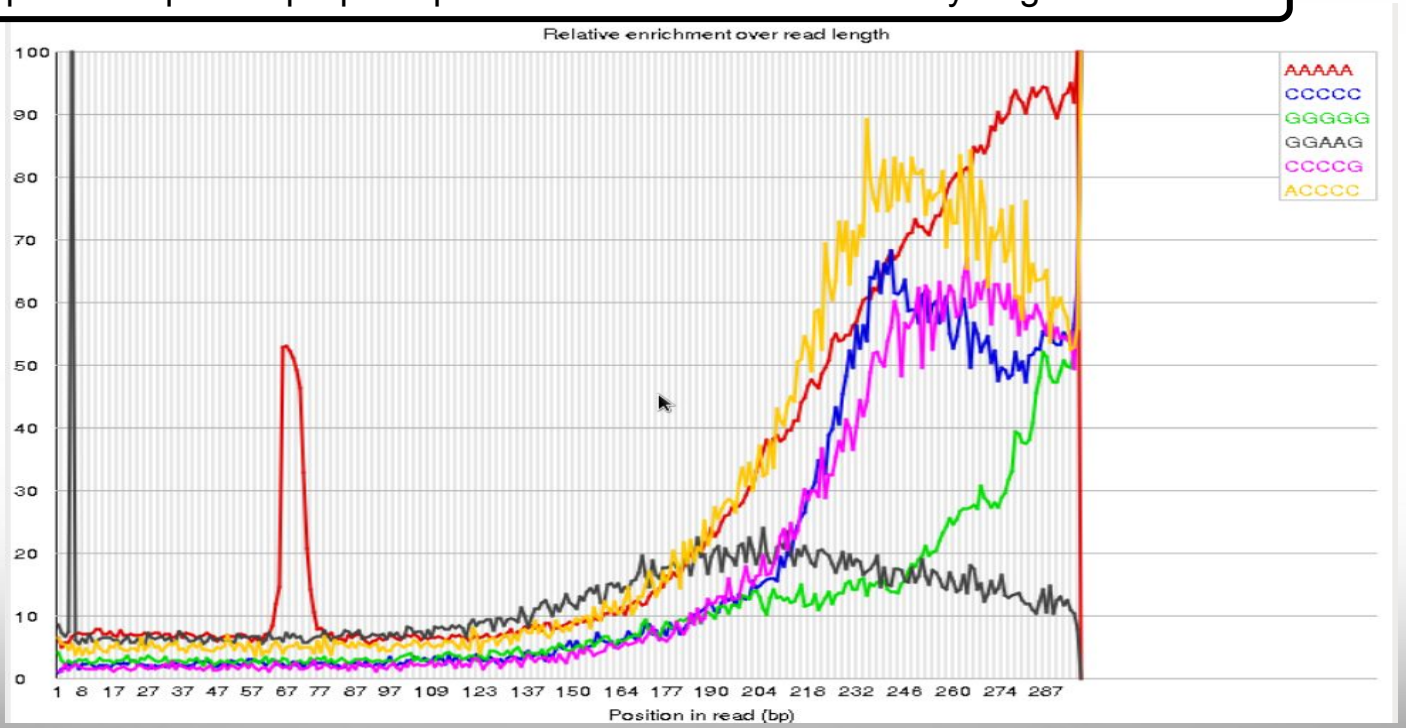
FastQC Output Image
Failed Per base sequence quality

Example 1. Expired MiSeq mate-pair kit (9 months expired)



FastQC Output Image
Failed Kmer Content

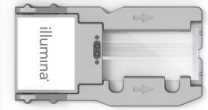
Example 2. Sequence prep adapters still on ends of DNA library fragments

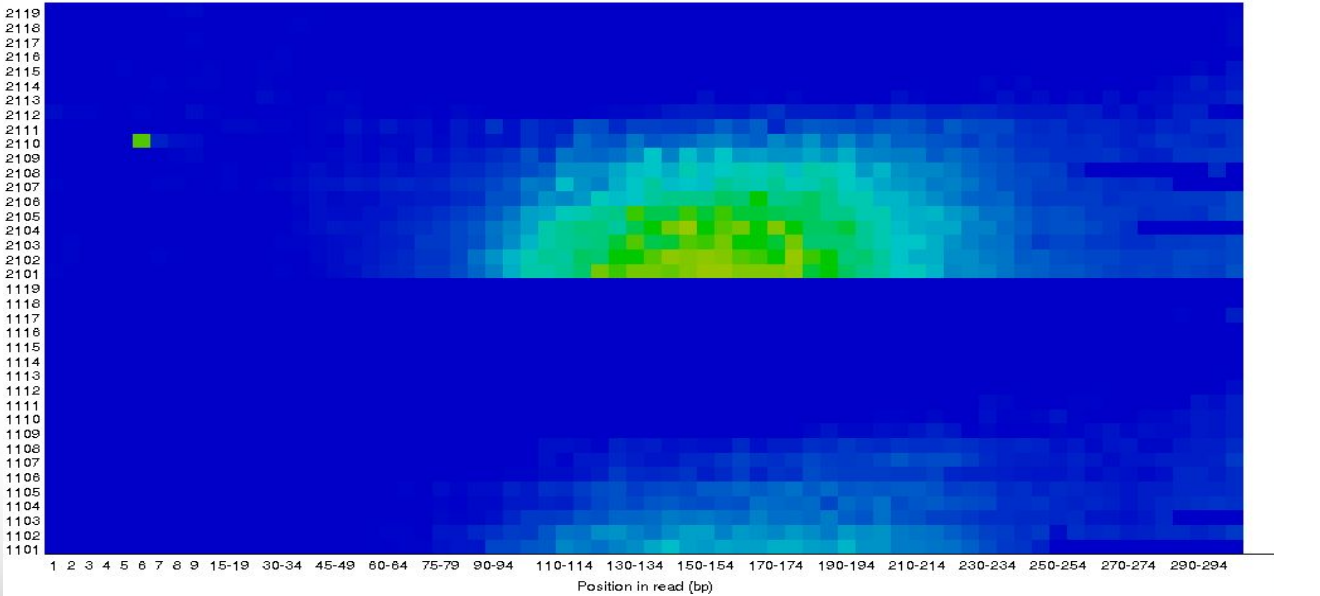


FastQC Output Image
Flowcell: not good per_tile quality

Example 3. Faulty flowcell

MiSeq flowcell





good quality — poor quality

QC Quality Trimming

- Sequence quality trimming tools

```
module spider Trimmomatic
```

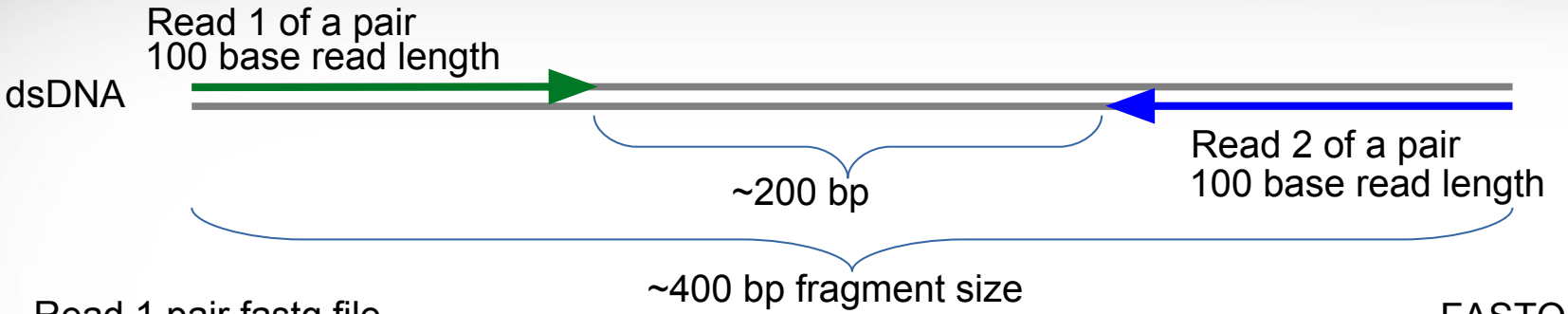
recommended tool

- Trimmomatic will maintain paired end read pairing after trimming
- Trim reads based on quality scores
  - Trim the same number of bases from each read or
  - Use a sliding window to calculate average quality at ends of sequences
- Decide if you want to discard reads with Ns
  - some assemblers replace Ns with As or a random base G, C, A or T
- Trim adapter sequences
  - Trimmomatic has a file of Illumina adapter sequences

```
module load Trimmomatic/0.36-Java-1.8.0_92
```

```
ls $EBROOTTRIMMOMATIC/adapters/
```

Paired End Short Reads



Read 1 pair fastq file

FASTQ format

```
@M00861:1:000000000-A36BE:1:1101:14650:1529 1:N:0:8
TTCTTAAAAATACCATAAAAGGCTTAAACTTGCCATTTACGACGGATTAATTCCAACTCTTTTCGGCTATCTTCATCTTTTAAGGTTAAATGACTCATAACGG
+
FFFHBFFHHIIIIIIHFHHCGEFGHHIHHHIHD/?DGGHHH@DEB,5EGHGHHIIHIF?FGGHHCCBFDGHFHDGHGFFFGDFHH?DFHDFFHHFHFHFFHHH
```

Read 2 pair fastq file

```
@M00861:1:000000000-A36BE:1:1101:14650:1529 2:N:0:8
ACTAAAAATCAATTTTATCAATTTCAAGCTCTACCTTATTTACTCATTATTTTAGTGATGGCCACTTTAATAAAAATATTGGTAGCATATTTTGCAATAGCGG
+
BFFHIHHHFHHDGHIHHIHHHGHHHHHFHHDFFHIHIIIHIHDFHHHIHIIIH=AAFHIIIHFGFHHHHHGGHHIHHFGFFFEGGHHHDGHHH/CGHIFFHHH
```

Trimming PE Short Sequence Reads

File 1 from sequencer

100 bases

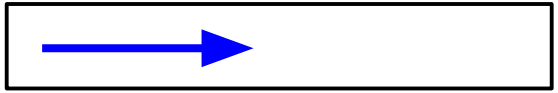
QC trim

100 bases

File 2 from sequencer

100 bases

QC trim

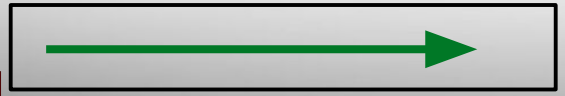


50 bases

minimun read length = 40

Resulting FASTQ Files with trimmed reads

Paired end 1 trimmed file



Paired end 2 trimmed file

Trimming PE Short Sequence Reads

File 1 from sequencer

100 bases

QC trim

100 bases

File 2 from sequencer

100 bases

QC trim

20 bases

minimun read length = 40

Resulting FASTQ Files with trimmed reads

Paired end 1 trimmed file

Paired end 2 trimmed file

Single end reads

Merge Overlapping Paired End Short Reads

fragment 1

dsDNA

---

fragment 2

dsDNA

Merge Overlapping Paired End Short Reads

fragment 1

dsDNA

fragment 2

dsDNA

Paired end read 1 (left)

Paired end read 2 (right)

Merge Overlapping Paired End Short Reads

fragment 1

dsDNA

fragment 2

dsDNA

Paired end read 1 (left)

Paired end read 2 (right)

Unpaired ‘merged’ read

Tools for merging overlapping reads:

```
module spider FLASH
module spider Coperead
module spider PEAR
```

Trimmomatic Exercise using GCATemplates on Ada

Genomic Computational Analysis Templates

```
gcatemplates
```

For practice, we will copy a template file

- Select #4 then find the template that contains trimmomatic
- Save the template script to your pwd
- Review the template script contents
- submit the template script to the scheduler
- Review the output files

Mapping Reads to a Reference Assembly

Mapping Reads to a Reference Assembly

- Align reads using bwa
  - `module spider BWA`
    - bwa index files for UCSC genomes found here
    - `/scratch/datasets/genome_indexes/ucsc/mm10/bwa_0.7.12_index/`
- Align reads using bowtie or bowtie2
  - `module spider Bowtie`
    - Bowtie index files for UCSC genomes found here:
    - `/scratch/datasets/genome_indexes/ucsc/mm10/bowtie_index/`
  - `module spider Bowtie2`
    - Bowtie2 index files for UCSC found here:
    - `/scratch/datasets/genome_indexes/ucsc/mm10/bowtie2_index/`

Visualize bam File Alignments

Sample bam and reference files

```
cd $SCRATCH/ngs_class
```

For this samtools demo, add symbolic links* to the example files in your working directory

```
ln -s /scratch/training/intro_to_ngs/alignments/dr34.sam
```

Add a symbolic link to the example reference genome fasta file

```
ln -s /scratch/training/intro_to_ngs/genomes/c_dubliniensis.fa
```

Use the tab key when typing these long paths

* The symbolic links are used to make the commands shorter for demonstration purposes only. You do not need to make symbolic links in order to use `samtools tview`

Sorting Alignment sam/bam Files

- Sequence Alignment/Map format (sam)
  - view sam files using the UNIX command:
```
more dr34.sam
```
- Binary Alignment/Map format (bam)
  - Compressed (binary) sam files need samtools to view
    -
```
module load SAMtools/1.7-GCCcore-6.3.0
```
  - Recommended: sort sam/bam file based on coordinate into bam format
  -
```
samtools sort -@ 1 -m 2G -o dr34.bam dr34.sam
```
  - Create an index of the bam file using samtools
    - A samtools index is needed prior to viewing bam files in browsers

```
samtools index dr34.bam
```

```
dr34.bam.bai
```

Viewing sam/bam Files

Viewing bam files using samtools

```
samtools view dr34.bam | more
```

view only alignments

```
samtools view -H dr34.bam
```

view only header

```
samtools view -h dr34.bam | more
```

view header + alignments

Alignment Statistics

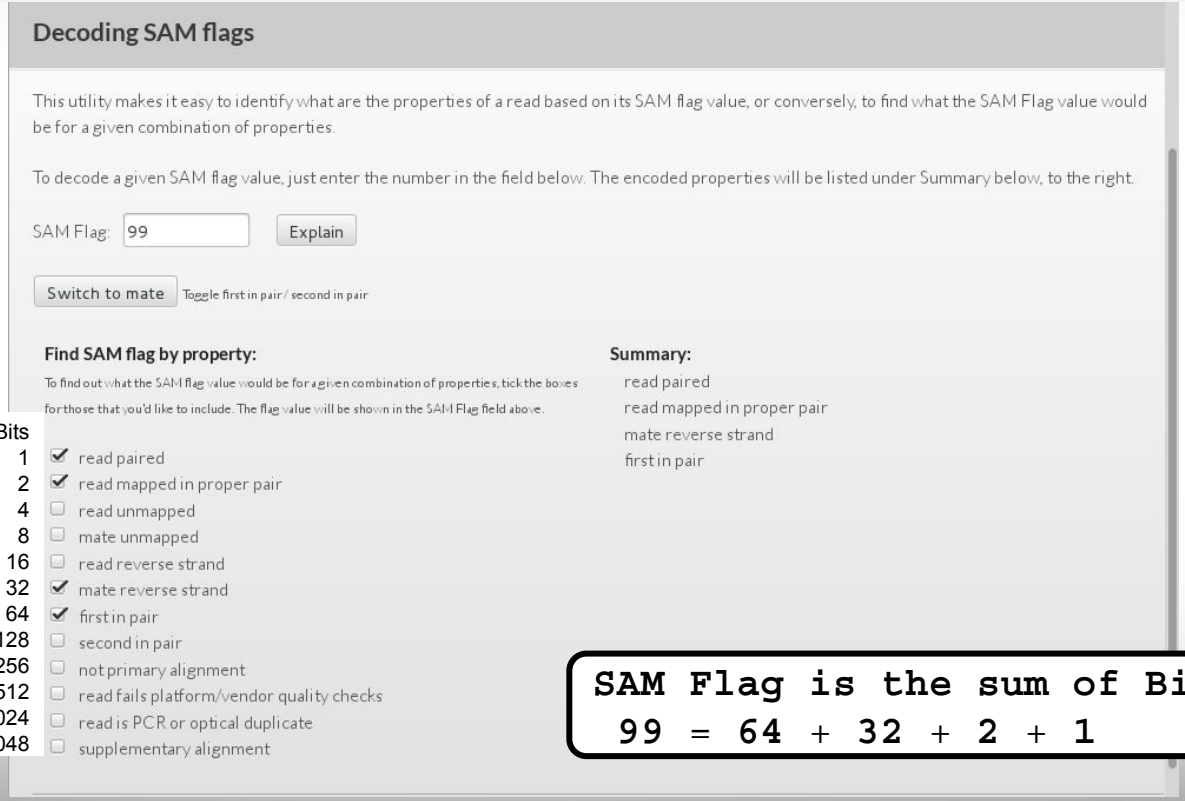
```
samtools flagstat dr34.bam
```

```
150000 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
140150 + 0 mapped (93.43% : N/A)
150000 + 0 paired in sequencing
75002 + 0 read1
74998 + 0 read2
85639 + 0 properly paired (57.09% : N/A)
136854 + 0 with itself and mate mapped
3296 + 0 singletons (2.20% : N/A)
909 + 0 with mate mapped to a different chr
56 + 0 with mate mapped to a different chr (mapQ>=5)
```

Both reads in the pair are mapped on the same chromosome and in FR or RF orientation

Sam Flags and Bits

https://broadinstitute.github.io/picard/explain-flags.html

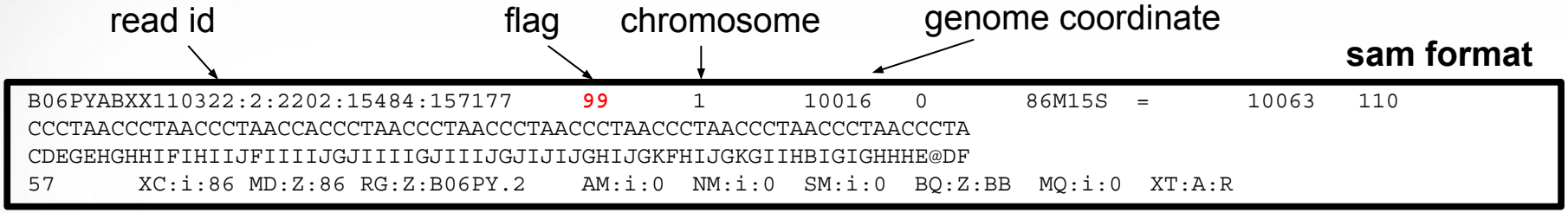


SAM Flag is the sum of Bits

99 = 64 + 32 + 2 + 1

Sam Flags and Bits

- Flags describe alignments (the flag value is the sum of bits)



bits: **1** **2** 4 8 16 **32** **64** 128 256 512 1024 2048

1 + 2 + 32 + 64 = 99

- Filter bam alignments based on bit in flag (-f and/or -F)
  - Keep only reads that are 'mapped in proper pair'

```
samtools view -h -b -f 2 dr34.bam > dr34_paired_reads.bam
```

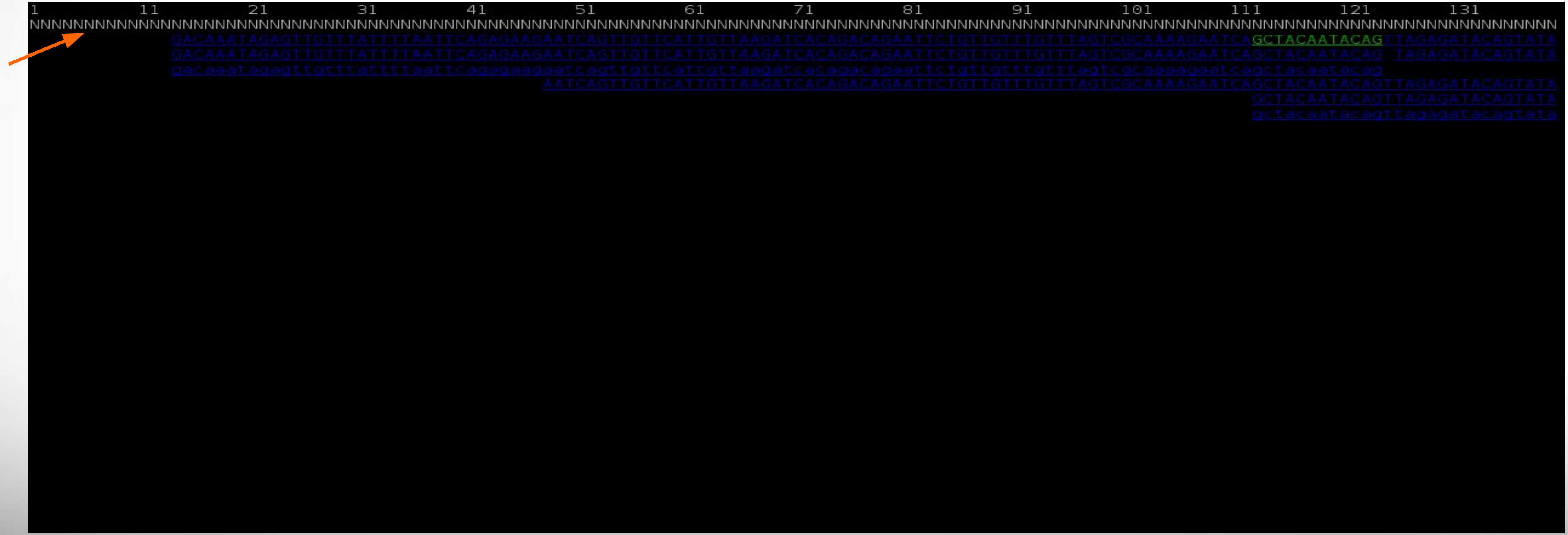
  - Keep all except reads that are 'PCR or optical duplicate'

```
samtools view -h -b -F 1024 dr34.bam > dr34_dedup_reads.bam
```

SAMtools without a Reference Genome

Reference genome represented on top as NNNNNNNNN

```
samtools tview dr34.bam
```



SAMtools with a Reference Genome

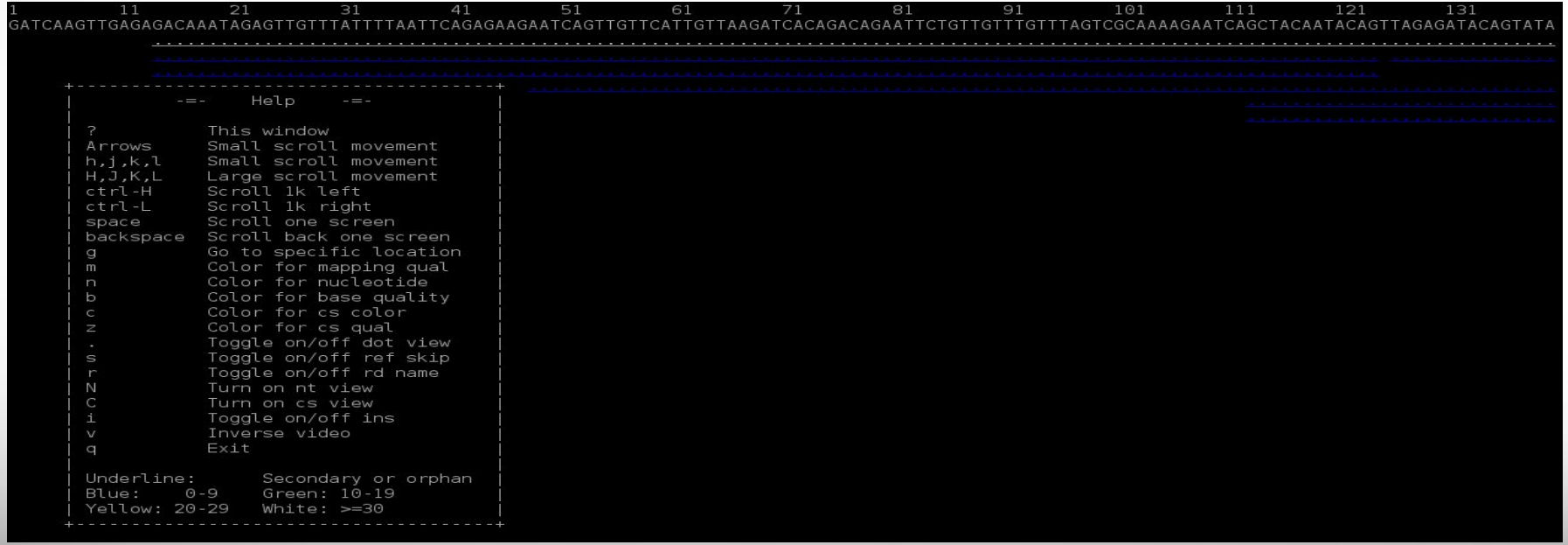
Reference genome sequence displayed on top

```
samtools tview dr34.bam c_dubliniensis.fa
```

SAMtools with a Reference Genome

Type ? for help menu

```
samtools tview dr34.bam c_dubliniensis.fa
```

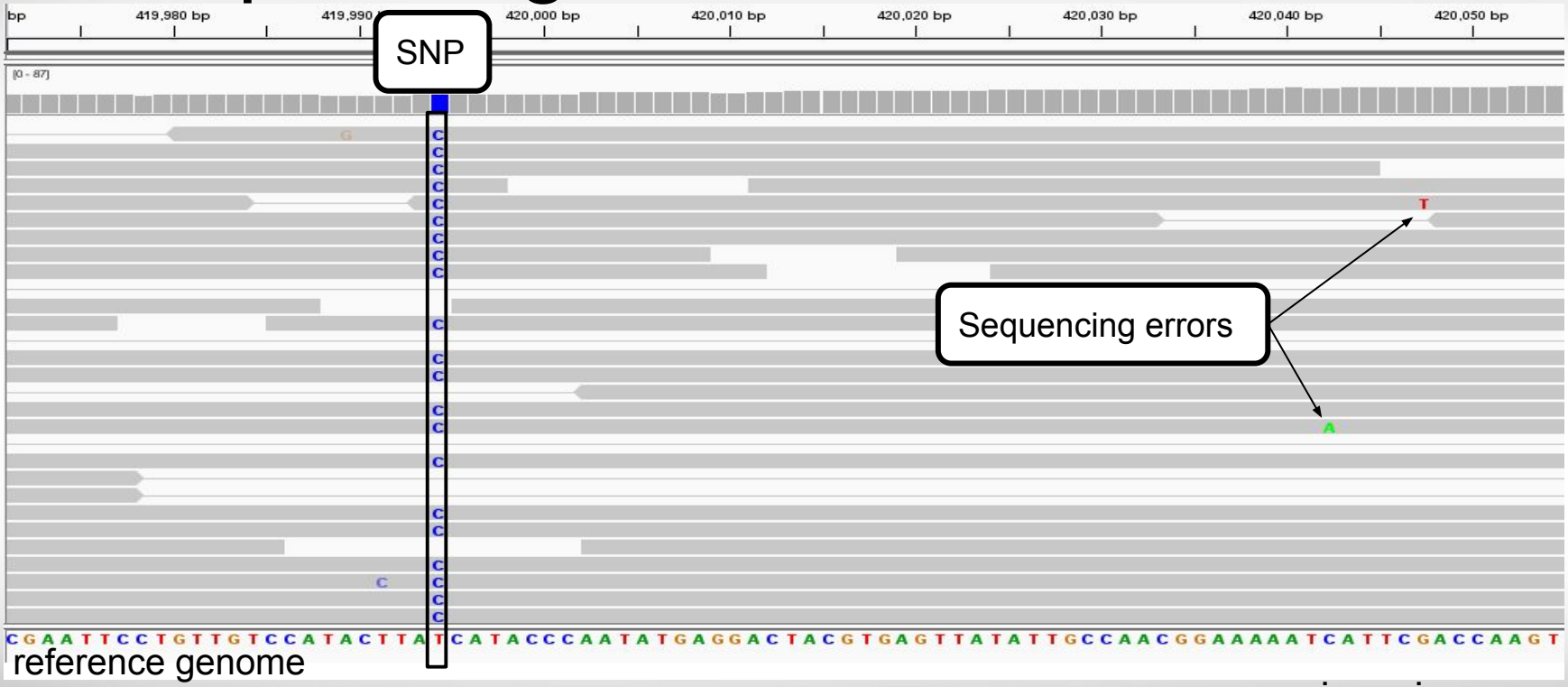


View at a Specific Coordinate

```
samtools tview dr34.bam c_dubliniensis.fa -p 1:315398
```

Sequence Error Correction In Short Reads

Sequencing Errors in Short Reads



Tool for correcting sequencing errors: `module spider Lighter`

Digital Normalization

Digital Normalization

Reduce memory requirements by reducing the number of redundant sequence reads if you have a very high sequencing coverage (> 200x)

```
module spider BBMap
```

Use the **bbnorm.sh** script in the BBMap module

A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data

C. Titus Brown[1,2,*], Adina Howe[2], Qingpeng Zhang[1], Alexis B. Pyrkosz[3], Timothy H. Brom[1]
1 Computer Science and Engineering, Michigan State University, East Lansing, MI, USA
2 Microbiology and Molecular Genetics, Michigan State University, East Lansing, MI, USA
3 USDA Avian Disease and Oncology Laboratory, East Lansing, MI, USA
* E-mail: ctb@msu.edu

Sequence Variant Calling

Sequence Variant Calling

- Start with aligning reads to a reference
  - GATK does not require QC trimming
  - Mark PCR duplicates with Picard
- Differentiate between sequencing errors and SNPs
  - Calling SNPs may require a min read depth of 10x (higher for indels)
  - Calling variants may require 1/3 of reads to contain SNP
  - Strand bias may result as a consequence of the sequencing chemistry's response to certain DNA sequence motifs but it can be detected computationally
- BLAST reads with SNPs to identify variant calls due to misalignments especially with duplicated genes
- Variant Call Format (vcf) – standard format of variant calls
- Identify multiple-nucleotide polymorphism (MNP)
  - Two SNPs within a single codon

	codon	translation
Reference:	TTT	Phe
SNP 1:	TTA	Leu
SNP 2:	TAT	Tyr
SNP 1 + 2:	TAA	STOP

Marking PCR Duplicates

- PCR duplicates are artifacts resulting from a PCR amplification step during NGS library preparations.
- PCR duplicates should be removed/marked as to not bias the frequency of variants or gene expression levels
  - Use picard tools to mark duplicates
  - Freebayes will ignore marked duplicates during variant calling

```
module spider picard
```

Variant Calling Tools

Use bam file of sequence reads aligned to a reference as input for the following four work flows

1. GATK

```
module spider GATK picard SAMtools
```

   - No need to QC trim reads, the GATK best practices pipeline will perform the necessary steps including marking PCR duplicates
   - You need a set of known variants for your species (dbSNP) or you can bootstrap your population to get variant frequency
   - Used in conjunction with other tools
     - samtools
     - picard

2. SAMtools and BCFtools

```
module spider SAMtools BCFtools
```

3. VarScan

```
module spider VarScan
```

4. FreeBayes

```
module spider FreeBayes
```

Sample vcf File Format

```
##fileformat=VCFv4.0
##fileDate=20110705
##reference=1000GenomesPilot-NCBI37
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF  ALT     QUAL FILTER INFO                              FORMAT      Sample1         Sample2
2      4370    rs6057    G    A       29   .      NS=2;DP=13;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:52,51  1|0:48:8:51,51
2      7330    .         T    A       3    q10    NS=5;DP=12;AF=0.017               GT:GQ:DP:HQ 0|0:46:3:58,50  0|1:3:5:65,3
2      110696  rs6055    A    G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27  2|1:2:0:18,2
2      130237  .         T    .       47   .      NS=2;DP=16;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60  0|0:48:4:56,51
2      134567  microsat1 GTCT G,GTACT 50   PASS   NS=2;DP=9;AA=G                   GT:GQ:DP    0/1:35:4        0/2:17:2
```

3 more columns not shown due to width of rows

vcf File Column Descriptions

```
##fileformat=VCFv4.0
##fileDate=20110705
##reference=1000GenomesPilot-NCBI37
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF  ALT     QUAL FILTER INFO                              FORMAT      Sample1         Sample2
2      4370    rs6057    G    A       29   .      NS=2;DP=13;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:52,51  1|0:48:8:51,51
2      7330    .         T    A       3    q10    NS=5;DP=12;AF=0.017               GT:GQ:DP:HQ 0|0:46:3:58,50  0|1:3:5:65,3
2      110696  rs6055    A    G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27  2|1:2:0:18,2
2      130237  .         T    .       47   .      NS=2;DP=16;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60  0|0:48:4:56,51
2      134567  microsat1 GTCT G,GTACT 50   PASS   NS=2;DP=9;AA=G                   GT:GQ:DP    0/1:35:4        0/2:17:2
```

variants that are phased are inherited together

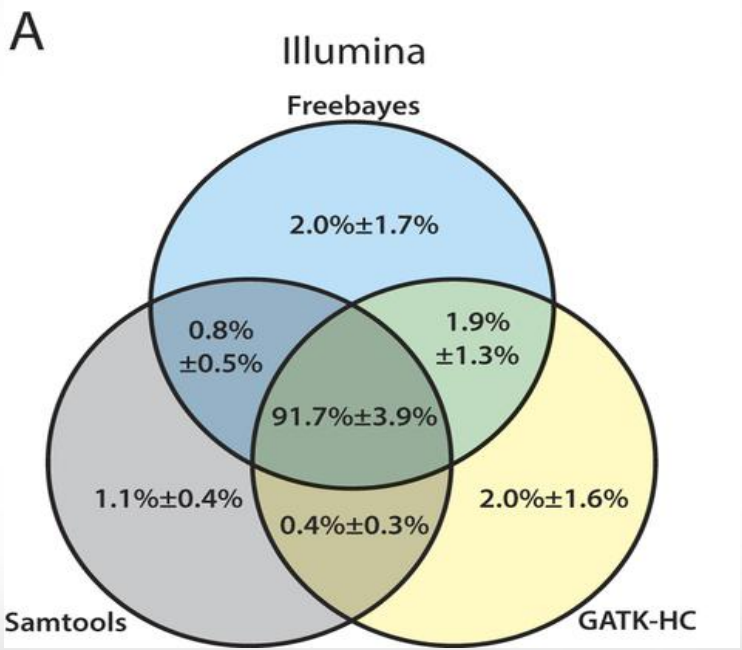
- | indicates phased variants
- / indicates non-phased variants

Sample1 haplotypes: **GTGT** and **GTTT**

Sample2 haplotypes: **ATTT** and **GAGT**

https://www.broadinstitute.org/gatk/guide/tagged?tag=phasing

Summarizing Variant Calls from Different Tools



The mean percentage with standard deviation of confidence variant calls with equal to or higher than the quality score threshold of 20 are represented for (A) Illumina data sets

Huang et al 2015 doi:10.1038/srep17875

Consequence of Amino Acid Change

- Assess consequence of amino acid change based on sequence conservation across multiple species using the PROVEAN tool
- Variants with a score equal to or below -2.5 are considered “deleterious”

```
module spider PROVEAN
```

```
## PROVEAN v1.1 output ##
# Query sequence file:  CTRG_00013.fa
# Variation file:   CTRG_00013.var
# Protein database: /scratch/datasets/blast/nr
[16:01:13] searching related sequences...
[16:16:36] clustering subject sequences...
# Number of clusters:   30
# Number of supporting sequences used: 245
[16:18:39] computing delta alignment scores...
## PROVEAN scores ##
# VARIATION SCORE
A431S   -0.455
E411K   -3.051
E226Q   -1.564
```

Verify that enough supporting sequences were found

“deleterious”

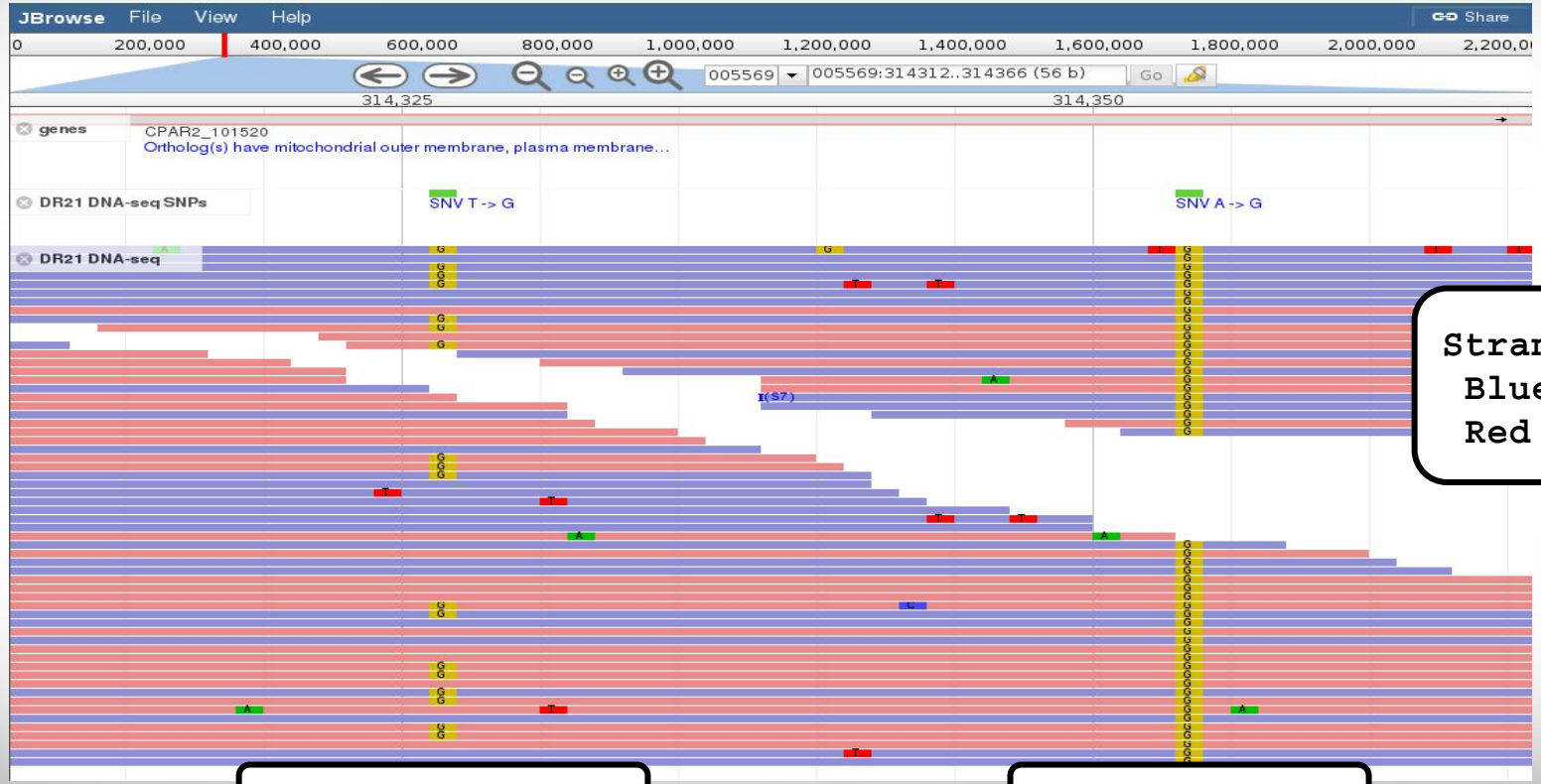
Annotate Variants

```
module spider snpEff
```

- A file of variant calls in vcf format is needed
- A reference sequence with gene annotations is needed
- snpEff annotates a vcf file
  - There are > 2,500 pre-built databases available and you can build your own if needed
  - Annotates MNP (multiple nucleotide polymorphism)
    - Codon change due to two SNPs: ACA → GGA

```
5       325795  .       AC      GG      23.8901 .
AB=0.428571;ABP=3.32051;AC=1;AF=0.5;AN=2;AO=3;CIGAR=2X;DP=7;DPB=7;DPRA=0;EPP=3.73412;
EPPR=3.0103;GTI=0;LEN=2;MEANALT=1;MQM=33;MQMR=48.5;NS=1;NUMALT=1;ODDS=5.49681;PAIRED=0;
PAIREDR=0.5;PAO=0;PQA=0;PQR=0;PRO=0;QA=114;QR=150;RO=4;RPL=3;RPP=9.52472;RPPR=3.0103;
RPR=0;RUN=1;SAF=2;SAP=3.73412;SAR=1;SRF=2;SRP=3.0103;SRR=2;TYPE=mnp;technology.ILLUMINA=1;
ANN=GG|missense_variant|MODERATE|CD36_51230|CD36_51230|transcript|CAX41505.1|
protein_coding|1/1|c.1657_1658delACinsGG|p.Thr553Gly|1657/1851|1657/1851|553/616||
GT:DP:RO:QR:AO:QA:GL    0/1:7:4:150:3:114:-6.7054,0,-11.1847
```

Viewing SNPs in a Diploid Organism



Example of Sequencing Strand Bias

RNA-seq Overview

RNA-seq Applications

- Differential Expression (DE) and transcript abundance
  - HISAT2, Bowtie, TopHat, Cufflinks, Cuffmerge, Cuffdiff
  - DESeq and DESeq2 (R package)
  - EdgeR (R package)
- Transcriptome assembly (find isoforms and rare transcripts)
  - *de novo* (Trinity, Oases, SOAPdenovo-Trans)
  - reference based (Trinity, StringTie)
- Genome Annotation
  - Align to assembly for validation of gene models
- Variant Calling
  - STAR/Picard/GATK (Haplotype Caller (HC) in RNA-seq mode)
- *de novo* genome assembly scaffolding
  - L_RNA_scaffolder
- Identify fusion transcripts
  - tophat-fusion

Sequence Depth for RNA-seq Differential Expression

RNA-seq differential expression studies: more sequence or more replication?
Liu, Yuwen, Zhou, Jie and White, Kevin P. Bioinformatics. 2014 Feb 1; 30(3): 301–304.
doi: 10.1093/bioinformatics/btt688 PMCID: PMC3904521

- Using more biological replicates instead of increasing sequencing depth resulted in improved accuracy of expression estimation
- Use more biological replicates at lower sequencing depth is more beneficial than fewer samples at a higher sequencing depth
- Increasing sequence depth is beneficial for exon or transcript-specific expression studies

RNA-seq Transcriptome Assembly

- Assembly with a reference genome

```
module spider Trinity
```

```
module spider HISAT2 Cufflinks
```

```
module spider Scripture
```

```
module spider StringTie
```

- *de novo a*ssembly without a reference genome

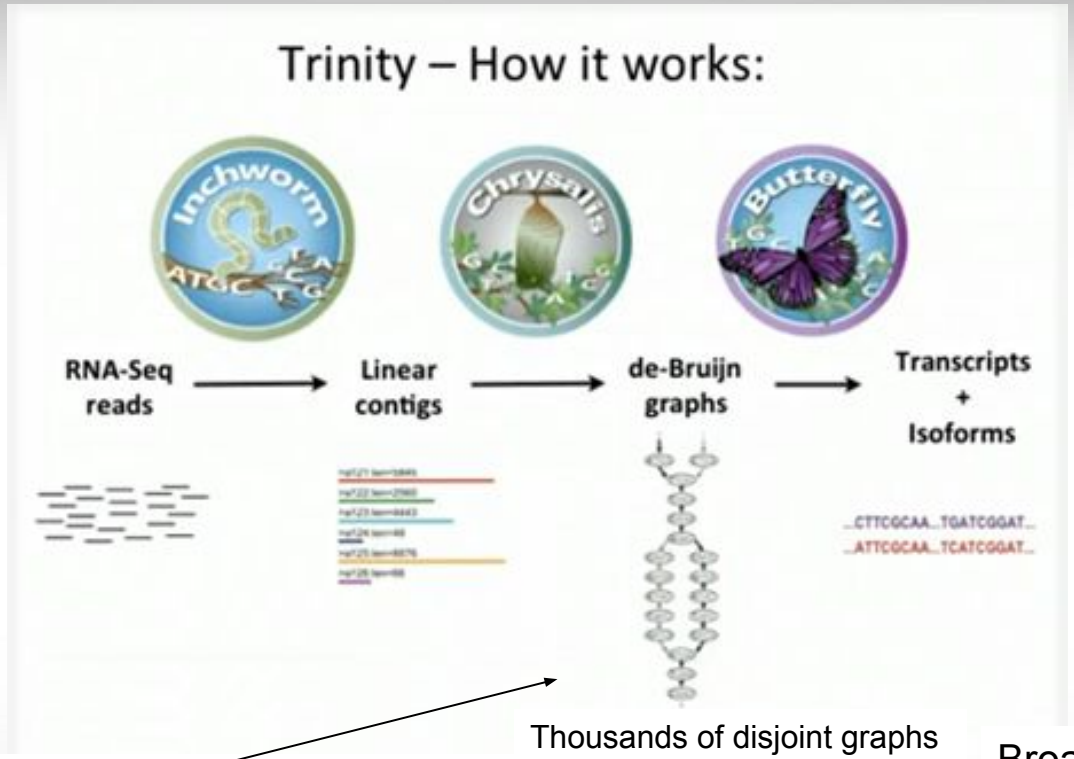
```
module spider Trinity
```

```
module spider Oases
```

Digital Normalization for Transcriptome Assembly

- Reduce memory requirements by reducing the number of redundant sequence reads if you have a very high sequencing coverage (> 200x)
- Trinity 2.4.0 automatically normalizes reads to a depth of 50
- The **`bbnorm.sh`** script in BBMap can normalize reads

```
module spider BBMap
```



http://www.rna-seqblog.com/a-collection-of-new-rna-seq-videos-from-the-broad-institute/

Running Trinity on Ada

- Trinity uses 100,000s of intermediate files
  - Contact **help@hprc.tamu.edu** and request a file quota increase before running Trinity
  - Run one Trinity job at a time and check resource usage
    - `showquota`
    - It is recommended not to run multiple Trinity jobs unless you know memory usage and an estimate of the number of temporary files
  - Trinity creates checkpoints and can be restarted if it stops due to file/disk quota met, out of memory or runtime
    - Checkpoints are not available when running Trinity in Galaxy
    - Checkpoints are not available if you use $TMPDIR with Trinity
      - need to rsync results from $TMPDIR at end of job script
      - checkpoints are stored in $TMPDIR which is deleted after job ends
- See GCATemplates for sample Trinity scripts

Analyzing hundreds or thousands of files

Multi-File Processing

If you have 100's or 1,000's of files to process with a tool that can only utilize one compute node, it could take days or weeks to process if you run one at a time.

```
spades.py --careful --memory 5 --threads 2 -o EC00144241_S85_build -1 EC00144241_S85_L001_R1_001.fastq.gz -2 EC00144241_S85_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC00502067_S15_build -1 EC00502067_S15_L001_R1_001.fastq.gz -2 EC00502067_S15_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC01236313_S68_build -1 EC01236313_S68_L001_R1_001.fastq.gz -2 EC01236313_S68_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC02035016_S50_build -1 EC02035016_S50_L001_R1_001.fastq.gz -2 EC02035016_S50_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC03218300_S17_build -1 EC03218300_S17_L001_R1_001.fastq.gz -2 EC03218300_S17_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC04615405_S96_build -1 EC04615405_S96_L001_R1_001.fastq.gz -2 EC04615405_S96_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC04856216_S88_build -1 EC04856216_S88_L001_R1_001.fastq.gz -2 EC04856216_S88_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC05148227_S53_build -1 EC05148227_S53_L001_R1_001.fastq.gz -2 EC05148227_S53_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC07674140_S20_build -1 EC07674140_S20_L001_R1_001.fastq.gz -2 EC07674140_S20_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC08067108_S58_build -1 EC08067108_S58_L001_R1_001.fastq.gz -2 EC08067108_S58_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC08328770_S90_build -1 EC08328770_S90_L001_R1_001.fastq.gz -2 EC08328770_S90_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC08834643_S65_build -1 EC08834643_S65_L001_R1_001.fastq.gz -2 EC08834643_S65_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC10485101_S57_build -1 EC10485101_S57_L001_R1_001.fastq.gz -2 EC10485101_S57_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC12123883_S45_build -1 EC12123883_S45_L001_R1_001.fastq.gz -2 EC12123883_S45_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC12334723_S72_build -1 EC12334723_S72_L001_R1_001.fastq.gz -2 EC12334723_S72_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC12344385_S95_build -1 EC12344385_S95_L001_R1_001.fastq.gz -2 EC12344385_S95_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC12600538_S14_build -1 EC12600538_S14_L001_R1_001.fastq.gz -2 EC12600538_S14_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC13365408_S44_build -1 EC13365408_S44_L001_R1_001.fastq.gz -2 EC13365408_S44_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC13880885_S84_build -1 EC13880885_S84_L001_R1_001.fastq.gz -2 EC13880885_S84_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC14371787_S81_build -1 EC14371787_S81_L001_R1_001.fastq.gz -2 EC14371787_S81_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC14418800_S13_build -1 EC14418800_S13_L001_R1_001.fastq.gz -2 EC14418800_S13_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC15011885_S10_build -1 EC15011885_S10_L001_R1_001.fastq.gz -2 EC15011885_S10_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
spades.py --careful --memory 5 --threads 2 -o EC15712676_S34_build -1 EC15712676_S34_L001_R1_001.fastq.gz -2 EC15712676_S34_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
```

Multi-File Processing

- The first step is to get an idea of the resource usage to process one sample.
  - Run one sample (preferable with the largest input file) to get estimated Walltime, max memory, number of cores
    - Utilize one entire node to get an accurate assessment
    - Make sure you are not almost at your file or disk quota
      - `showquota`
    - Use $TMPDIR for temporary files whenever possible
- For a tool that allows one or two-core processing on one node
  - Use TAMULauncher
- For a tool that allows multi-core utilization of the entire compute node
  - Use a job array

Example Dataset

- As an example, we will create a commands file for spades assemblies
  - 100 samples (E. coli)
  - Pair-1 and pair-2 sequence files for each sample (200 files total)
  - Small bacterial genomes that only need about 2 GB of memory
  - Will use 2 cores per command which equals 5GB of memory (Ada)
    - Can run 10 assemblies per node; total of 50GB memory usage
      - 1 core = 2.5 GB of memory for 1 hour charged as 1 SU
      - 10 commands per node = 50GB of total requested memory
    - spades.py --careful --memory 5 --threads 2 -o EC00144241_S85_build -1 EC00144241_S85_L001_R1_001.fastq.gz -2 EC00144241_S85_L001_R2_001.fastq.gz --tmp-dir $TMPDIR
  - Sequence files are located in directory
    - /scratch/training/intro_to_ngs/reads/

Preparing A Commands File

Use UNIX shell commands to loop through your files and generate a command for each file or file set.

- We want to capture the unique sample name such as: EC23875571_S8
- List files using a mix of wildcards and characters so that you only capture read pair 1 file names
- All pair-1 files end with _L001_R1_001.fastq.gz

```
ls /scratch/training/intro_to_ngs/reads/EC*_L001_R1*fastq.gz
```

Preparing A Commands File cont.

Use the command from the previous slide to create a list of files to loop through
Use shell variable substrings to capture the sample name from the filename

```
/scratch/training/intro_to_ngs/reads/EC85687002_S5_L001_R1_001.fastq.gz
```

```
for file in $(ls /scratch/training/intro_to_ngs/reads/EC*_L001_R1*fastq.gz)
do
name=${file##*/}
echo ${name%_L001*}
done
```

```
${var#*SubStr}   # will drop beginning of string up to first occurrence of 'SubStr'
${var##*SubStr}  # will drop beginning of string up to last occurrence of 'SubStr'
${var%SubStr*}   # will drop part of string from last occurrence of 'SubStr' to the end
${var%%SubStr*}  # will drop part of string from first occurrence of 'SubStr' to the end
```

Preparing A Commands File cont.

Save the output to a commands file appending each line using >>

```
data=/scratch/training/intro_to_ngs/reads
for file in $(ls $data/EC*_L001_R1*fastq.gz)
do
name=${file##*/}
echo spades.py --careful --memory 5 --threads 2 -o builds/${name%_L001*} \
-1 $data/${name%_L001*}_L001_R1_001.fastq.gz \
-2 $data/${name%_L001*}_L001_R2_001.fastq.gz --tmp-dir \$TMPDIR >> commands.txt
done
```

- Spades will automatically create the `builds/` directory
- Use the backslash in front of $TMPDIR so echo will print the dollar sign
- If your commands.txt file is incorrect, you need to delete it before creating it again

TAMULauncher Job Script

To run 50 assemblies concurrently; 10 per node on 5 nodes:

```
#BSUB -L /bin/bash              # use bash for environment
#BSUB -J tamulaunch_spades      # job name
#BSUB -n 100                    # assigns 100 total cores for execution
#BSUB -R "span[ptile=20]"       # assigns 20 cores per node
#BSUB -R "rusage[mem=2700]"     # reserves 2700MB memory per core
#BSUB -M 2700                   # sets to 2700MB per process enforceable memory
#BSUB -W 48:00                  # sets to 48 hour the job's runtime wall-clock limit
#BSUB -o stdout.%J              # directs the job's standard output to stdout.jobid
#BSUB -e stderr.%J              # directs the job's standard error to stderr.jobid

module load SPAdes/3.11.1-GCCcore-6.3.0

# run 10 commands per node since each command uses 2 threads and Ada = 20 cores/node
tamulauncher --commands-pernode 10 commands.txt
```

This job takes about 2hr 30min with a final charge of about 255 SUs

Job Array

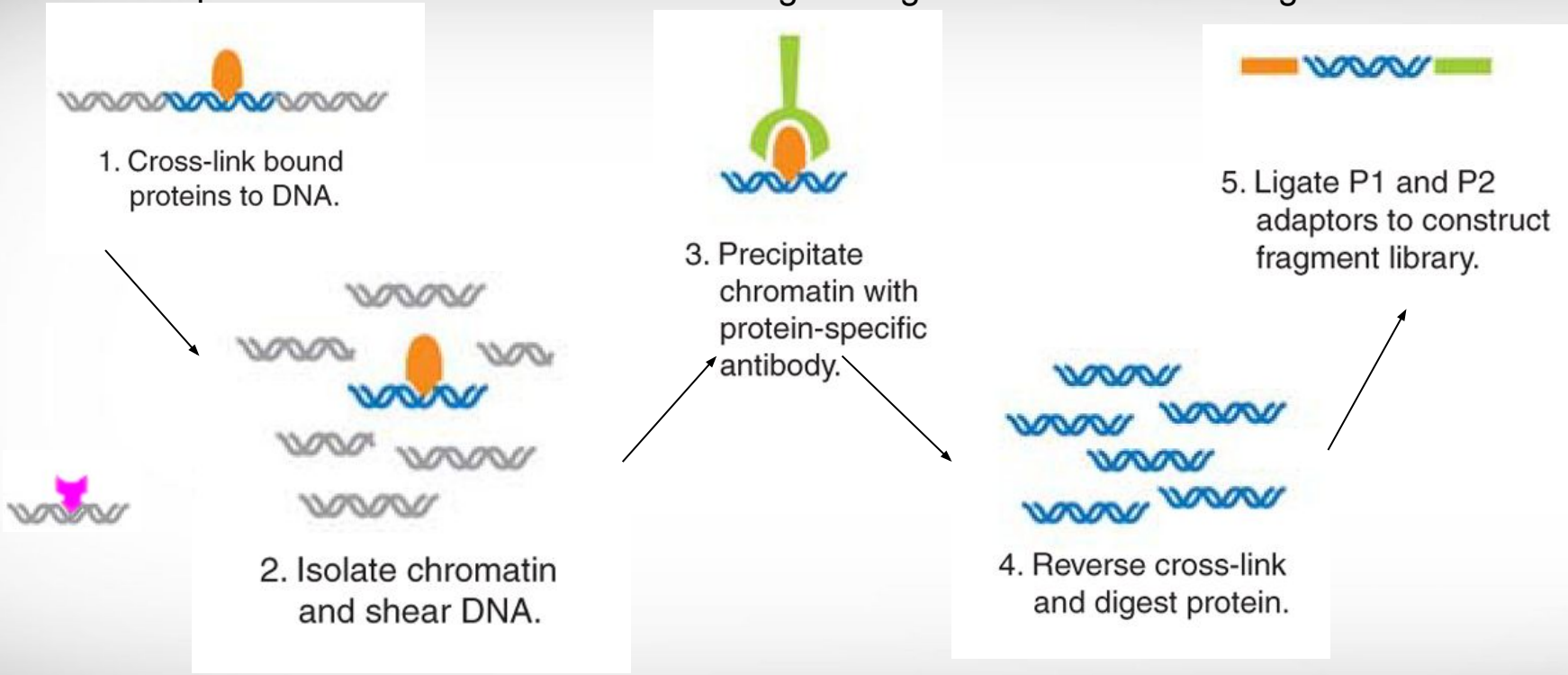
For jobs where you need to use the entire node just for one assembly.

- You can use up to 50 nodes at a time (1000 cores) on Ada but this example only uses 10.
- You can use the same commands file just update the memory & threads for each command
  - --memory 50 --threads 20
- You can access one line of your commands.txt per array index using **`$LSB_JOBINDEX`**
- The job script represents one assembly, i.e. one spades assembly run per node

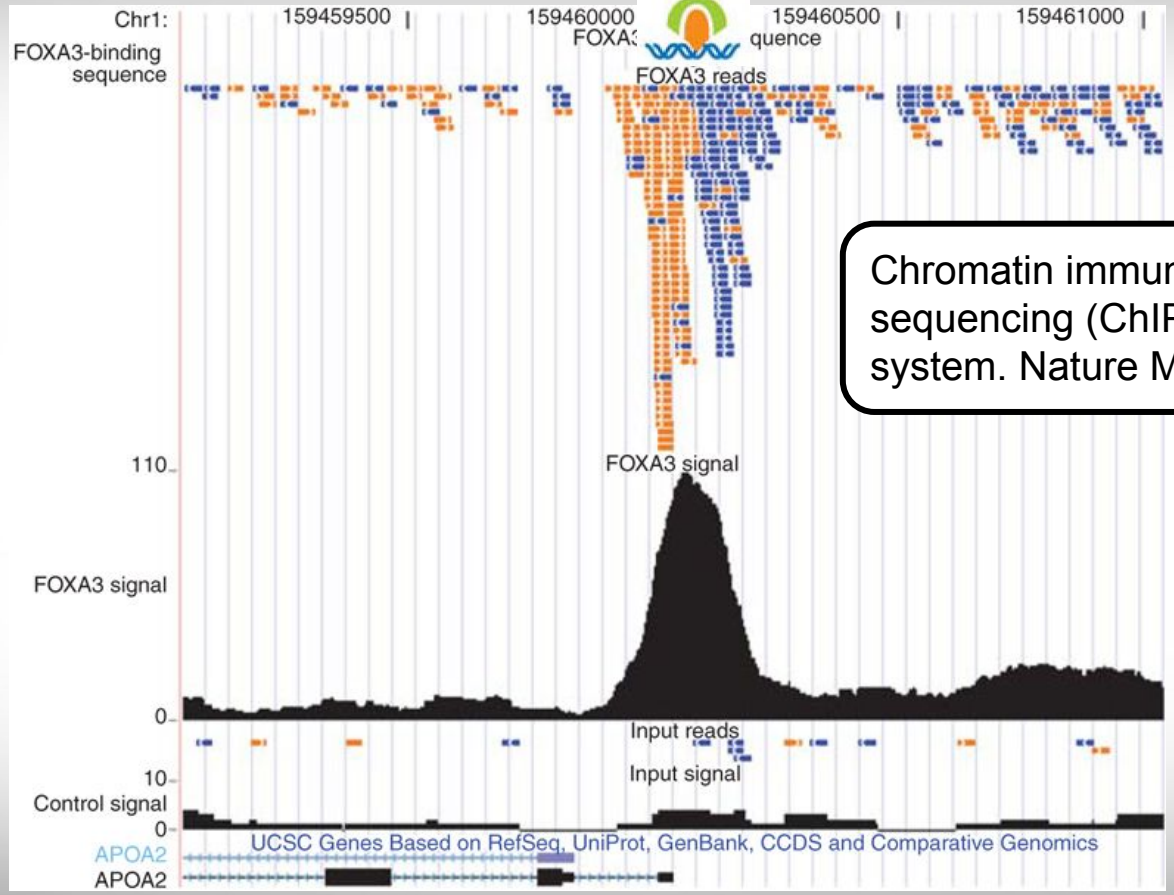
```
#BSUB -L /bin/bash                    # use bash for environment
#BSUB -J spades_array[1-100]%10       # run an array of 100 jobs but only 10 concurrently
#BSUB -n 20                           # assigns 20 total cores for execution
#BSUB -R "span[ptile=20]"             # assigns 20 cores per node
#BSUB -R "rusage[mem=2700]"           # reserves 2700MB memory per core
#BSUB -M 2700                         # sets to 2700MB per process enforceable memory
#BSUB -W 24:00                        # sets to 48 hour the job's runtime wall-clock limit
#BSUB -o stdout.%J.%I                 # job's standard output to stdout.jobid.index
#BSUB -e stderr.%J.%I                 # job's standard error to stderr.jobid.index

module load SPAdes/3.11.1-GCCcore-6.3.0

command=`sed -n ${LSB_JOBINDEX}p commands.txt`
$command
```

ChIP-seq

Chromatin immunoprecipitation (ChIP) is a technique for identifying and characterizing elements in protein-DNA interactions involved in gene regulation or chromatin organization.

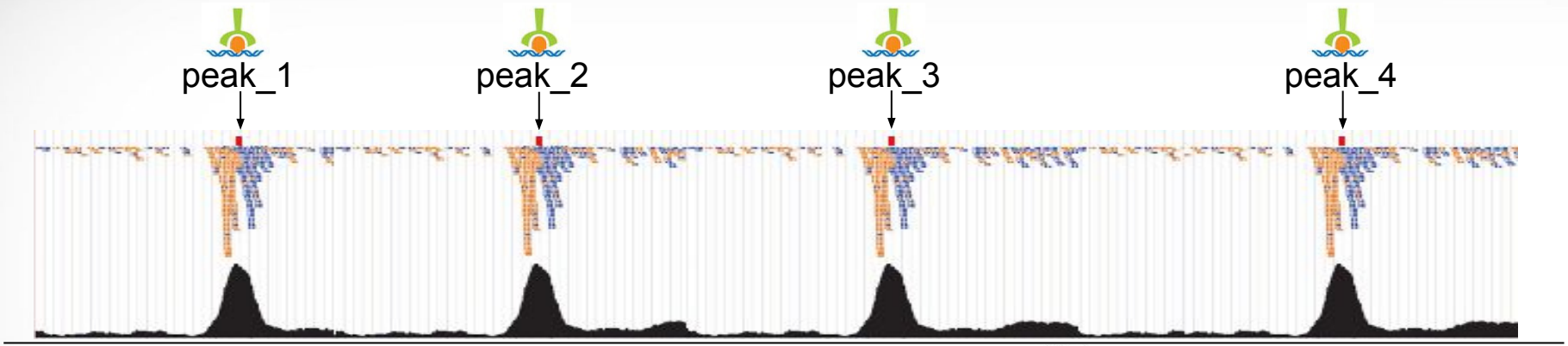


Chromatin immunoprecipitation sequencing (ChIP-Seq) on the SOLiD™ system
Nature Methods 6, (2009)



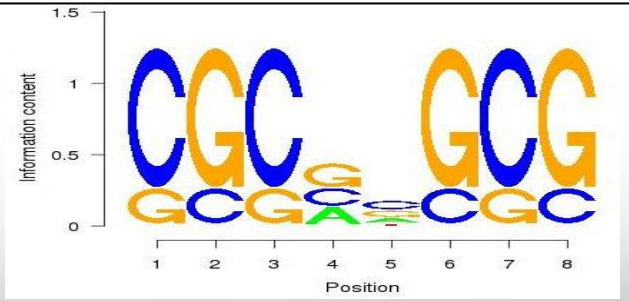
Chromatin immunoprecipitation sequencing (ChIP-Seq) on the SOLiD™ system. Nature Methods 6, (2009)

The goal is to find a consensus DNA sequence among the sequences at each peak which will give us the DNA sequence motif that a protein recognizes and binds



A sequence logo can be used to represent the DNA sequence motif where the protein binds

Generate a sequence logo with the R package seqLogo




```
module load R_tamu/3.3.1-intel-2015B-default-mt
```

ChIP-seq Tools

- Protein-DNA interactions
  - `module spider MACS`
  - `module spider MACS2`
- Subdivision of ChIP-seq regions into discrete signal peaks
  - `module spider PeakSplitter`
- Peak caller
  - `module spider PeakRanger`
  - `module spider BroadPeak`
- Identify enriched domains from histone modification ChIP-seq data
  - `module spider SICER`

HPRC Resources

- Free Help
  - Send an email to help@hprc.tamu.edu if you have any questions regarding Bioinformatics tools usage on HPRC clusters
    - First spend some time investigating the error
      - read log files, stdout file, stderr file, tool manual
      - Google search
      - Google user groups: many are tool specific
    - Include details about your issue
      - Which cluster or which Galaxy you are using
      - Which tool you are using
      - Which modules you have loaded
      - Commands you used in your job script
      - Error messages you are seeing
- HPRC NGS data analysis tools Documentation
  - https://hprc.tamu.edu/wiki/index.php/Ada:Bioinformatics

Thank you.

Any question?