WESTGRID

compute canada
regional partner

UBC

UBC100

Advanced Research Computing

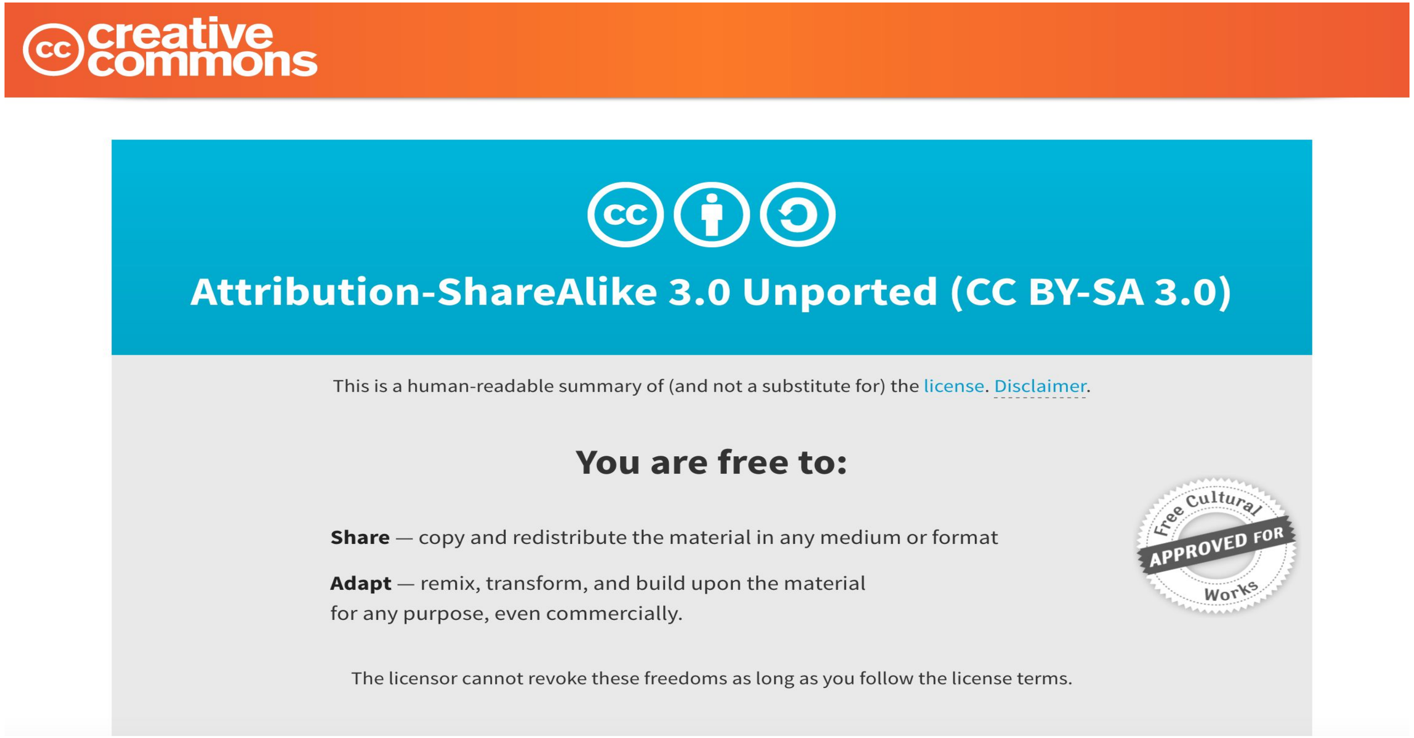
Education Outreach and Training Tutorials

Introduction to Short Read Mapping: The foundation of next generation sequencing analysis

April 3rd, 2019 (10:00AM-11:00PM PST)

Phillip A Richmond, Oriol Fornes

Copyright Information



creative commons

Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material
for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

The material is open source, and in this presentation no previous external work was utilized.

Welcome!

- Welcome to the Introduction to Short Read Mapping
- I am co-teaching this seminar with Dr. Oriol Fornes, who studies gene regulation and frequently processes short-read data on the Cedar compute cluster.
- This is not meant to be a follow-along seminar, but the commands, datasets, and scripts will be available afterwards for your own exploration
- This presentation will be recorded and the slides will remain available

http://bit.ly/2WD1ORc

WESTGRID

UBC UBC100

Advanced Research Computing

Interactive Experience

We hope this is an interactive experience for all of you.

Questions/Problems can be posted to the Etherpad:

https://etherpad.openstack.org/p/EOT_APRIL2019

Dr. Oriol Fornes will be here to help answer questions while I’m presenting.

WESTGRID

UBC

UBC100

Advanced Research Computing

Speaker Bio

Phillip Richmond

PhD Candidate, Wasserman Lab, BC Children's Hospital Research Institute

Bioinformatics Program, University of British Columbia

https://phillip-a-richmond.github.io

Research:Maximizing the Utility of Whole Genome Sequencing in the Diagnosis of Rare Genetic Disorders

Previous work in Genomics: Genomic Contributions to Ethanol Sensitivity in Mice, Polyploid Evolution in Yeast, Brewing Yeast Genomics, Cancer Cell Epigenetics, Addiction Predisposition

Also loves teaching genomics, and my puppy Sherlock Holmes

WESTGRID

UBC UBC100

Advanced Research Computing

Session Outline

- Introduction to next generation sequencing data & diverse data types
- Transcriptional regulatory datasets and where to find them
- Mapping reads to the genome using BWA mem
- Peak calling and creating pileup files using MACS2
- Visualizing data in IGV

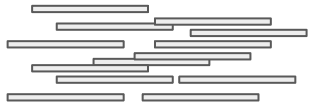
Advanced Research Computing

Session Outline

- Introduction to next generation sequencing data & diverse data types
- Transcriptional regulatory datasets and where to find them
- Mapping reads to the genome using BWA mem
- Peak calling and creating pileup files using MACS2
- Visualizing data in IGV

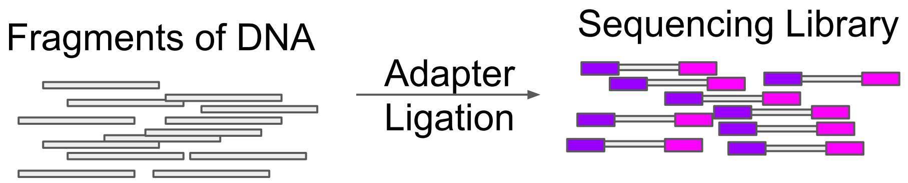
Next generation sequencing: Short-read sequencing

Fragments of DNA

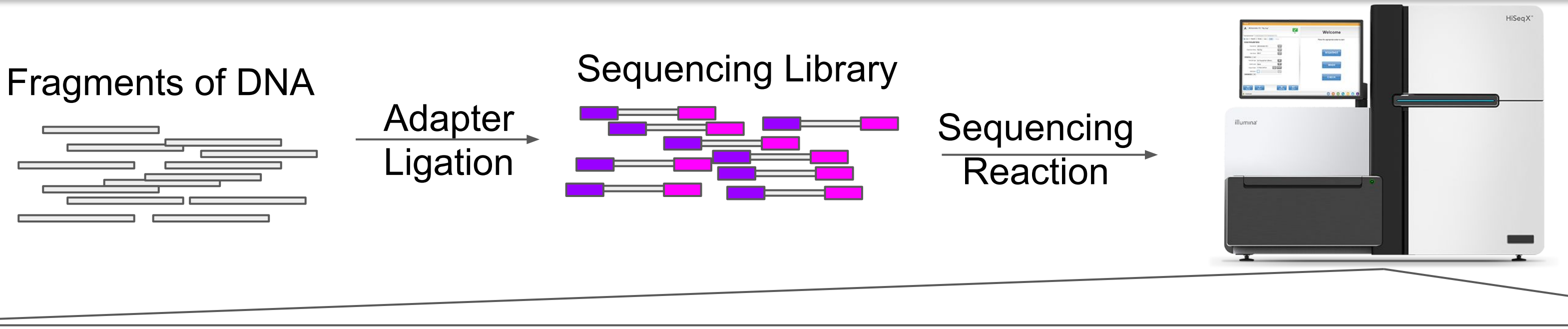


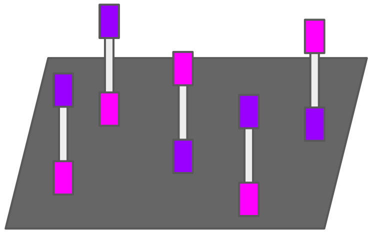
Advanced Research Computing

Next generation sequencing: Short-read sequencing



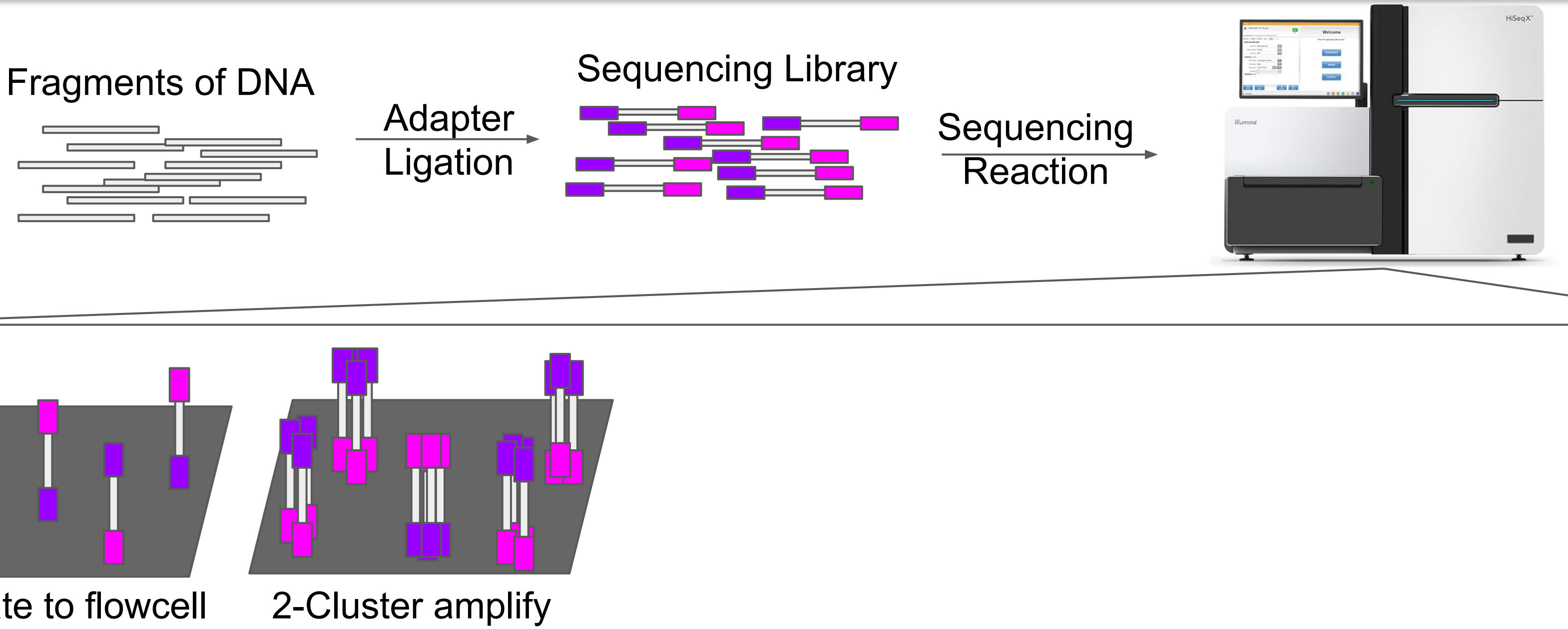
Next generation sequencing: Short-read sequencing





1-Ligate to flowcell

Next generation sequencing: Short-read sequencing

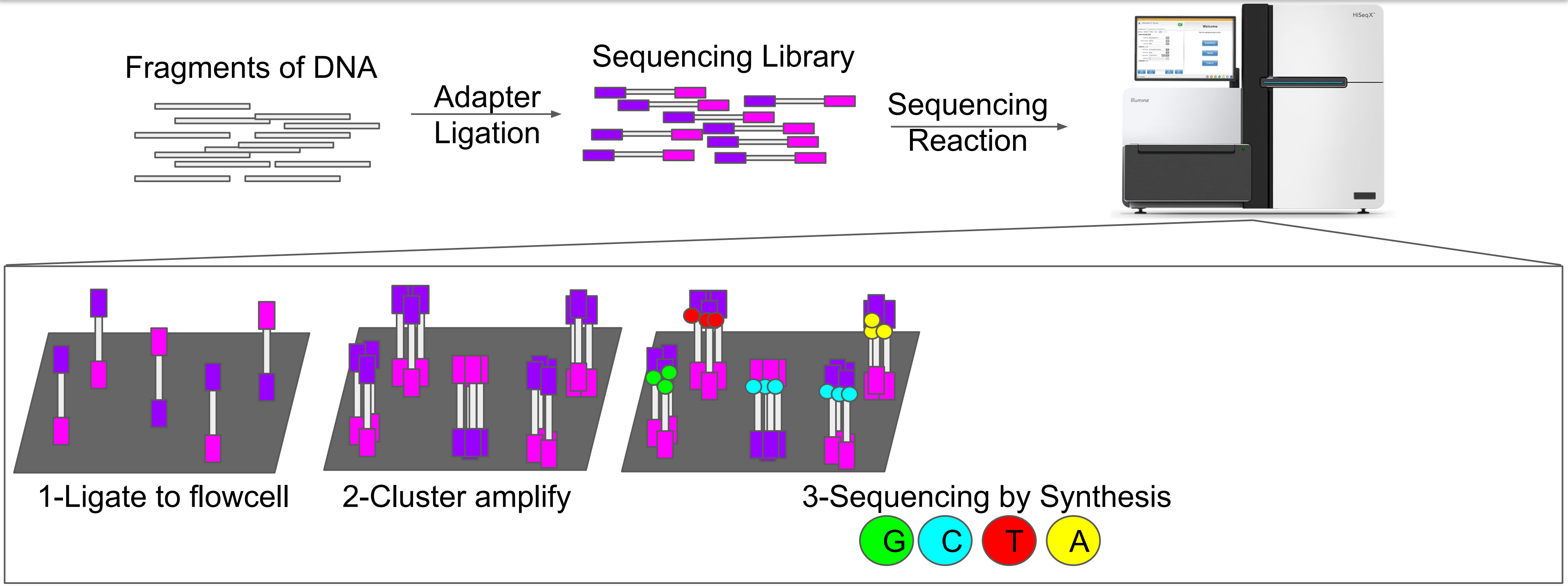


WESTGRID

UBC UBC100

Advanced Research Computing

Next generation sequencing: Short-read sequencing

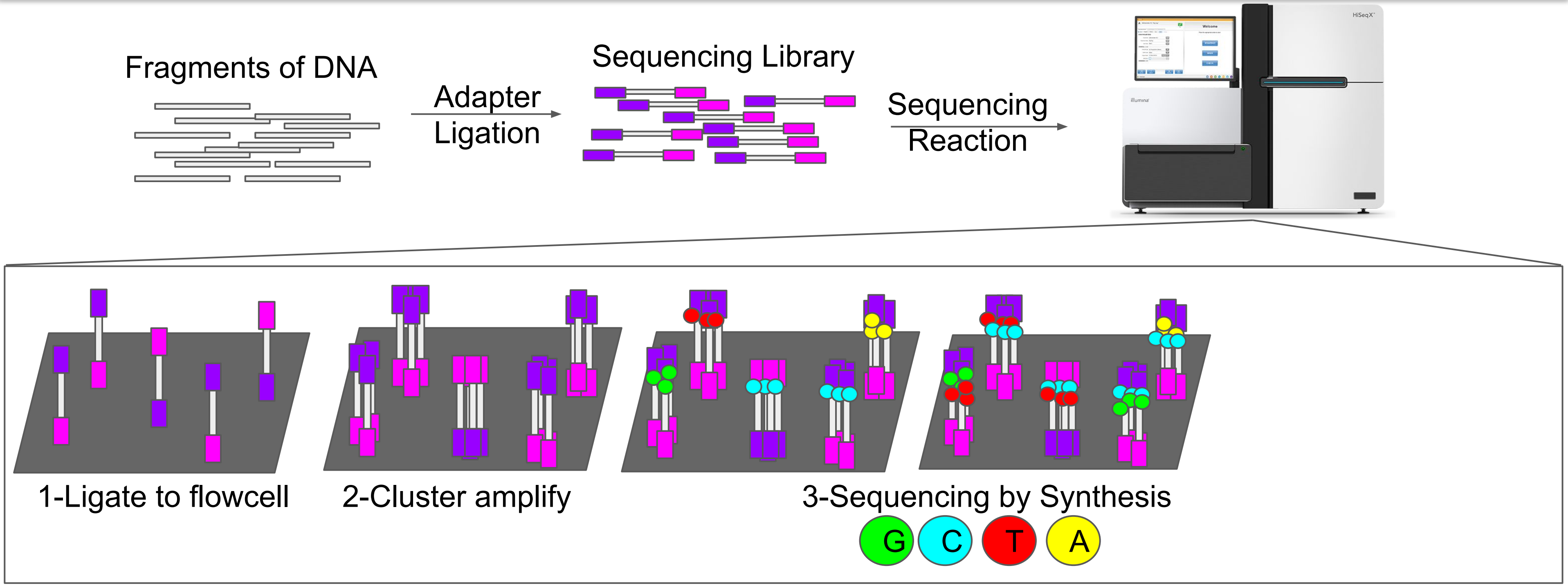


WESTGRID

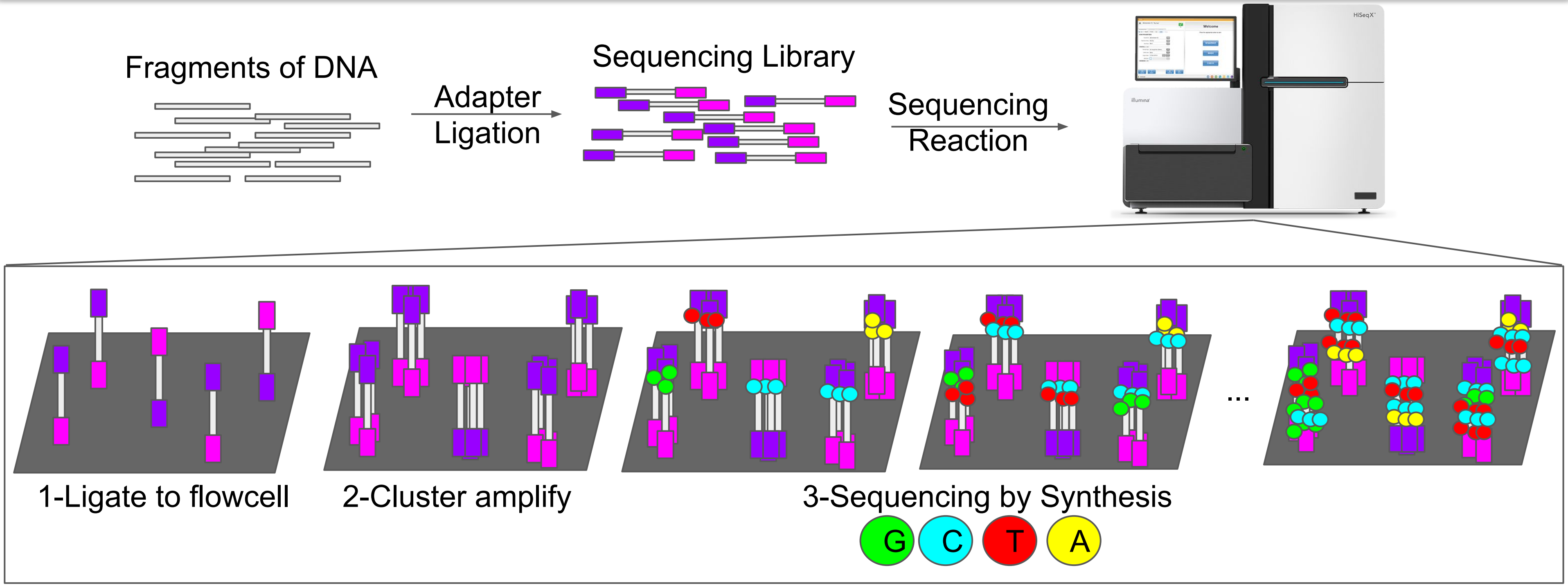
UBC UBC100

Advanced Research Computing

Next generation sequencing: Short-read sequencing



Next generation sequencing: Short-read sequencing



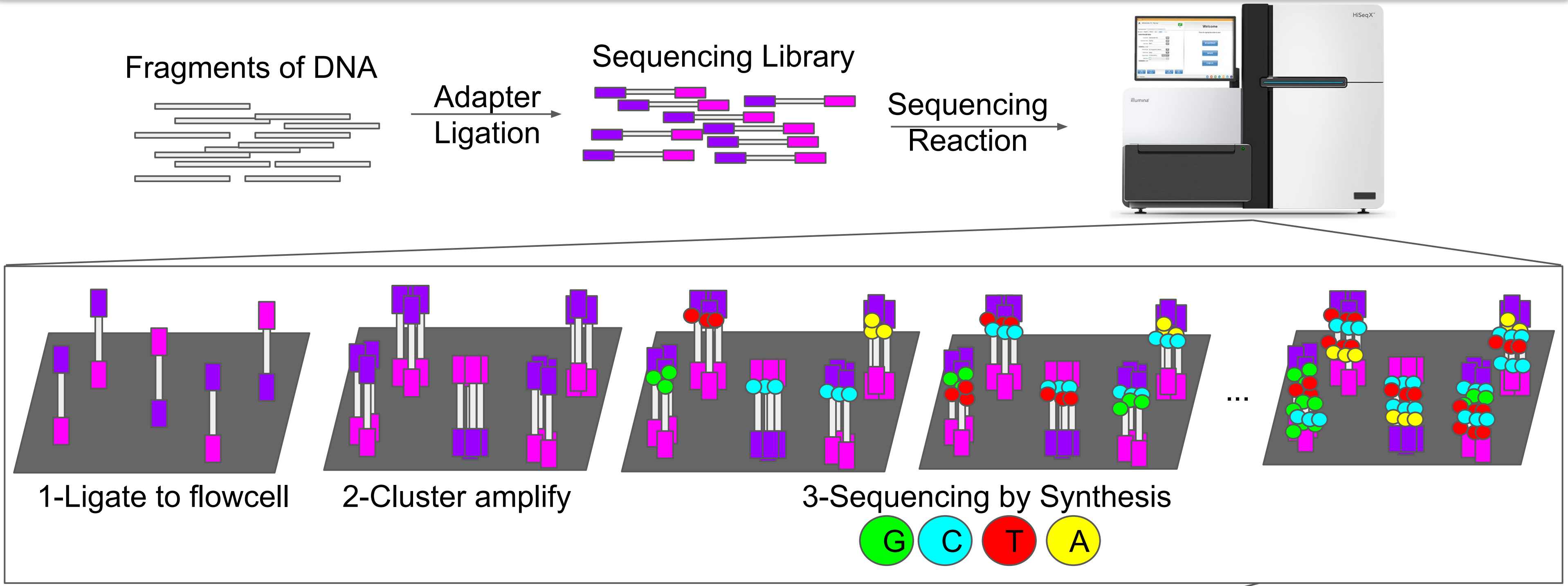
WESTGRID

UBC

UBC100

Advanced Research Computing

Next generation sequencing: Short-read sequencing



Convert to Fastq

```
@Read1
TCTTGCGTACGTCTTCGATCGTA
+
!!@$@##@!%!@#$!!LLBBDKSNK
```

Advanced Research Computing

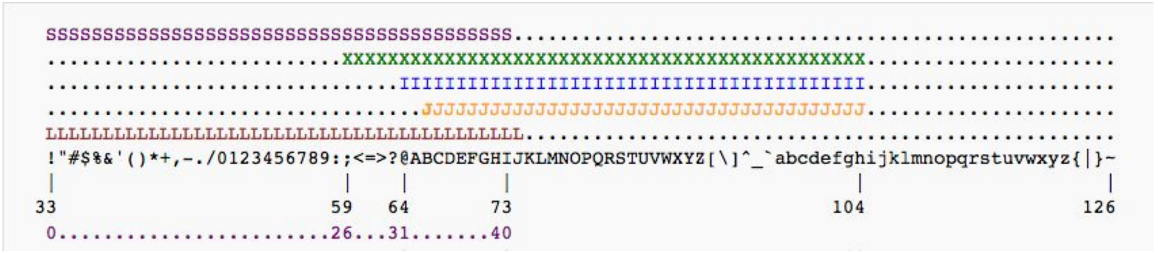
Diverse Input, Same Output Format

- Different inputs still result in the same output data format
- Examples:
    - DNA-seq, ChIP-seq, RNA-seq, GRO-seq, ATAC-seq
- For non-DNA assays (e.g. RNA-seq/GRO-seq), they undergo a conversion from RNA-->cDNA before sequencing

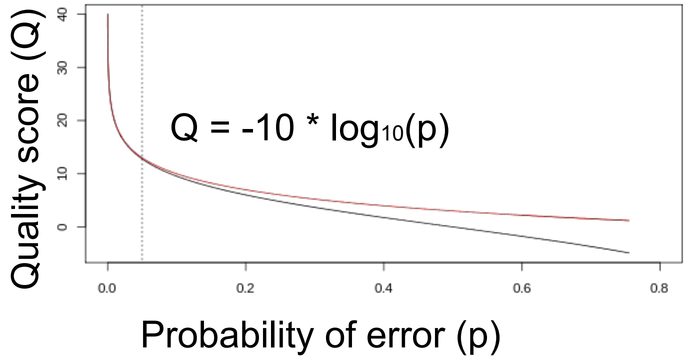
EXAMPLE	MEANING
@K00171:617:HMMTNBBXX:1:1101:28686:1648 1:N:0:GACTAGTA	@Readname:And:Flowcell:Info 1 or 2 for read pair:N:0:Barcode
TCTTGCGTACGTCTTCGATCGTA	Sequence
+	“Plus Sign”
!!@$@##@!%!@#$!!LLBBDKSNK	ASCII-Quality Scores

Diverse Input Data, Same Output Format

EXAMPLE	MEANING
@K00171:617:HMMTNBBXX:1:1101:28686:1648 1:N:0:GACTAGTA	@Readname:And:Flowcell:Info 1 or 2 for read pair:N:0:Barcode
TCTTGCGTACGTCTTCGATCGTA	Sequence
+	“Plus Sign”
BBBBCCA?>><>=;:BBBBBBBBB	**ASCII-Quality Scores**



```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
.................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL....................................................
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
|                         |    |        |                              |                     |
33                        59   64       73                            104                   126
0........................26...31.......40
```



Session Outline

- Introduction to next generation sequencing data & diverse data types
- Transcriptional regulatory datasets and where to find them
- Mapping reads to the genome using BWA mem
- Peak calling and creating pileup files using MACS2
- Visualizing data in IGV

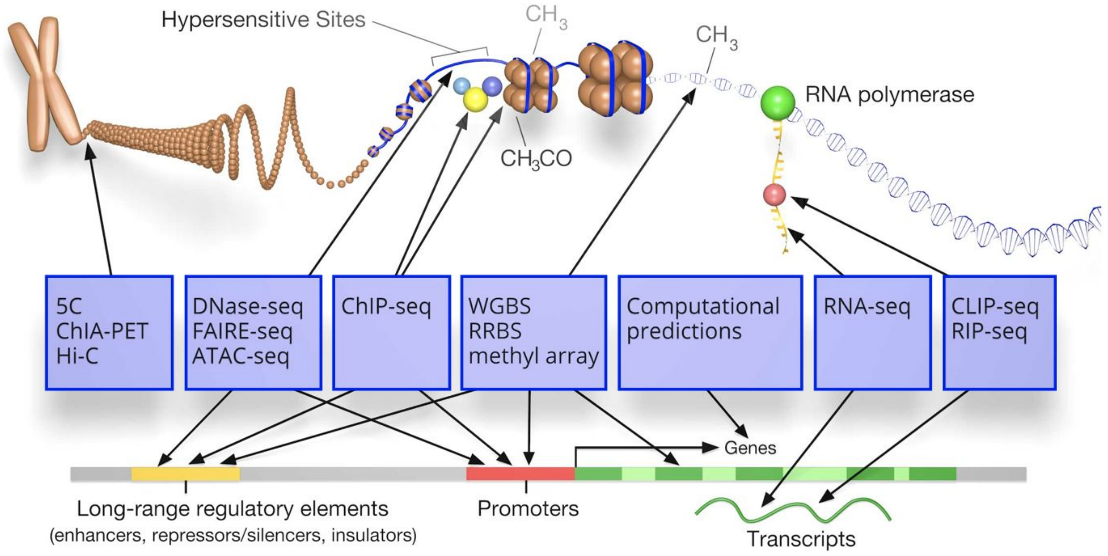
WESTGRID

UBC UBC100

Advanced Research Computing

ENCODE - Encyclopedia of DNA Elements

ENCODE is one of the many places to find open source data: www.encodeproject.org

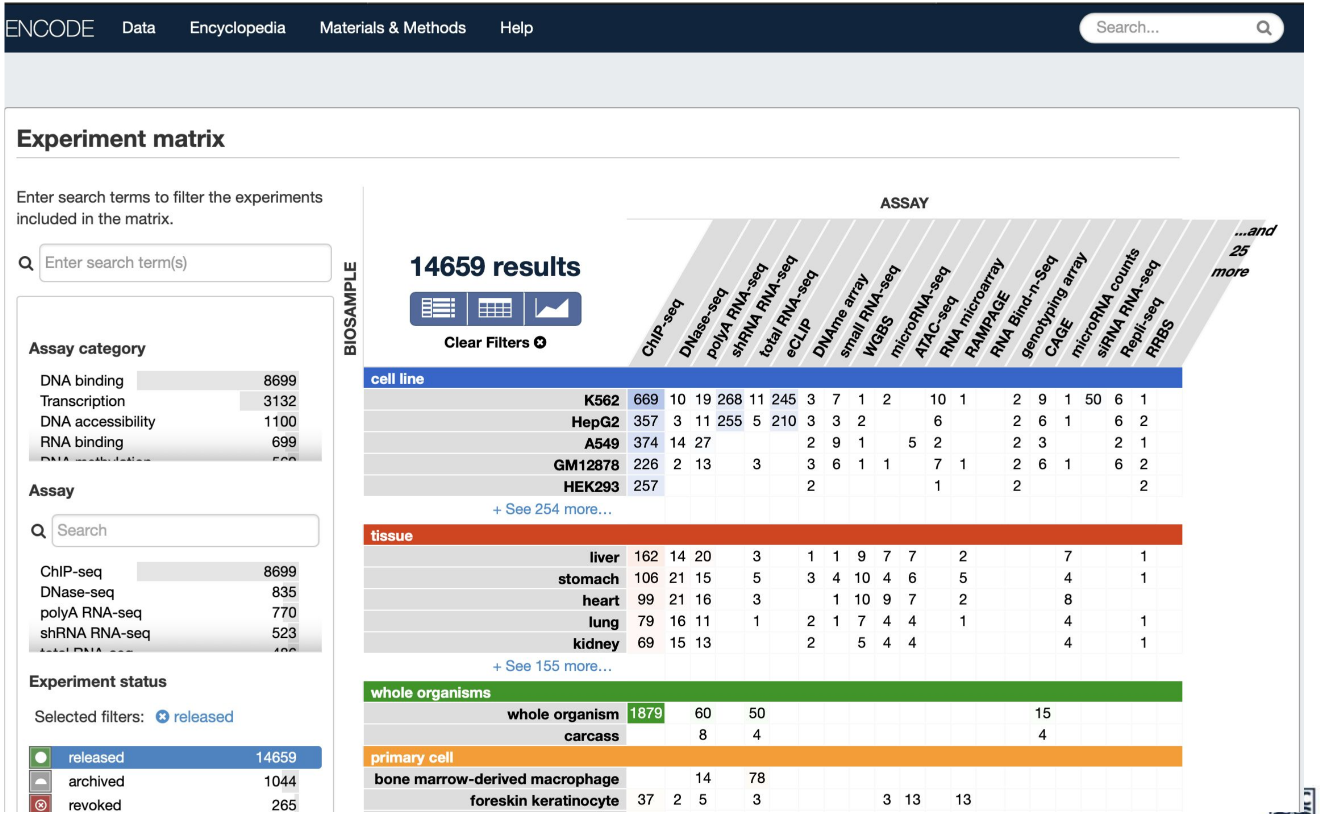


Based on an image by Darryl Leja (NHGRI), Ian Dunham (EBI), Michael Pazin (NHGRI)

encodeproject.org

Advanced Research Computing

You can download a diverse set of data across tissues/cell types



ENCODE Data Encyclopedia Materials & Methods Help Search...

Experiment matrix

Enter search terms to filter the experiments included in the matrix.

Enter search term(s)

Assay category

DNA binding	8699
Transcription	3132
DNA accessibility	1100
RNA binding	699

Assay

Search

ChIP-seq	8699
DNase-seq	835
polyA RNA-seq	770
shRNA RNA-seq	523

Experiment status

Selected filters: ⊗ released

●	released	14659
	archived	1044
⊗	revoked	265

14659 results

Clear Filters ⊗

BIOSAMPLE / ASSAY

	ChIP-seq	DNase-seq	polyA RNA-seq	shRNA RNA-seq	total RNA-seq	eCLIP	DNAme array	small RNA-seq	WGBS	microRNA-seq	ATAC-seq	RNA microarray	RAMPAGE	RNA Bind-n-Seq	genotyping array	CAGE	microRNA counts	siRNA RNA-seq	Repli-seq	RRBS	...and 25 more
cell line																					
K562	669	10	19	268	11	245	3	7	1	2		10	1		2	9	1	50	6	1	
HepG2	357	3	11	255	5	210	3	3	2			6			2	6	1		6	2	
A549	374	14	27				2	9	1		5	2			2	3			2	1	
GM12878	226	2	13		3		3	6	1	1		7	1		2	6	1		6	2	
HEK293	257						2					1			2					2	
+ See 254 more...																					
tissue																					
liver	162	14	20		3		1	1	9	7	7		2				7			1	
stomach	106	21	15		5		3	4	10	4	6		5				4			1	
heart	99	21	16		3			1	10	9	7		2				8				
lung	79	16	11		1		2	1	7	4	4		1				4			1	
kidney	69	15	13				2		5	4	4						4			1	
+ See 155 more...																					
whole organisms																					
whole organism	1879		60		50											15					
carcass			8		4											4					
primary cell																					
bone marrow-derived macrophage			14		78																
foreskin keratinocyte	37	2	5		3					3	13		13								

WESTGRID

UBC100

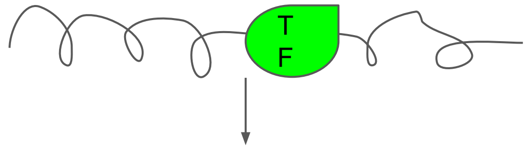
Advanced Research Computing

Chromatin Immunoprecipitation (ChIP-seq)

Chromatin Immunoprecipitation Sequencing (ChIP-seq) protocol:

Purpose: To find which sequences of DNA a specific protein interacts with, i.e Transcription Factor (TF).

1-Crosslink
DNA:Protein



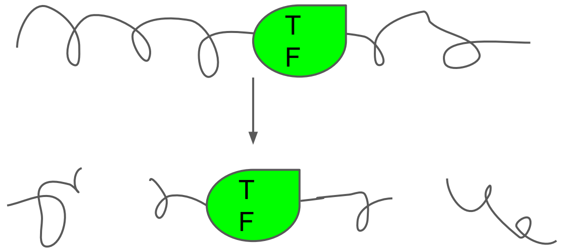
Chromatin Immunoprecipitation (ChIP-seq)

Chromatin Immunoprecipitation Sequencing (ChIP-seq) protocol:

Purpose: To find which sequences of DNA a specific protein interacts with, i.e Transcription Factor (TF).

1-Crosslink DNA:Protein

2-Shear



Chromatin Immunoprecipitation (ChIP-seq)

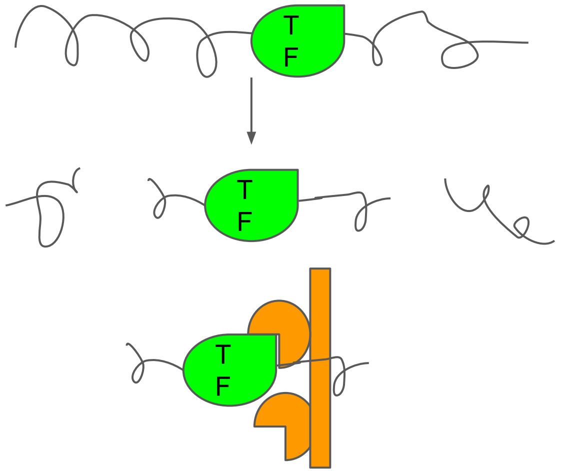
Chromatin Immunoprecipitation Sequencing (ChIP-seq) protocol:

Purpose: To find which sequences of DNA a specific protein interacts with, i.e Transcription Factor (TF).

1-Crosslink DNA:Protein

2-Shear

3-Pull Down protein using anti-protein antibody on a column, wash away other DNA



Advanced Research Computing

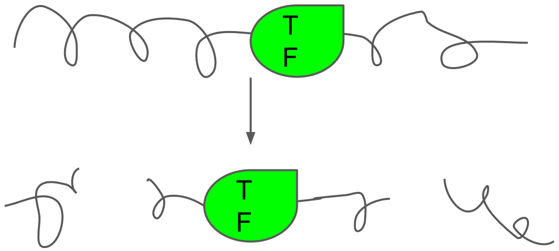
Chromatin Immunoprecipitation (ChIP-seq)

Chromatin Immunoprecipitation Sequencing (ChIP-seq) protocol:

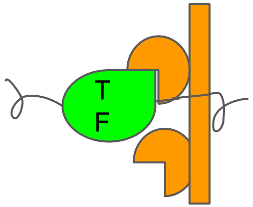
Purpose: To find which sequences of DNA a specific protein interacts with, i.e Transcription Factor (TF).

1-Crosslink DNA:Protein

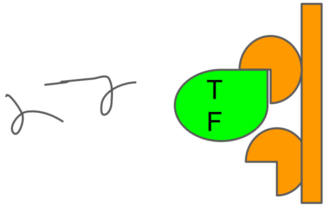
2-Shear



3-Pull Down protein using anti-protein antibody on a column, wash away other DNA



4-Reverse Crosslink



Chromatin Immunoprecipitation (ChIP-seq)

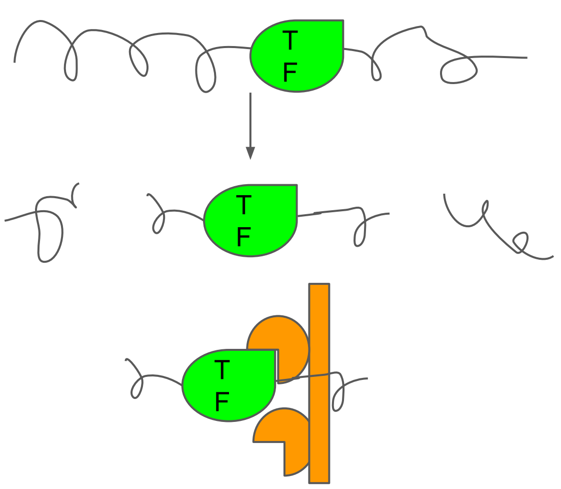
Chromatin Immunoprecipitation Sequencing (ChIP-seq) protocol:

Purpose: To find which sequences of DNA a specific protein interacts with, i.e Transcription Factor (TF).

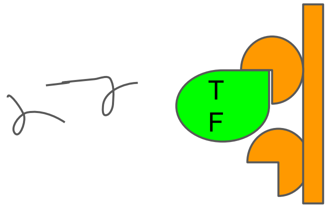
1-Crosslink DNA:Protein

2-Shear

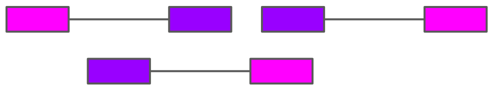
3-Pull Down protein using anti-protein antibody on a column, wash away other DNA



4-Reverse Crosslink



5-Ligate sequencing adapters

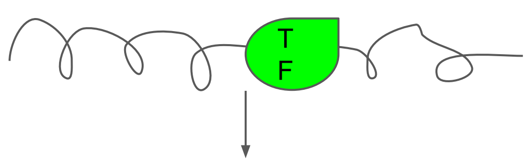


Chromatin Immunoprecipitation (ChIP-seq)

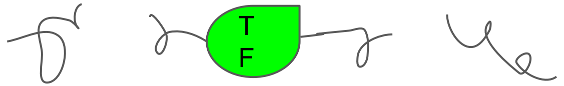
Chromatin Immunoprecipitation Sequencing (ChIP-seq) protocol:

Purpose: To find which sequences of DNA a specific protein interacts with, i.e Transcription Factor (TF).

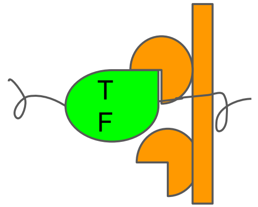
1-Crosslink DNA:Protein



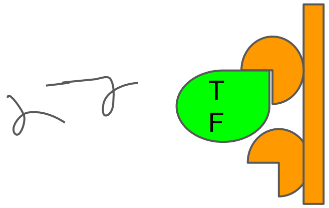
2-Shear



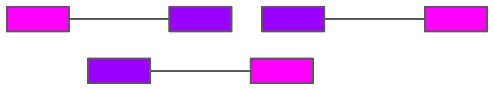
3-Pull Down protein using anti-protein antibody on a column, wash away other DNA



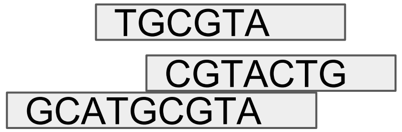
4-Reverse Crosslink



5-Ligate sequencing adapters



6-Sequence Library

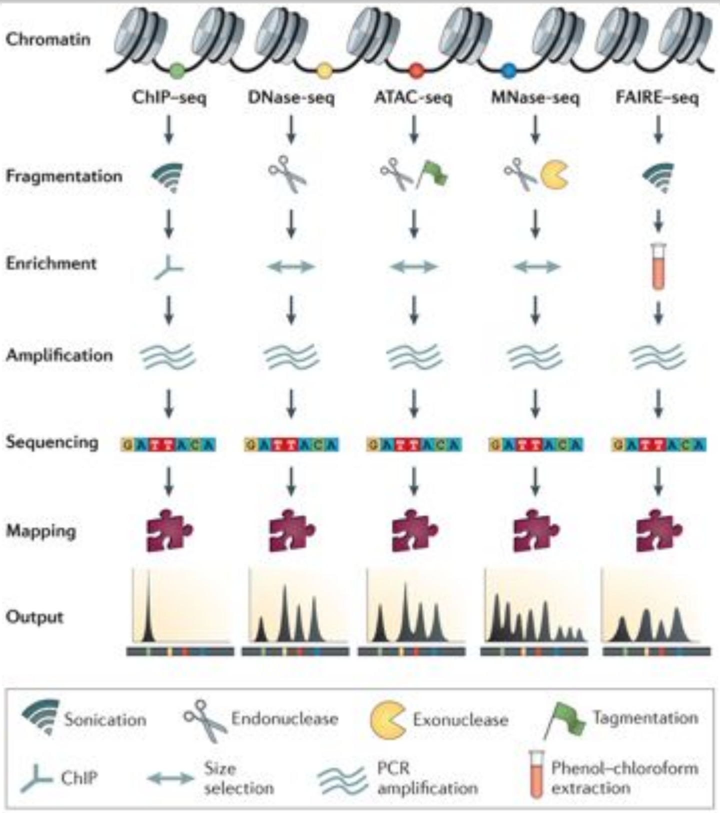


WESTGRID

UBC UBC100

Advanced Research Computing

ATAC-seq represents open chromatin



Nature Reviews | Genetics

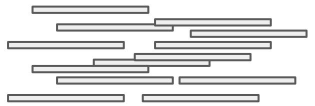
WESTGRID

UBC UBC100 Advanced Research Computing

Mapping data to a reference: ChIP-seq Peak Calling

- Individually, the short sequencing reads do not have much information
- Collectively, they can represent something useful
- Analyzing short-read data takes two common forms:
  - **Reference-based mapping**
  - Assembly

Example: ChIP-seq for a Transcription Factor

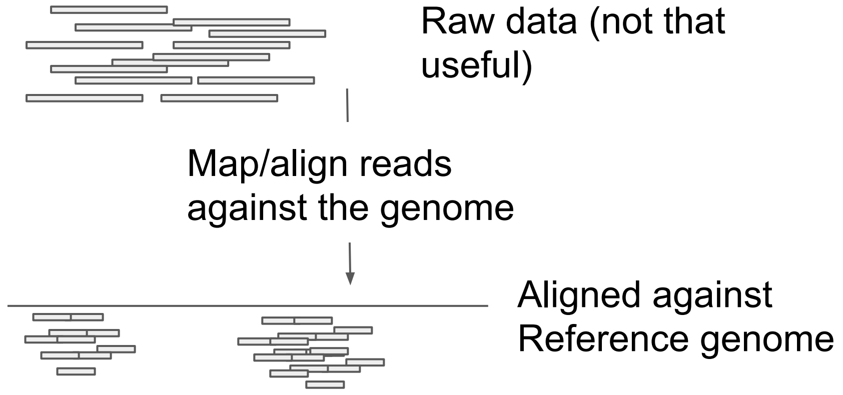


Raw data (not that useful)

Mapping data to a reference: ChIP-seq Peak Calling

- Individually, the short sequencing reads do not have much information
- Collectively, they can represent something useful
- Analyzing short-read data takes two common forms:
    - **Reference-based mapping**
    - Assembly

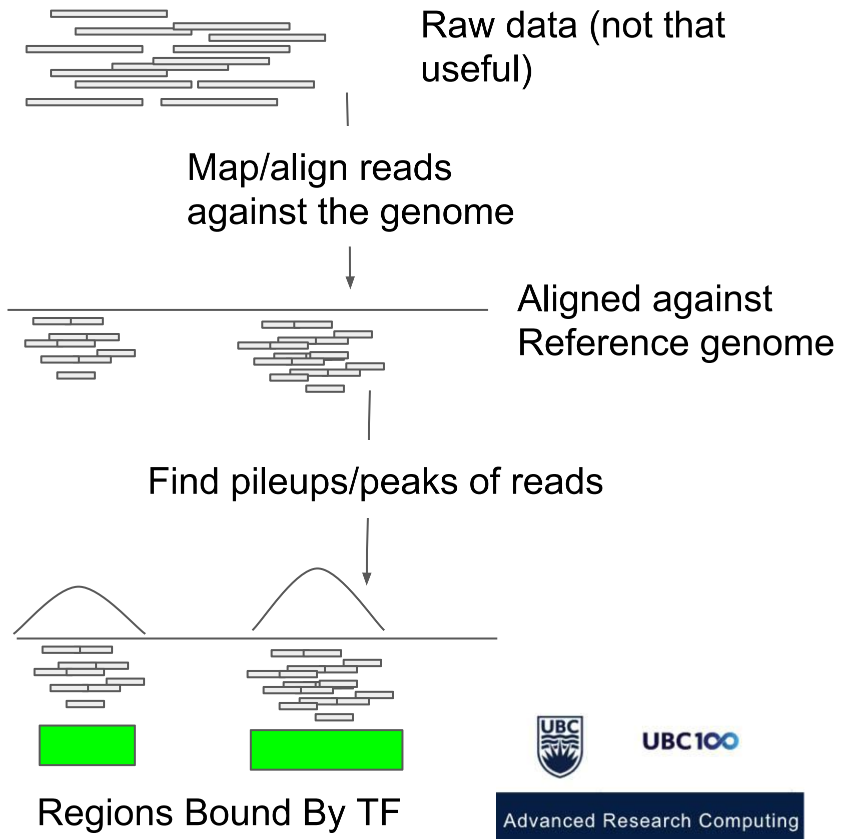
Example: ChIP-seq for a Transcription Factor

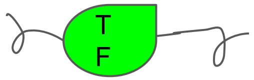

Raw data (not that useful)
Map/align reads against the genome
Aligned against Reference genome

Mapping data to a reference: ChIP-seq Peak Calling

- Individually, the short sequencing reads do not have much information
- Collectively, they can represent something useful
- Analyzing short-read data takes two common forms:
  - **Reference-based mapping**
  - Assembly

Example: ChIP-seq for a Transcription Factor


Raw data (not that useful)
Map/align reads against the genome
Aligned against Reference genome
Find pileups/peaks of reads
Regions Bound By TF



Session Outline

- Introduction to next generation sequencing data & diverse data types
- Transcriptional regulatory datasets and where to find them
- Mapping reads to the genome using BWA mem
- Peak calling and creating pileup files using MACS2
- Visualizing data in IGV

Advanced Research Computing

For those of you who want to “follow along”

I’ll be showing you files and pipeline scripts which are available to you to reuse/repurpose as you see fit.

NOTE: I do not expect you to follow along on the command line exploring the files. You can always re-watch this recording and hit pause. Maybe listen to it in slow motion. Whatever floats your boat.

Logging into cedar:

```
$ ssh username@cedar.computecanada.ca
```

A place of learning

The main directory for today's workshop data/scripts is:

/scratch/richmonp/TRAINING/APRIL2019/

If you want, you can make a temporary directory here to play around with. If you do, name it something unique.

```
$ mkdir /scratch/richmonp/TRAINING/APRIL2019/SHERLOCK/
```

Change SHERLOCK to your own directory name if you want to rerun this script.

All you need is scripts

/scratch/richmonp/TRAINING/APRIL2019/SCRIPTS/ has 3 scripts inside it:

H3K27Ac_Workshop.sh

POLR2A_Workshop.sh

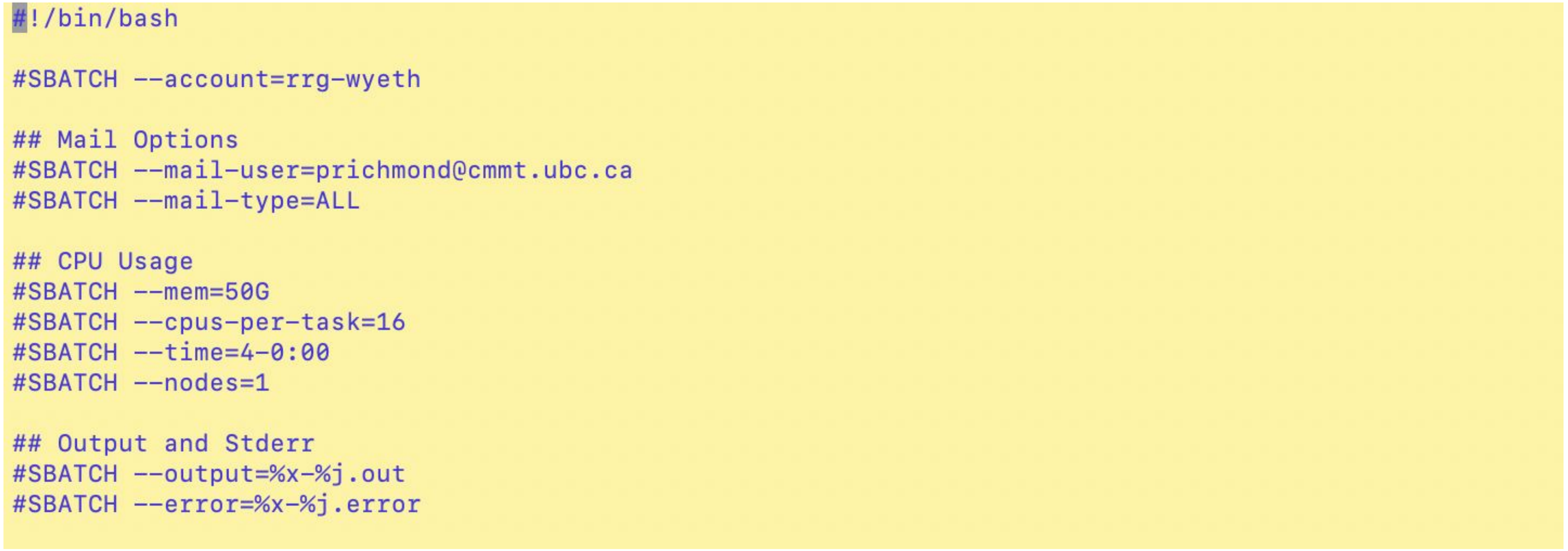
ATAC-Seq_Workshop.sh

I'm going to copy these so I can play with them:

```
$ cp /scratch/richmonp/TRAINING/APRIL2019/SCRIPTS/*sh    SHERLOCK/
```

There are also scripts in this directory without the _Workshop, they are the ones I've already edited to work for my personal directory.

Breakdown of the script: Welcome to the mellow yellow



```
#!/bin/bash

#SBATCH --account=rrg-wyeth

## Mail Options
#SBATCH --mail-user=prichmond@cmmt.ubc.ca
#SBATCH --mail-type=ALL

## CPU Usage
#SBATCH --mem=50G
#SBATCH --cpus-per-task=16
#SBATCH --time=4-0:00
#SBATCH --nodes=1

## Output and Stderr
#SBATCH --output=%x-%j.out
#SBATCH --error=%x-%j.error
```

This header information contains info about the account to bill for these hours, I want it to mail me, how much RAM and CPUs I need over a single node, and where to send standard error and output

WESTGRID

UBC UBC100

Advanced Research Computing

Breakdown of the script: Welcome to the mellow yellow

```
#!/bin/bash

#SBATCH --account=rrg-wyeth

## Mail Options
#SBATCH --mail-user=prichmond@cmmt.ubc.ca
#SBATCH --mail-type=ALL

## CPU Usage
#SBATCH --mem=50G
#SBATCH --cpus-per-task=16
#SBATCH --time=4-0:00
#SBATCH --nodes=1

## Output and Stderr
#SBATCH --output=%x-%j.out
#SBATCH --error=%x-%j.error
```

You will need to change these to be relevant to your own use case

This header information contains info about the account to bill for these hours, I want it to mail me, how much RAM and CPUs I need over a single node, and where to send standard error and output

WESTGRID

UBC UBC100 Advanced Research Computing

Load my necessary tools

```
# Requirements
module load python/2.7.14
pip install numpy --user
pip install macs2 --user
module load bwa
module load samtools
MACS2=$HOME/.local/bin/macs2
```

I’m also going to load the necessary modules, and install a local version of MACS2 to my home directory.

Then, I set the MACS2 variable (blue guy) to be the command which calls the MACS2 tool. You’ll see why later

You’re going to need a reference genome next

```
# Genome
GENOME="/cvmfs/ref.mugqic/genomes/species/Homo_sapiens.GRCh38/genome/Homo_sapiens.GRCh38.fa"
INDEX="/cvmfs/ref.mugqic/genomes/species/Homo_sapiens.GRCh38/genome/bwa_index/Homo_sapiens.GRCh38.fa"
```

Next, I specify the genome I want to use to map my data against. I realize you won’t all work in human, but if you work in a model organism I recommend checking out this repository for genomes:

/cvmfs/ref.mugqic/genomes/

Here, I’m using their **BWA index**, and their **Fasta file**

Reference Genome, Fasta file format

Reference genomes are packaged into fasta files.
Format:

```
>chromosome1_Name OtherChromInfo AccessionInfo Etc.
NNNNNNATTCGTTGATGGATAGCATGATCAGTAGACATGACATGACAGATGAGGGATATGATGACCA
CCACCCAGATTCCCGGCCGGCCGGCCGGCCCGGGCCGGCCGGCCGGGCCCGGCTATATATATATA
CATAG ….
>chromosome2_Name OtherChromInfo AccessionInfo Etc.
NNNNNNNCCCCGGCCGGCCGGCCGGCCCGGGCCGGCCGGCCGGGCCCGGCTATATATATATACAT
AGATGATCAGTAGACATGACATGACAGATGAGGGATATGATGACCACCACCCAGATTGGAGTTGCCA
GAT
```

We need to “index” this genome in order to map to it. There are many different genome indexing strategies. For bwa, we use the command bwa index, which creates an FM-Index of the genome.

```
$ bwa index <in.fasta>
```

This will generate these files:
genome.fa.amb, genome.fa.ann, genome.fa.bwt, genome.fa.pac, genome.fa.sa

Set some more variables

```
# Globals
ID="ATAC-seq"
WORKDIR=/scratch/richmonp/TRAINING/APRIL2019/DATA2/$ID
PLATFORM="Illumina"
SAMPLE="heart_left_ventricle"
THREADS=16
mkdir $WORKDIR
cd $WORKDIR
```

I'm setting an identifier, a working directory (change this if you want to use the script yourself) ,the sample name, and the threads I'm using. I also make a working directory and change into it.

I HIGHLY RECOMMEND using variables like this within your scripts. It will make it possible to easily change out a single variable or path, and the script can remain functional

Set some more variables

```
# Globals
ID="ATAC-seq"
WORKDIR=/scratch/richmonp/TRAINING/APRIL2019/DATA2/$ID
PLATFORM="Illumina"
SAMPLE="heart_left_ventricle"
THREADS=16
mkdir $WORKDIR
cd $WORKDIR
```

You will need to change this directory to be relevant to your own use case

I'm setting an identifier, a working directory (change this if you want to use the script yourself) ,the sample name, and the threads I'm using. I also make a working directory and change into it.

I HIGHLY RECOMMEND using variables like this within your scripts. It will make it possible to easily change out a single variable or path, and the script can remain functional

WESTGRID

UBC UBC100 Advanced Research Computing

Set even more variables, and download some data

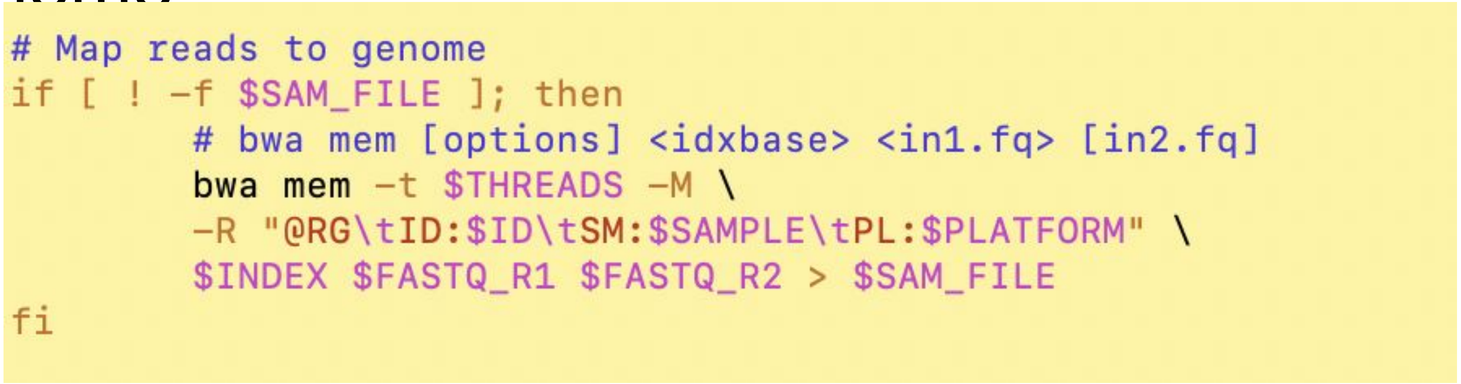
```
# Files
FASTQ_R1=$WORKDIR/$SAMPLE.R1.fastq.gz
FASTQ_R2=$WORKDIR/$SAMPLE.R2.fastq.gz
SAM_FILE=$WORKDIR/$SAMPLE.sam
BAM_FILE=$WORKDIR/$SAMPLE.bam
MACS2_DIR=$WORKDIR/MACS2
PEAKS_FILE=$MACS2_DIR/${SAMPLE}_peaks.narrowPeak

# Download ENCODE data
wget https://www.encodeproject.org/files/ENCFF766IGD/@@download/ENCFF766IGD.fastq.gz
wget https://www.encodeproject.org/files/ENCFF075UOA/@@download/ENCFF075UOA.fastq.gz
mv $WORKDIR/ENCFF766IGD.fastq.gz $FASTQ_R1
mv $WORKDIR/ENCFF075UOA.fastq.gz $FASTQ_R2
```

Here I set some file names, including for files that don’t exist yet.

Then I download some data, and rename it according to the files I want them to be called.

If you want to explore lots of these datasets to download, use the www.encodedata.org website.

Let the games begin: Mapping Reads to the Genome



```
# Map reads to genome
if [ ! -f $SAM_FILE ]; then
        # bwa mem [options] <idxbase> <in1.fq> [in2.fq]
        bwa mem -t $THREADS -M \
        -R "@RG\tID:$ID\tSM:$SAMPLE\tPL:$PLATFORM" \
        $INDEX $FASTQ_R1 $FASTQ_R2 > $SAM_FILE
fi
```

The little if/fi statements are to check if the output file exists, and if it does not exist, then perform the little command inside the block.

The BWA mem command is in the block, and at a minimum it needs an indexed genome, and an input fastq. I also add options -t for multithreading (using more cores), -R for a readgroup identifier (required for many tools), and -M for mapping split/secondary hits (not always needed). I also capture the standard out and place it into a SAM file.

The output SAM file

@SQ - Sequence (contig/chromosome) from reference file

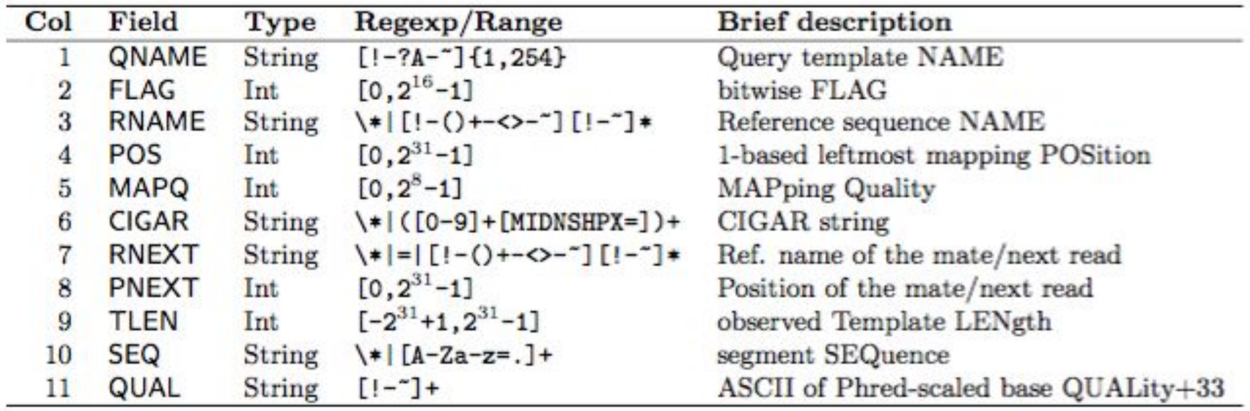
@PG - Program information about mapping

@RG - Read group information (we won’t have any here)

Tab delimited, each line is 1 read. Pairs will be next to each other in the file (e.g.

Line1: Read1

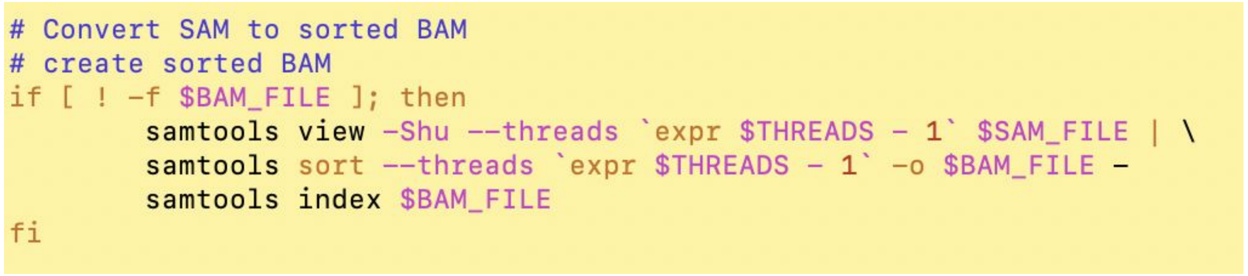
Line2: Read2



Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,254}`	Query template NAME
2	FLAG	Int	[0,2^{16}-1]	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	[0,2^{31}-1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2^{8}-1]	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next read
8	PNEXT	Int	[0,2^{31}-1]	Position of the mate/next read
9	TLEN	Int	[-2^{31}+1,2^{31}-1]	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

https://**sam**tools.github.io/hts-specs/**SAM**v1.pdf

Then we convert, sort, and index the bam file



```
# Convert SAM to sorted BAM
# create sorted BAM
if [ ! -f $BAM_FILE ]; then
        samtools view -Shu --threads `expr $THREADS - 1` $SAM_FILE | \
        samtools sort --threads `expr $THREADS - 1` -o $BAM_FILE -
        samtools index $BAM_FILE
fi
```

Here, I'm using the | to skip the step of saving the bam file, and then sorting it.

I link the two commands together to first convert the sam into bam using samtools view, and then sorting it using samtools sort.

I also add a multi-threading option, but samtools asks for "additional threads" so I take my thread# - 1.

The index command will create a .bai file next to the .bam file (file.bam.bai), which is needed for downstream tools

An easier version of samtools can be found here

```
$ module load samtools/1.3.1
```

We will use 3 samtools operations: view, sort, and index (in that order)

```
$ samtools   view  -b  <in.sam>   -o  <out.bam>
$ samtools   view  -b  Sample1.sam   -o  Sample1.bam
```

```
$ samtools   sort   <in.bam>  -o <out.sorted.bam>
$ samtools   sort   Sample1.bam  -o  Sample1.sorted.bam
```

```
$ samtools   index  <in.sorted.bam>
$ samtools   index   Sample1.sorted.bam
```

Advanced Research Computing

Session Outline

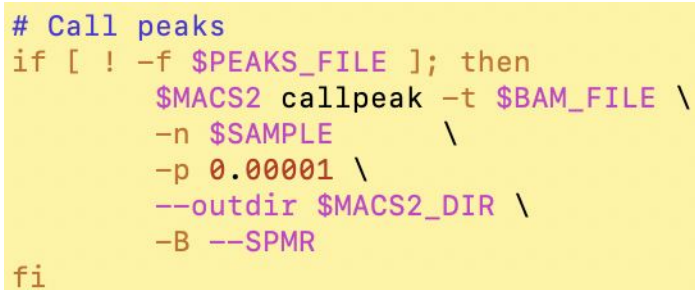
- Introduction to next generation sequencing data & diverse data types
- Transcriptional regulatory datasets and where to find them
- Mapping reads to the genome using BWA mem
- Peak calling and creating pileup files using MACS2
- Visualizing data in IGV

WESTGRID

UBC UBC100

Advanced Research Computing

The last component of the pipeline is to call peaks



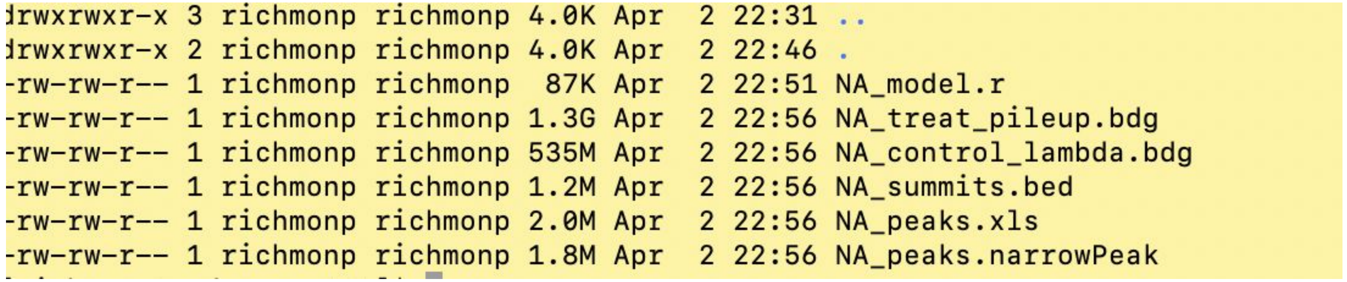
```
# Call peaks
if [ ! -f $PEAKS_FILE ]; then
        $MACS2 callpeak -t $BAM_FILE \
        -n $SAMPLE      \
        -p 0.00001 \
        --outdir $MACS2_DIR \
        -B --SPMR
fi
```

Here I'm calling peaks using MACS2.

I'm adding a sample name, I want it to output a bedgraph of normalized coverage for visualization. I'm using the ENCODE standard cutoff.

For ATAC-seq there is no "control", but for ChIP-seq pipelines there is sometimes a control sample, which you can provide as background for the peak caller.

Output files from MACS2



```
drwxrwxr-x 3 richmonp richmonp 4.0K Apr  2 22:31 ..
drwxrwxr-x 2 richmonp richmonp 4.0K Apr  2 22:46 .
-rw-rw-r-- 1 richmonp richmonp  87K Apr  2 22:51 NA_model.r
-rw-rw-r-- 1 richmonp richmonp 1.3G Apr  2 22:56 NA_treat_pileup.bdg
-rw-rw-r-- 1 richmonp richmonp 535M Apr  2 22:56 NA_control_lambda.bdg
-rw-rw-r-- 1 richmonp richmonp 1.2M Apr  2 22:56 NA_summits.bed
-rw-rw-r-- 1 richmonp richmonp 2.0M Apr  2 22:56 NA_peaks.xls
-rw-rw-r-- 1 richmonp richmonp 1.8M Apr  2 22:56 NA_peaks.narrowPeak
```

You'll get a set of files, and the ones which we will visualize are the:

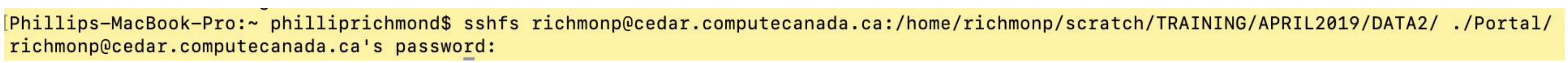
*_treat_pileup.bdg,

*_summits.bed,

*_peaks.narrowPeak, which we will convert into a bed file (just rename it .bed)

Now, for visualization

I like to use OSX-Fuse / sshfs to connect my computer to Cedar. If you don’t have it installed, google how to do so. There is also another 2-minute learn-along describing this process. If you ask me I’ll dig it up.



```
Phillips-MacBook-Pro:~ philliprichmond$ sshfs richmonp@cedar.computecanada.ca:/home/richmonp/scratch/TRAINING/APRIL2019/DATA2/ ./Portal/
richmonp@cedar.computecanada.ca's password:
```

WESTGRID

UBC UBC100

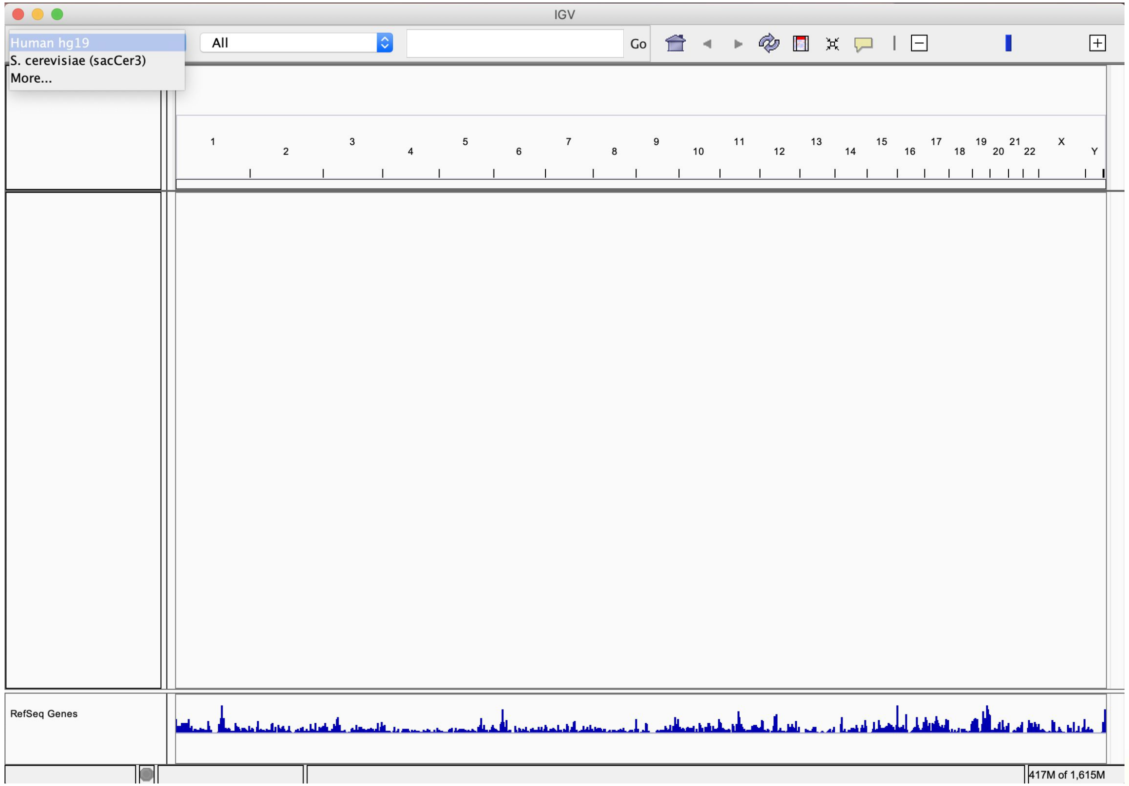
Advanced Research Computing

Then I'll open IGV

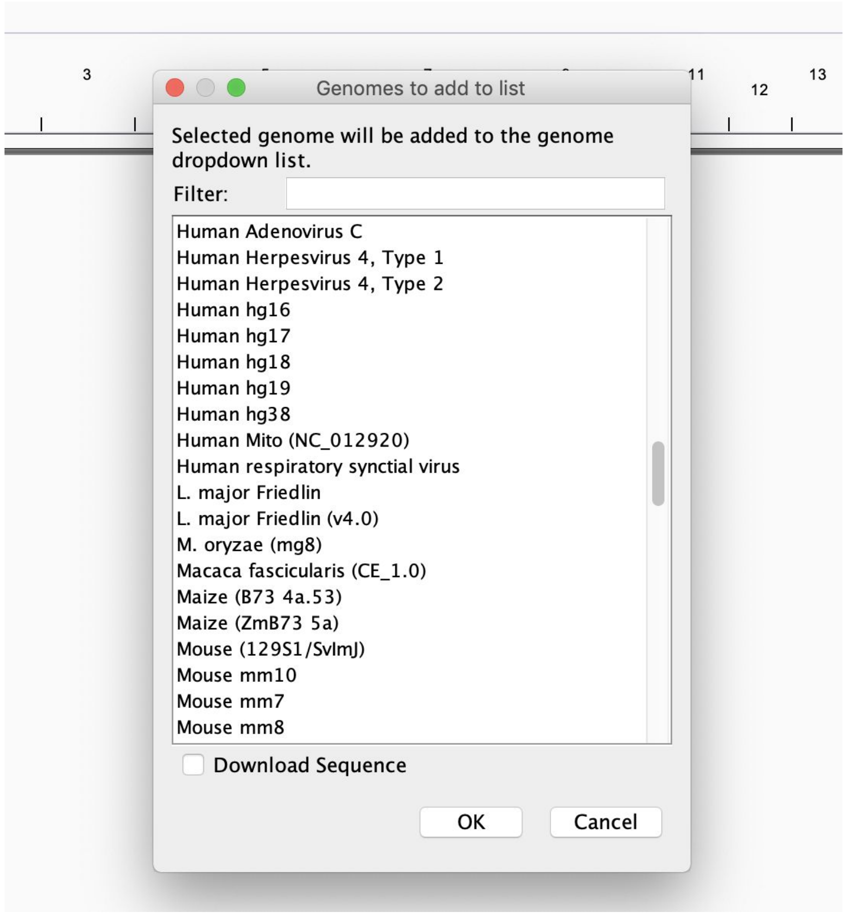
If you don't have IGV, I recommend downloading it here after this webinar:

http://software.broadinstitute.org/software/igv/download

Select the hg38 genome, if it isn't in your list..



Then go and get it



Genomes to add to list

Selected genome will be added to the genome dropdown list.

Filter:

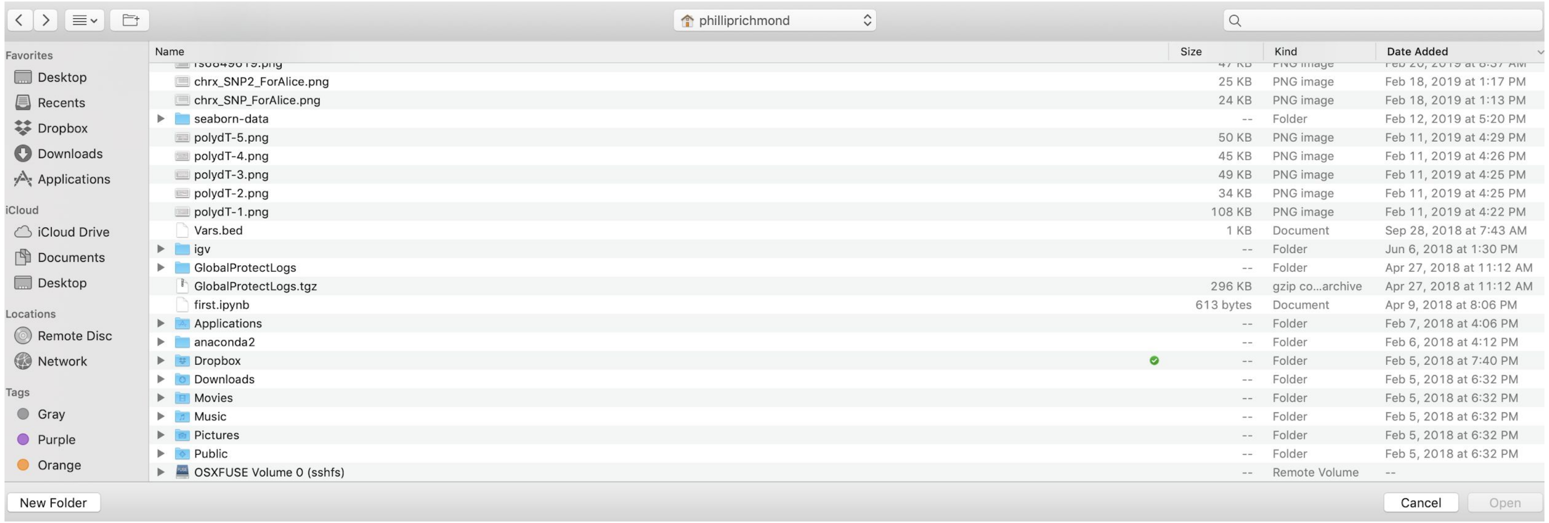
Human Adenovirus C
Human Herpesvirus 4, Type 1
Human Herpesvirus 4, Type 2
Human hg16
Human hg17
Human hg18
Human hg19
Human hg38
Human Mito (NC_012920)
Human respiratory synctial virus
L. major Friedlin
L. major Friedlin (v4.0)
M. oryzae (mg8)
Macaca fascicularis (CE_1.0)
Maize (B73 4a.53)
Maize (ZmB73 5a)
Mouse (129S1/SvImJ)
Mouse mm10
Mouse mm7
Mouse mm8

☐ Download Sequence

OK Cancel

And then load your files via File> Load from file...

And you’re going to want to select the OSXFUSE Volume 0 (sshfs)



philliprichmond

Favorites
- Desktop
- Recents
- Dropbox
- Downloads
- Applications

iCloud
- iCloud Drive
- Documents
- Desktop

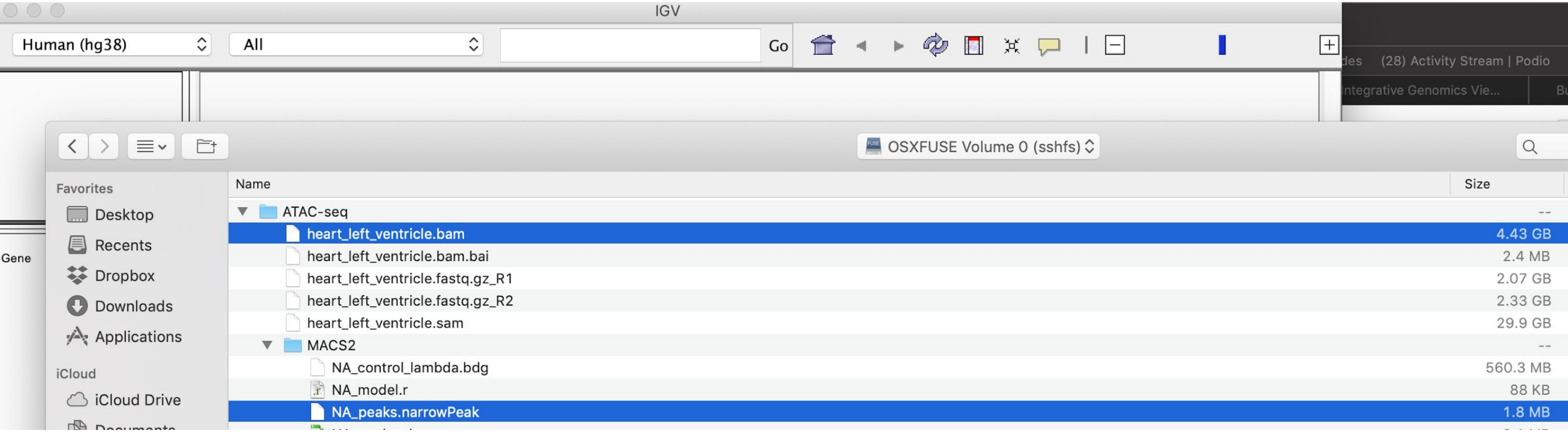
Locations
- Remote Disc
- Network

Tags
- Gray
- Purple
- Orange

Name	Size	Kind	Date Added
chrx_SNP2_ForAlice.png	25 KB	PNG image	Feb 18, 2019 at 1:17 PM
chrx_SNP_ForAlice.png	24 KB	PNG image	Feb 18, 2019 at 1:13 PM
seaborn-data	--	Folder	Feb 12, 2019 at 5:20 PM
polydT-5.png	50 KB	PNG image	Feb 11, 2019 at 4:29 PM
polydT-4.png	45 KB	PNG image	Feb 11, 2019 at 4:26 PM
polydT-3.png	49 KB	PNG image	Feb 11, 2019 at 4:25 PM
polydT-2.png	34 KB	PNG image	Feb 11, 2019 at 4:25 PM
polydT-1.png	108 KB	PNG image	Feb 11, 2019 at 4:22 PM
Vars.bed	1 KB	Document	Sep 28, 2018 at 7:43 AM
igv	--	Folder	Jun 6, 2018 at 1:30 PM
GlobalProtectLogs	--	Folder	Apr 27, 2018 at 11:12 AM
GlobalProtectLogs.tgz	296 KB	gzip co...archive	Apr 27, 2018 at 11:12 AM
first.ipynb	613 bytes	Document	Apr 9, 2018 at 8:06 PM
Applications	--	Folder	Feb 7, 2018 at 4:06 PM
anaconda2	--	Folder	Feb 6, 2018 at 4:12 PM
Dropbox	--	Folder	Feb 5, 2018 at 7:40 PM
Downloads	--	Folder	Feb 5, 2018 at 6:32 PM
Movies	--	Folder	Feb 5, 2018 at 6:32 PM
Music	--	Folder	Feb 5, 2018 at 6:32 PM
Pictures	--	Folder	Feb 5, 2018 at 6:32 PM
Public	--	Folder	Feb 5, 2018 at 6:32 PM
OSXFUSE Volume 0 (sshfs)	--	Remote Volume	--

New Folder | Cancel | Open

Load in the .bam files, and the .narrowPeak files



Name	Size
ATAC-seq	--
heart_left_ventricle.bam	4.43 GB
heart_left_ventricle.bam.bai	2.4 MB
heart_left_ventricle.fastq.gz_R1	2.07 GB
heart_left_ventricle.fastq.gz_R2	2.33 GB
heart_left_ventricle.sam	29.9 GB
MACS2	--
NA_control_lambda.bdg	560.3 MB
NA_model.r	88 KB
NA_peaks.narrowPeak	1.8 MB

Advanced Research Computing

And explore away!

This is a good region for the heart transcriptional regulation:

Chr10:21160000-23400000

Around the gene NEBL.

I'll now explore this data interactively and open to questions / comments.

To visualize the bedgraph files effectively...

You’ll need to convert them to bigwig

This can be done, and if you’re interested in learning how let me know.

In fact, Oriol has been working on it this morning so we should have the command ready for you soon!

:)

Open question and answer period

WESTGRID

UBC UBC100

Advanced Research Computing

Acknowledgements

- Phil Richmond (Teacher)
  - PhD Student Wasserman Lab, enjoys teaching
- Oriol Fornes (Co-teacher)
  - Post-doc, Deputy Group Leader, Wasserman Lab

WESTGRID

UBC UBC100

Advanced Research Computing

FLASH DEBUGGING

```
$ samtools sort Sample1.bam -o Sample1.sorted.bam
Crazy characters printing to the screen
```

Fix: This sort command doesn't use a -o
Unless you specify -T and -O as well.

```
$ samtools sort Sample1.bam Sample1.sorted
```

```
$ samtools view -bS Sample1.sam Sample1.bam
Crazy characters printing to the screen
```

Fix: This commands needs a -o for the output

```
$ samtools view -bS Sample1.sam -o Sample1.bam
```

```
$ samtools index Sample1.bam
[E::hts_idx_push] unsorted positions
samtools index: "Sample1.bam" is corrupted or unsorted
```

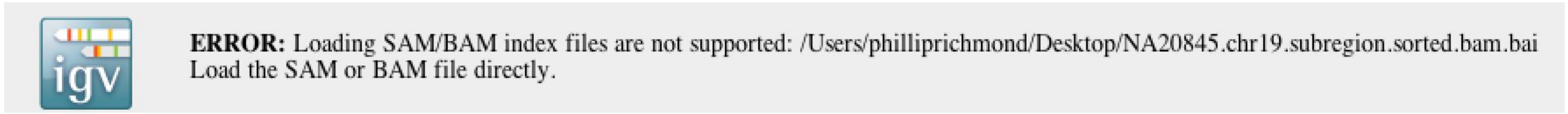
Fix: Order matters. Sort before you index

```
$ samtools index Sample1.sorted.bam
```

```
$ bwa mem -t ../GENOME/genome.fa Sample_R1.fastq Sample_R2.fastq
[E::bwa_idx_load_from_disk] fail to locate the index files
```

Fix: the -t option requires an integer. Otherwise, all the other positional arguments are out of place.

```
$ bwa mem -t 4 ../GENOME/genome.fa Sample_R1.fastq Sample_R2.fastq
```



```
ERROR: Loading SAM/BAM index files are not supported: /Users/philliprichmond/Desktop/NA20845.chr19.subregion.sorted.bam.bai
Load the SAM or BAM file directly.
```

Fix: Make sure you load the .bam file,
The .bai file just needs to be in the same directory
As the .bam file