Please download slides titled:

"Introduction to NGS bioinformatics"

https://sites.tufts.edu/biotools/tutorials/

Intro to Bioinformatics using Tufts HPC

Rebecca Batorsky

Sr Bioinformatics Specialist

Dec 2019

Outline

You'll need:

- Cluster Account – please let me know if you don't have one
- Basic knowledge of Linux

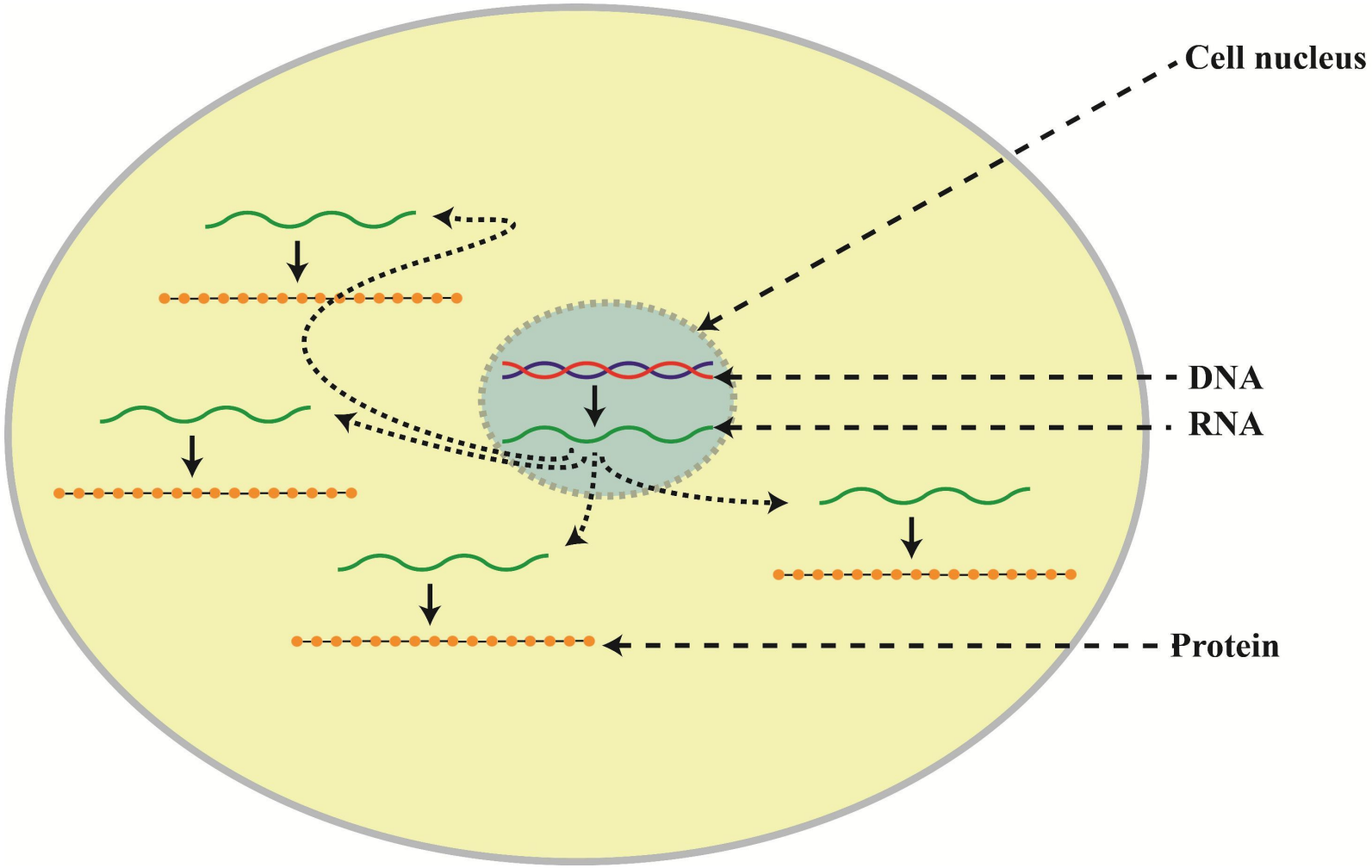
Our goals:

- Writing and running bash scripts
- Intro to several bioinformatics tools: BWA, Samtools, Picard, GATK
- Variant Calling and Interpretation

Course format:

- Short explanations followed by hands on exercises
- Working with a partner is encouraged
- Please ask questions!

DNA and RNA in a cell

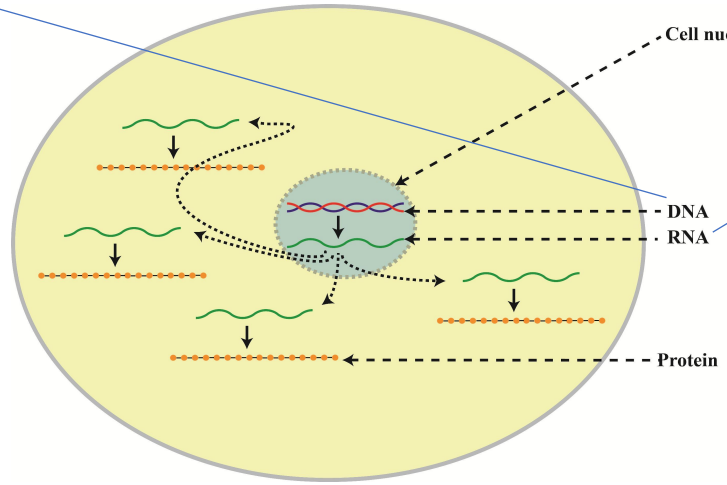


https://i0.wp.com/science-explained.com/wp-content/uploads/2013/08/Cell.jpg

Two common analysis goals

DNA Sequencing

- Fixed copy of a gene per cell
- Analysis goal:
  Variant calling and interpretation



RNA Sequencing

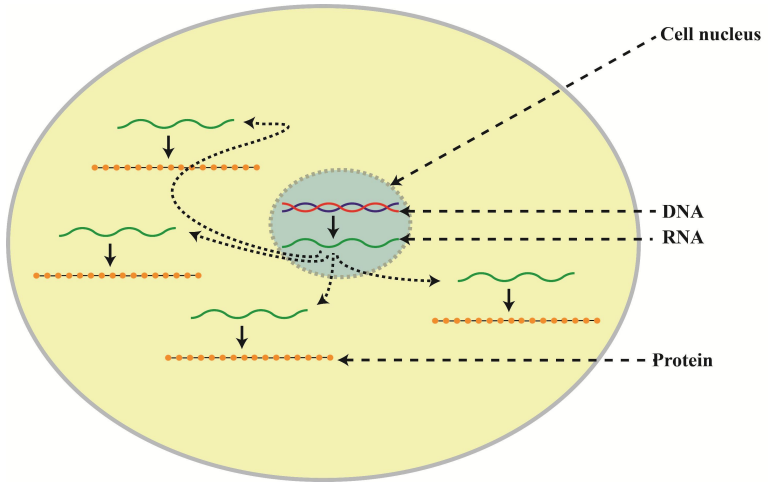
- Copy of a transcript per cell depends on gene expression
- Analysis goal: Differential expression and interpretation

https://i0.wp.com/science-explained.com/wp-content/uploads/2013/08/Cell.jpg

Today we will cover DNA sequencing

DNA Sequencing

- Fixed copy of a gene per cell
- Analysis goal:
  Variant calling and interpretation



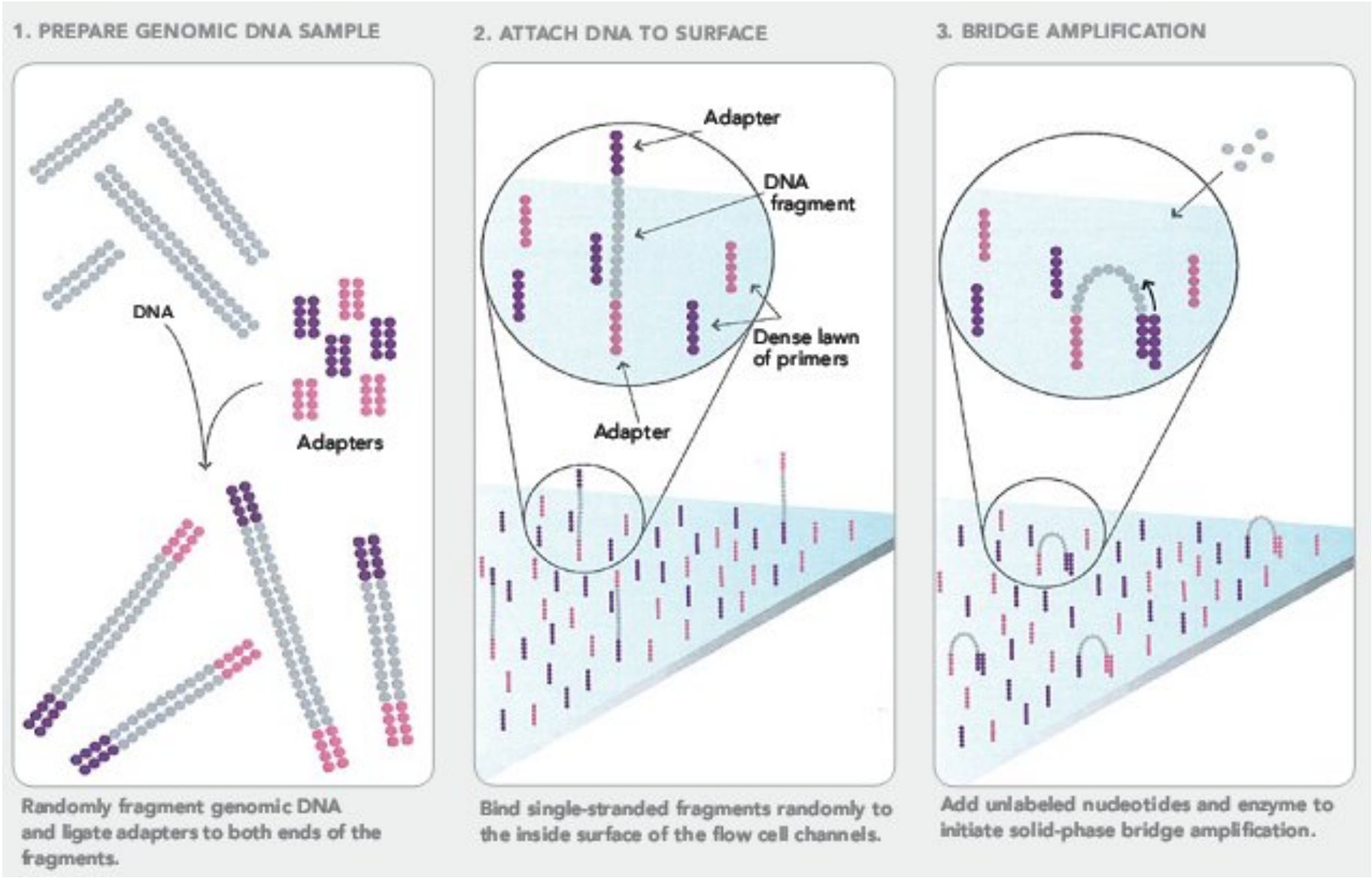
Not today!

RNA Sequen

- Copy of a g cell depends on gene ex
- Analy l: Diffe l expre n and inter tion

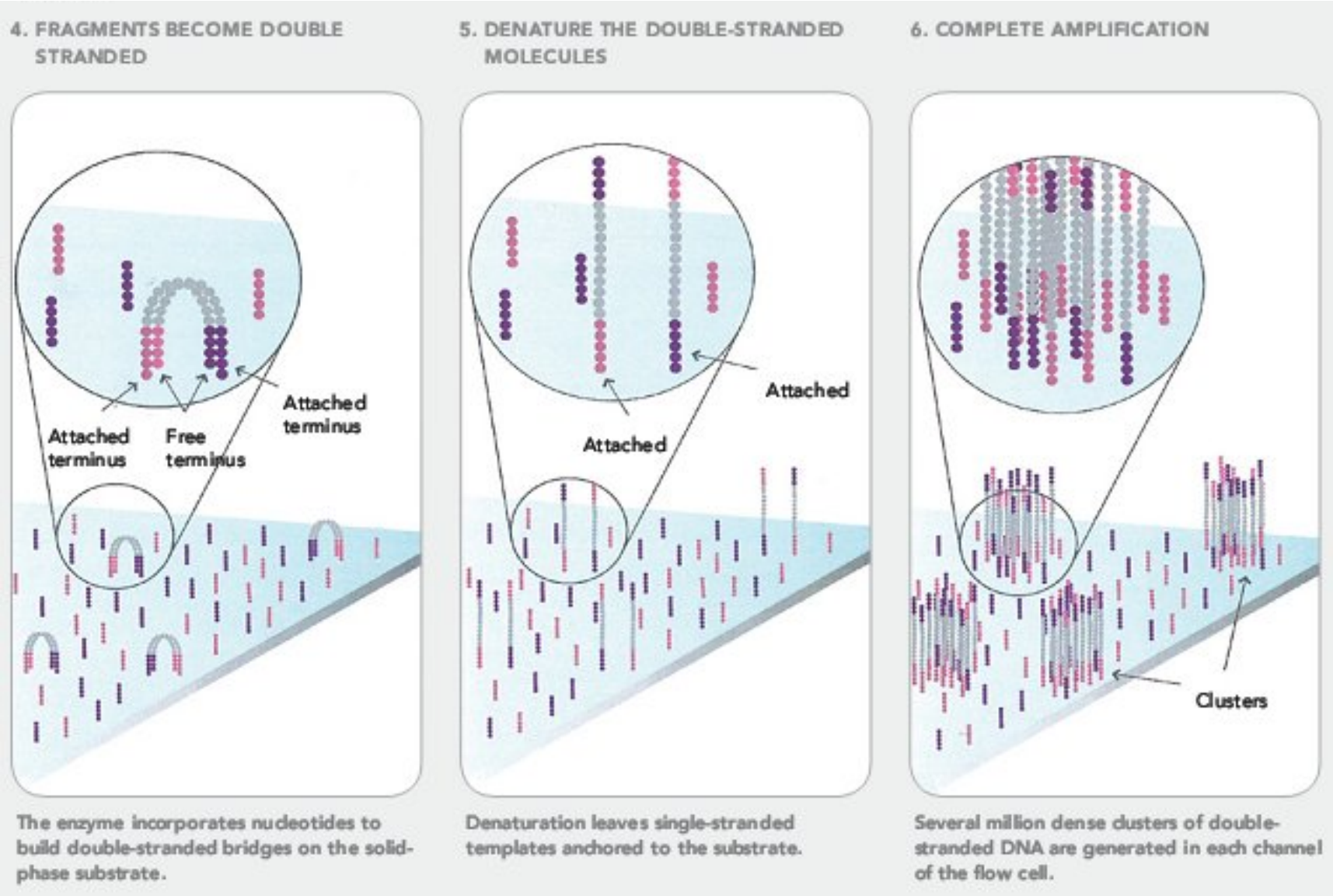
https://i0.wp.com/science-explained.com/wp-content/uploads/2013/08/Cell.jpg

Next Generation Sequencing (NGS)



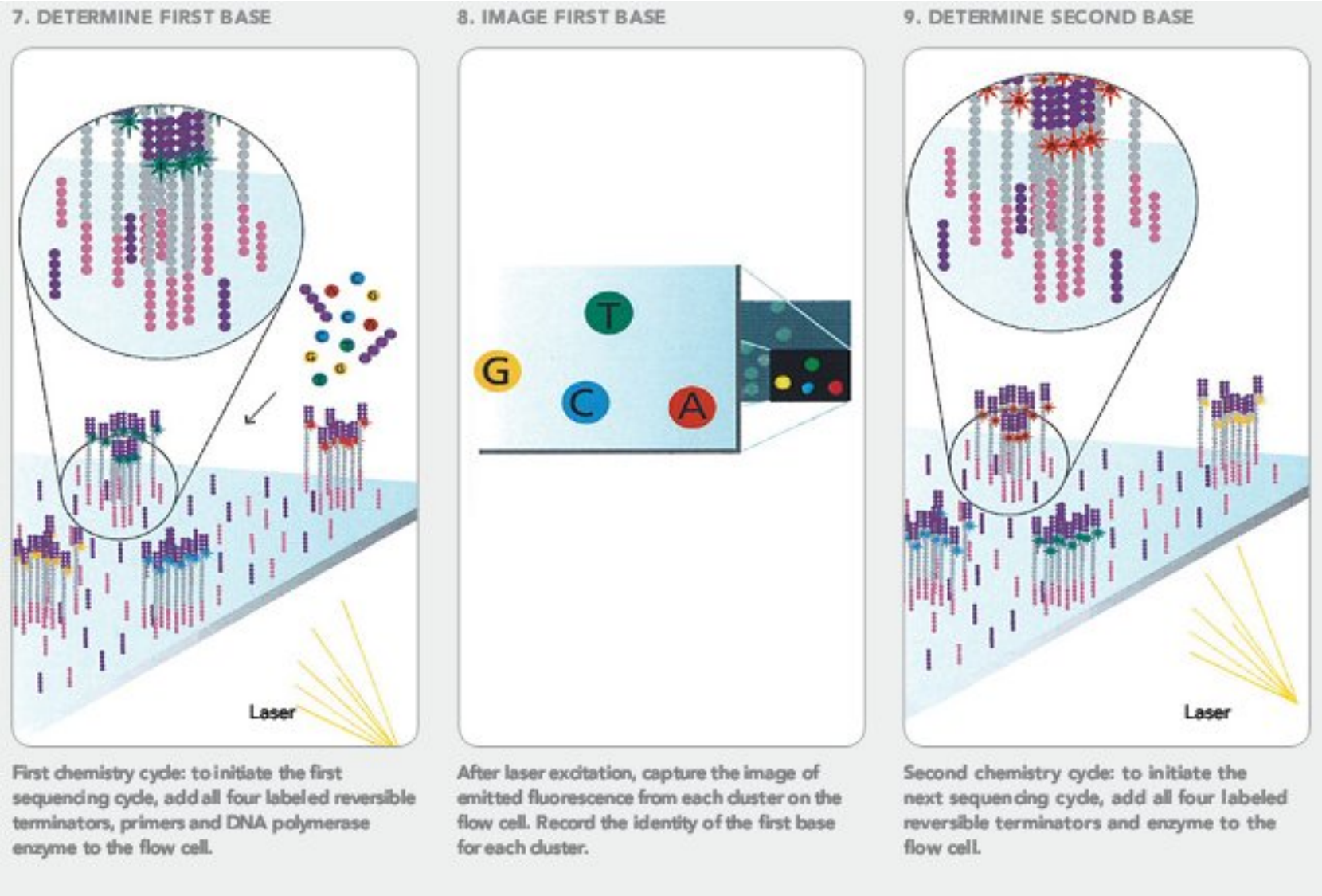
https://sites.google.com/site/himbcorelab/illumina_sequencing

Next Generation Sequencing (NGS)



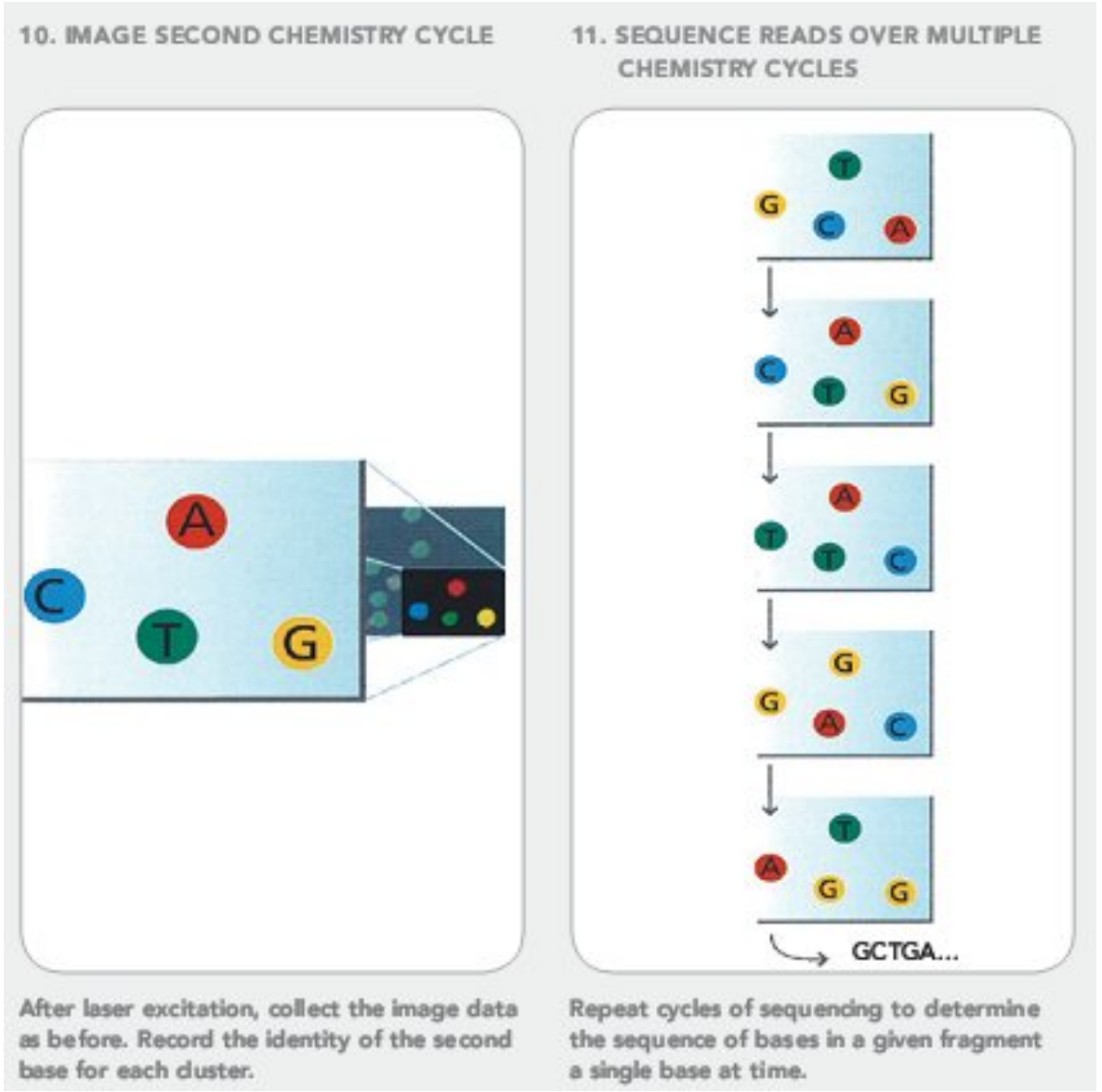
https://sites.google.com/site/himbcorelab/illumina_sequencing

Next Generation Sequencing (NGS)



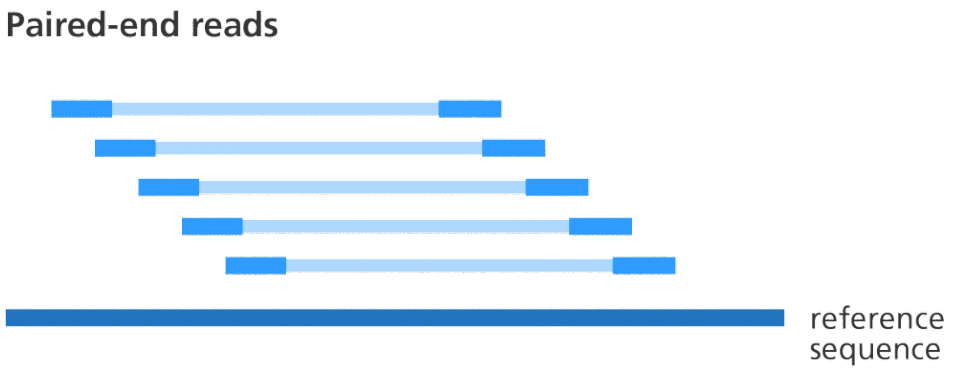
https://sites.google.com/site/himbcorelab/illumina_sequencing

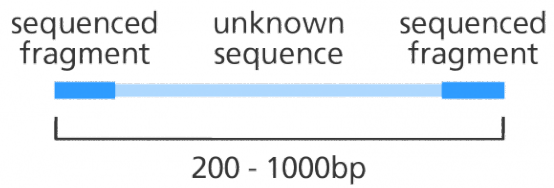
Next Generation Sequencing (NGS)



https://sites.google.com/site/himbcorelab/illumina_sequencing

Paired end vs Single end reads





"Insert Size"

https://www.biostars.org/p/267167/

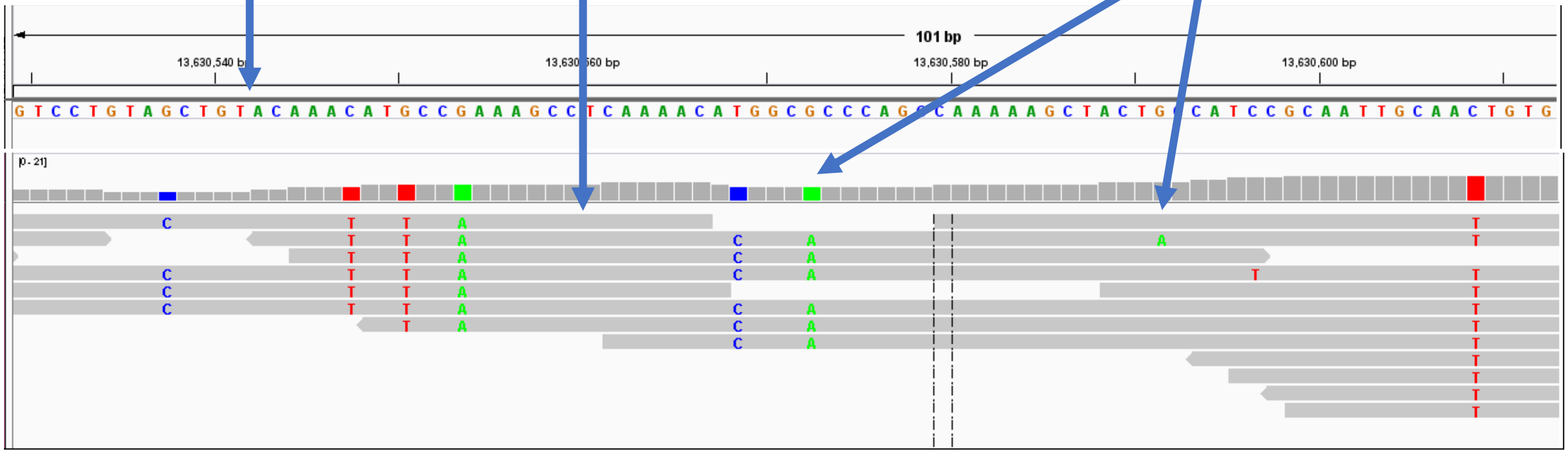
How do we make sense of short reads?

Today: we'll **align** to a **reference sequence** and look for **variants**

https://www.abmgood.com/marketing/knowledge_base/next_generation_sequencing_data_analysis.php

Variant Calling

- A **reference sequence** is a previously determined sequence from your organism
- **Reads** are aligned to the reference based on sequence similarity
- **Variants** are positions where your sequences differ from the reference

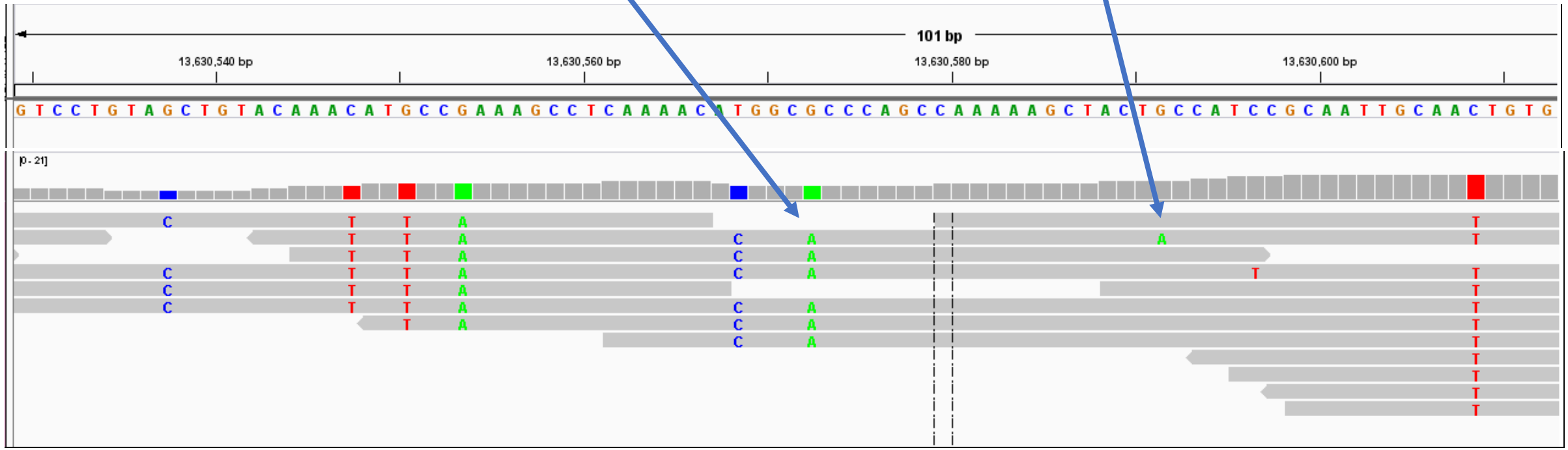


Variant Calling

Position 13,635,567
G -> A
6/6 reads -> High confidence

position 13,630,586
G -> A
1/8 reads -> Low confidence

We would like a list of variants along with the confidence



Interpretation

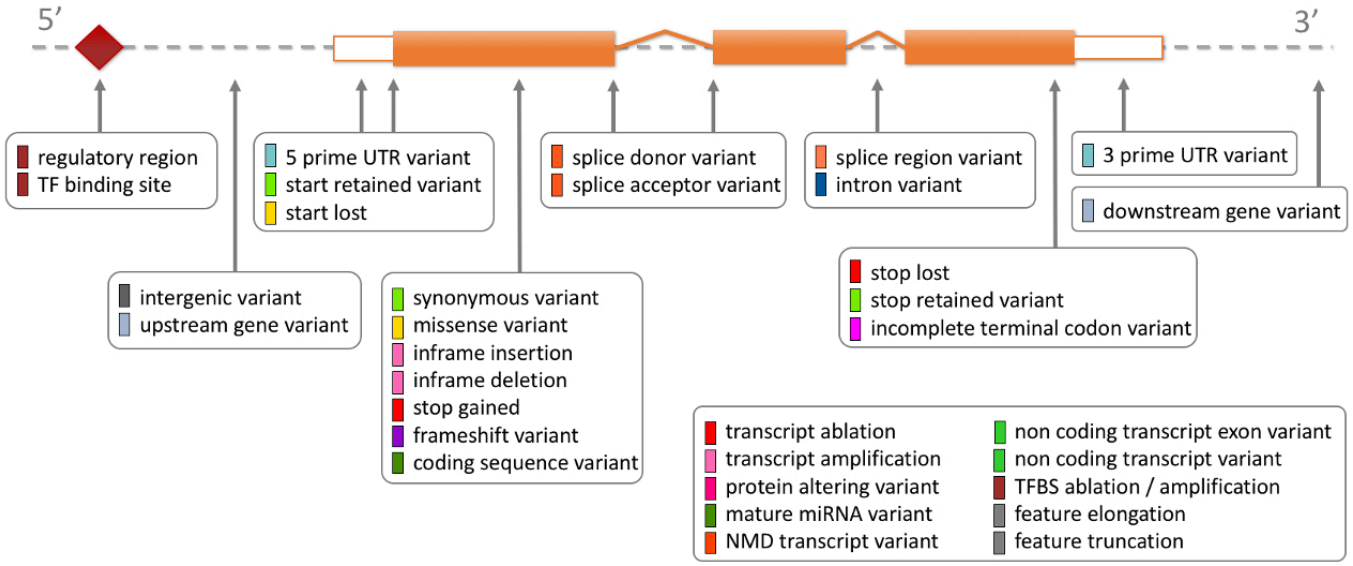
Position 13,635,567
G -> A
6/6 reads -> High confidence

ClinVar: Database of variants in relation to human health

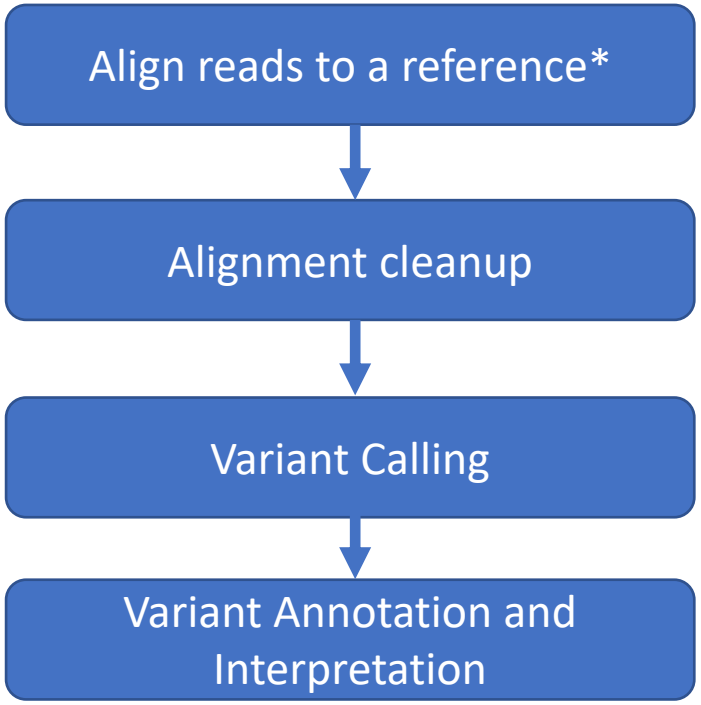
NM_005902.3(SMAD3):c.364G>A (p.Val122Met) — Cite this record

Interpretation:	**Conflicting interpretations of pathogenicity** **Likely pathogenic(1);Uncertain significance(1)**
Review status:	★☆☆☆ criteria provided, conflicting interpretations
Submissions:	2 (Most recent: Jun 10, 2016)
Last evaluated:	Feb 24, 2016
Accession:	VCV000155836.1
Variation ID:	155836
Description:	single nucleotide variant

Variant Effect Predictor (VEP) : what is the predicted consequence of the variant in a gene transcript?



Variant Calling workflow



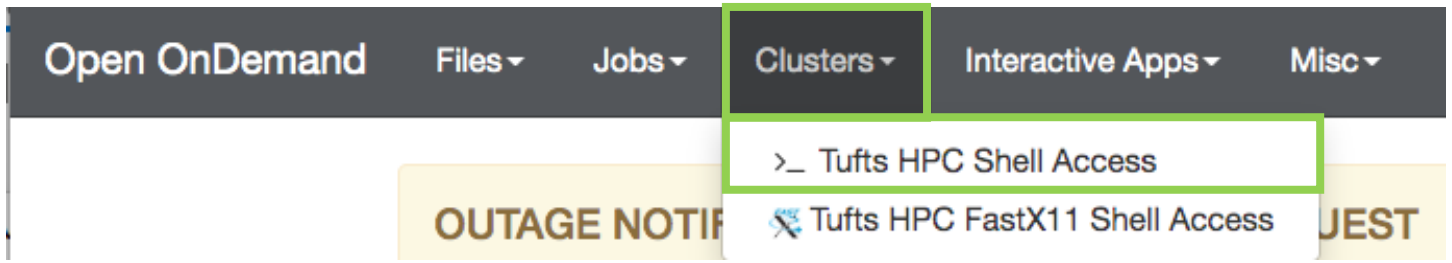
* Note we are skipping the quality control and cleanup of raw reads

https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

Log in to the Tufts HPC cluster

1. Open a **Chrome** browser and log in to **ondemand.cluster.tufts.edu**

2. Choose **Clusters** -> **Tufts HPC Shell Access**



3. Type password – it's hidden for security purposes

```
tutln01@login.cluster.tufts.edu's password:
```

4. You'll get a welcome message ending in a bash prompt

```
[tutln01@login001 ~]$
```

5. Type `clear` to clear the screen

Find 500M for your analysis in home dir

Option 1: Home directory

Check your storage options on the **login node** of the cluster by typing:

```
showquota
```

Result:

```
Home Directory Quota
Disk quotas for user tutln01 (uid 31394):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
hpcstore03:/hpc_home/home
                 2948M   5120M   5120M           20841   4295m   4295m

Listing quotas for all groups you are a member of
Group: facstaff Usage: 16723494272KB    Quota: 214748364800KB   Percent Used: 7.00%
```

Your data usage

Your data limit

Find 500M for your analysis in project dir

Option 1: Home directory

Check your storage options on the cluster by typing

```
showquota
```

Result:

```
Home Directory Quota
Disk quotas for user tutln01 (uid 31394):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
hpcstore03:/hpc_home/home
                  2948M   5120M   5120M           20841   4295m   4295m

 Your can use the storage spaces associated with these groups:

Listing quotas for all groups you are a member of
Group: facstaff  Usage: 16723494272KB    Quota: 214748364800KB   Percent Used: 7.00%
```

Option 2: Group storage

Have paths like /cluster/tufts/your-lab-name/your-user-name/

Set up for our analysis

1. Get an interactive session on a compute node by typing:

```
srun --pty  --reservation bioworkshop -p preempt -t 3:00:00  --mem 16G  -N 1 -n 4 bash
```

→

```
srun: job 47030246 queued and waiting for resources
srun: job 47030246 has been allocated resources
```

2. Change to your home:

```
[tutln01@pcomp031 ~]$
```

```
cd
```

Or project directory:

```
cd /cluster/tufts/your-lab-name/your-user-name
```

3. Copy the course folder

```
cp /cluster/tufts/bio/tools/intro-to-ngs.tar.gz .
```

3. Unzip the course folder

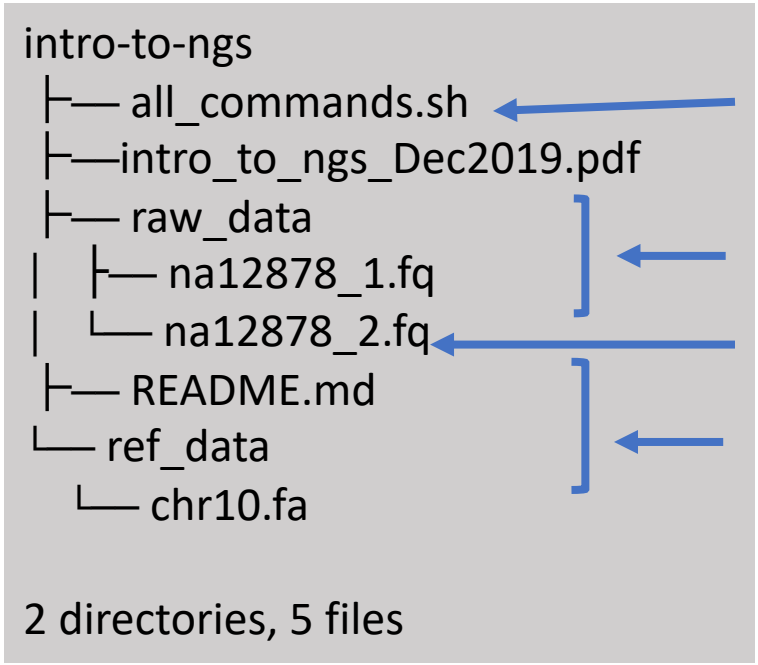
```
tar -xvzf intro-to-ngs.tar.gz
```

Also available via: git clone https://gitlab.tufts.edu/rbator01/intro-to-ngs.git

Get our data!

We've downloaded a folder called "intro-to-ngs"
Take a look at the repository contents by typing:

```
tree intro-to-ngs
```



Bash script with all commands

Folder with the raw data

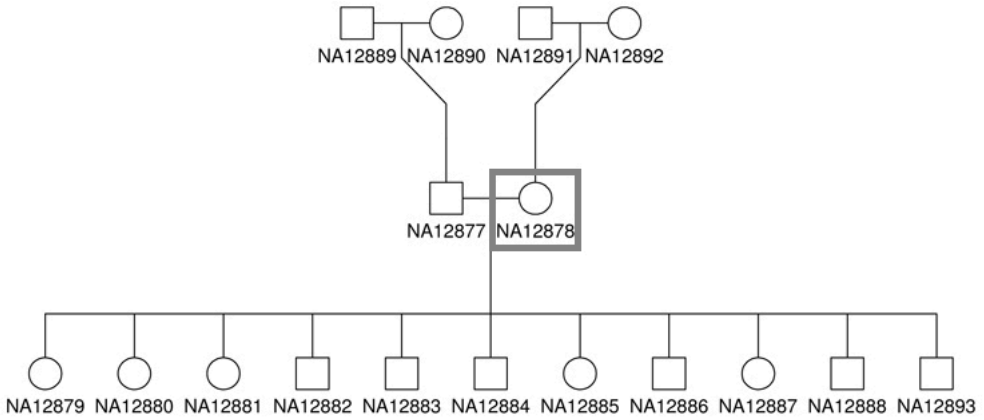
Repository instructions

Folder with the reference sequence

Data for this class

GIAB was initiated in 2011 by the National Institute of Standards and Technology "to develop the technical infrastructure (reference standards, reference methods, and reference data) to enable translation of whole human genome sequencing to clinical practice" [1]

The source DNA, known as NA12878, was taken from a single person: the daughter in a father-mother-child 'trio' (she is also mother to 11 children of her own) [4]. Father-mother-child 'trios' are often sequenced to utilize genetic links between family members.



https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course/blob/master/sessionVI/lessons/01_alignment.md

For this class, I've created a small dataset

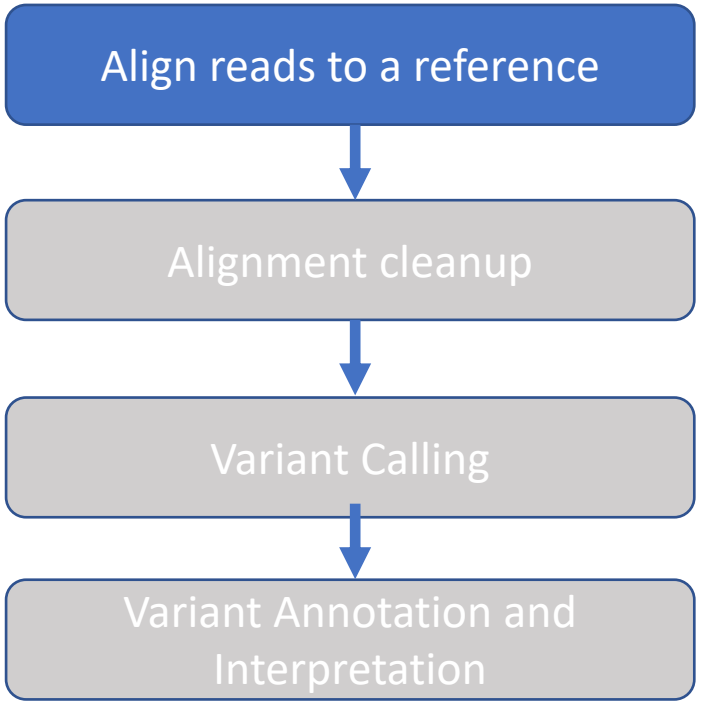
Sample: NA12878

Gene: Cyp2c19 on chromosome 10

Sequencing: Illumina, Paired End, **Exome**

Exome sequencing is a method to concentrate the sequenced DNA fragments in coding regions (exons) of the genome

Alignment to a reference



https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

BWA alignment

Burrows-Wheeler Aligner (BWA) is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome.

It has three algorithms:

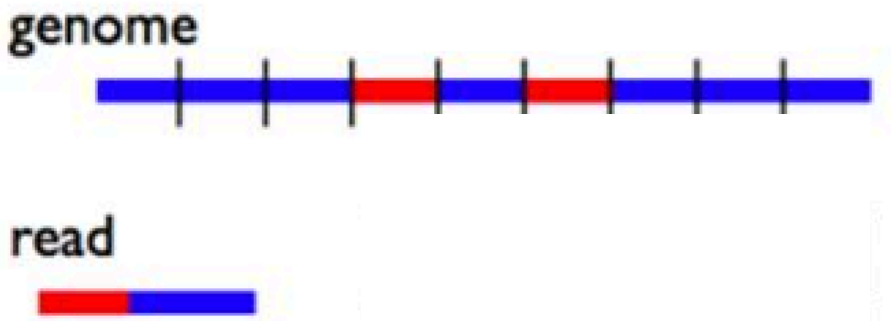
- **BWA-backtrack:** designed for Illumina sequence reads up to 100bp (3-step)
- **BWA-SW:** designed for longer sequences ranging from 70bp to 1Mbp, long-read support and split alignment
- **BWA-MEM:** optimized for 70-100bp Illumina reads

http://bio-bwa.sourceforge.net/

BWA alignment

- **BWA-MEM:** optimized for 70-100bp Illumina reads

Naïve approach to read alignment:
Compare a read to every position in the reference genome -> SLOW!

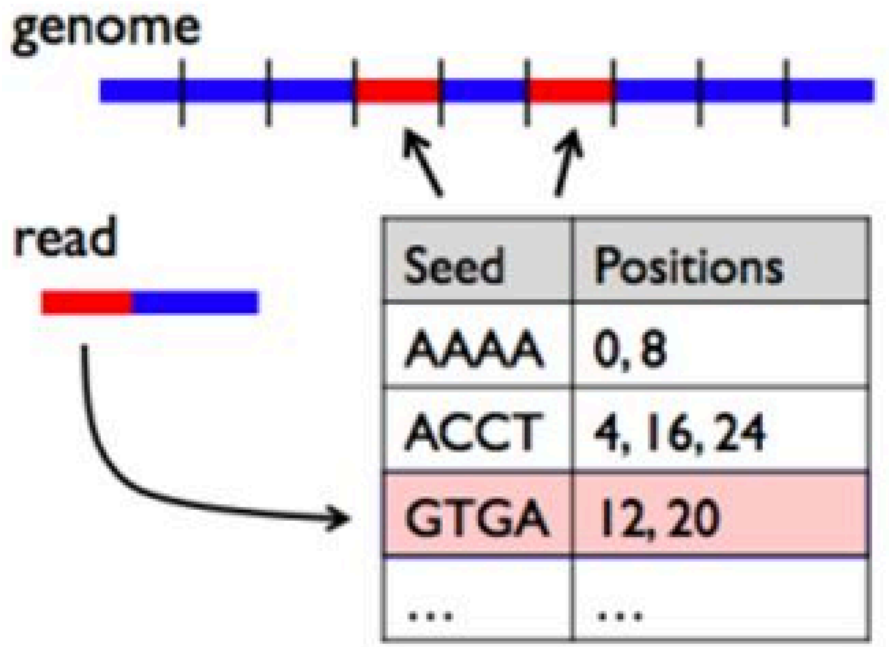


BWA alignment

- **BWA-MEM:** optimized for 70-100bp Illumina reads

Naïve approach to read alignment:
Compare a read to every position in the reference genome -> SLOW!

The solution is to create an "index" of our **reference sequence** for faster lookup.



Underlying the BWA index is the Burrows-Wheeler Transform:
http://web.stanford.edu/class/cs262/presentations/lecture4.pdf

BWA index

Change to our reference data directory:

```
cd intro-to-ngs/ref_data
```

Preview our genome using the command “head”

```
head chr10.fa
```

```
>chr10 AC:CM000672.2 gi:568336…
NNNNNNNNNNNNNNNNNNNN
…
```

FASTA format:

>sequence name

sequence

BWA index

Change to our reference data directory:

```
cd intro-to-ngs/ref_data
```

Preview our genome using the command “head”

```
head chr10.fa
```

→

```
>chr10
NNNNNNNNNNNNNNNNNNNN
…
```

Let’s try and run BWA:

```
bwa
```

Error! →

```
-bash: bwa: command not found
```

Load the module and try again:

```
module load bwa/0.7.17
```

```
bwa
```

→

```
Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.17-r1198-dirty
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index         index sequences in the FASTA format
…
```

BWA index

Let's see how to use BWA index:

```
bwa index
```

```
Usage:   bwa index [options] <in.fasta>
Options: -a STR    BWT construction algorithm …
```

Run the command:

```
bwa index chr10.fa
```

```
[bwa_index] Pack FASTA... 1.01 sec
[bwa_index] Construct BWT for the packed sequence...
[BWTIncCreate] textLength=267594844,
availableWord=30828588
…
```

This will take some time but when complete you can look at the generated files:

```
ls
```

```
chr10.fa
chr10.fa.amb
chr10.fa.ann
chr10.fa.bwt
chr10.fa.pac
chr10.fa.sa
```

Original sequence
Location of non-ATGC nucleotides
Sequence names, lengths
BWT suffix array
Binary encoded sequence
Suffix array index

FASTQ format

Now we can take a look at our raw data:

```
cd ../raw_data
ls
```

→

```
na12878_1.fq
na12878_2.fq
```

Now we can take a look at our raw data, in **FASTQ format** :

```
head na12878_1.fq
```

→

```
@SRR098401.109756285/1
GACTCACGTAACTTTAAACTCTAACAGAAATATACTA…
+
CAEFGDG?BCGGGEEDGGHGHGDFHEIEGGDDDD…
```

1. Sequence identifier: @Read ID / 1 or 2 of pair
2. Sequence
3. + (optionally lists the sequence identifier again)
4. Quality string

Base Quality Scores

The symbols we see in the read quality string are an encoding of the quality score:

```
Quality encoding: !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
                  |         |         |         |         |
   Quality score: 0........10........20........30........40
```

A quality score is a prediction of the probability of an error in base calling:

Quality Score	Probability of Incorrect Base Call	Inferred Base Call Accuracy
10 (Q10)	1 in 10	90%
20 (Q20)	1 in 100	99%
30 (Q30)	1 in 1000	99.9%

https://www.illumina.com/science/education/sequencing-quality-scores.html

Base Quality Scores

The symbols we see in the read quality string are an encoding of the quality score:

```
Quality encoding: !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
                  |         |         |         |         |
    Quality score: 0........10........20........30........40
```

A quality score is a prediction of the probability of an error in base calling:

Quality Score	Probability of Incorrect Base Call	Inferred Base Call Accuracy
10 (Q10)	1 in 10	90%
20 (Q20)	1 in 100	99%
30 (Q30)	1 in 1000	99.9%

Back to our read:

```
@SRR098401.109756285/1
GACTCACGTAACTTTAAACTCTAACAGAAATATACTA…
+
CAEFGDG?BCGGGEEDGGHGHGDFHEIEGGDDDD…
```

C –> Q = 34 -> Probability < 1/1000 of an error

https://www.illumina.com/science/education/sequencing-quality-scores.html

BWA alignment

Let's see how to run bwa mem:

```
bwa mem
```

```
Usage: bwa mem [options] <idxbase> <in1.fq> [in2.fq]
Algorithm options:
      -t INT        number of threads [1]
      -k INT        minimum seed length [19]
…
```

Since our alignment command will be long, let's write a script:

```
cd ..
```

Go back up to **intro-to-ngs** directory

```
mkdir results
```

Make a directory to store results

```
nano bwa.sh
```

Open a text editor with a file called bwa.sh

BWA alignment

Enter the following text:

```
bwa mem \
-t 2 \
-M \
-R "@RG\tID:reads\tSM:na12878" \
-o results/na12878.sam \
ref_data/chr10.fa \
raw_data/na12878_1.fq \
raw_data/na12878_2.fq
```

Ending a line with a backslash will be read as a line continuation (be careful to put a space **BEFORE** the backslash and **NOT AFTER** the backslash)

BWA alignment

Enter the following text:

```
bwa mem \
-t 2 \
-M \
-R "@RG\tID:reads\tSM:na12878" \
-o results/na12878.sam \
ref_data/chr10.fa \
raw_data/na12878_1.fq \
raw_data/na12878_2.fq
```

Parallelize with two threads

BWA alignment

Enter the following text:

```
bwa mem \
-t 2 \
-M \
-R "@RG\tID:reads\tSM:na12878" \
-o results/na12878.sam \
ref_data/chr10.fa \
raw_data/na12878_1.fq \
raw_data/na12878_2.fq
```

Needed for GATK compatibility:

- -M “mark shorter split hits as secondary” : https://www.biostars.org/p/97323/

BWA alignment

Enter the following text:

```
bwa mem \
-t 2 \
-M \
-R "@RG\tID:reads\tSM:na12878" \
-o results/na12878.sam \
ref_data/chr10.fa \
raw_data/na12878_1.fq \
raw_data/na12878_2.fq
```

Needed for GATK compatibility:

- Add a read group (RG) and sample name tag (SM) – this will show up in our alignment file

Required fields:
@RG – Read group header line identifier
ID - read group ID
SM – sample name

Our alignment file will contain this string:

```
"@RG	ID:reads	SM:na12878"
```

BWA alignment

Enter the following text:

```
bwa mem \
-t 2 \
-M \
-R "@RG\tID:reads\tSM:na12878" \
-o results/na12878.sam \
ref_data/chr10.fa \
raw_data/na12878_1.fq \
raw_data/na12878_2.fq
```

Place the output in the results folder and give it a name

BWA alignment

Enter the following text:

```
bwa mem \
-t 2 \
-M \
-R "@RG\tID:reads\tSM:na12878" \
-o results/na12878.sam \
ref_data/chr10.fa \
raw_data/na12878_1.fq \
raw_data/na12878_2.fq
```

Our bwa command accepts the arguments:

```
bwa mem [options] <idxbase> <in1.fq> [in2.fq]
```

Exit nano by typing ^X and follow prompts to save and name the file bwa.sh

BWA alignment

Let's run our script:

```
sh bwa.sh
```

```
[M::bwa_idx_load_from_disk] read 0 ALT contigs
[M::process] read 9304 sequences (707104 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (0, 2256, 0, 0)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (120, 160, 216)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 408)
[M::mem_pestat] mean and std.dev: (172.35, 67.15)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 504)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[M::mem_process_seqs] Processed 9304 reads in 1.034 CPU sec, 0.518 real sec
```

Take a look at the results directory:

```
ls results
```

→ `na12878.sam`

Sequence Alignment Map (SAM)

Take a look at the output file:

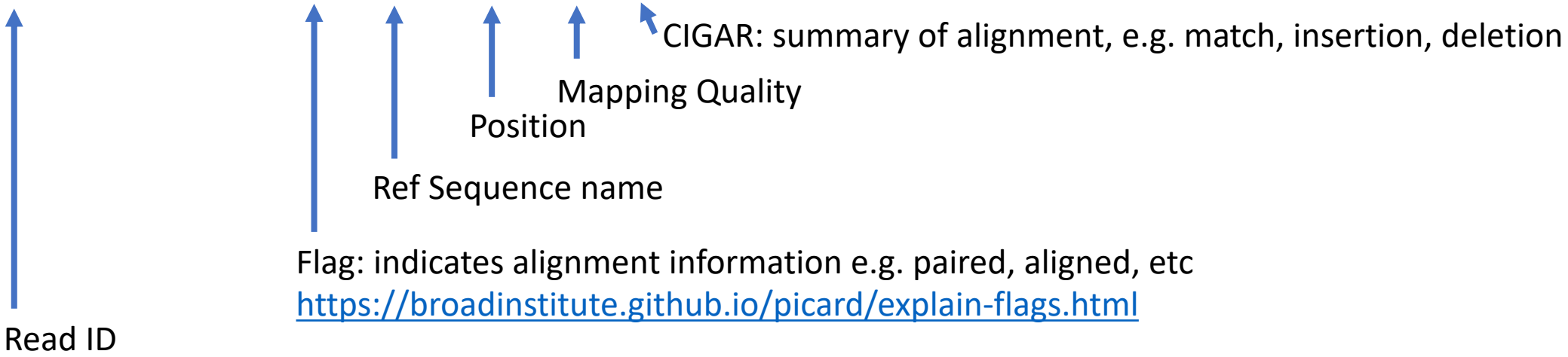
```
cd results
head na12878.sam
```

Header section:

```
@SQ     SN:chr10        LN:133797422
@RG     ID:reads        SM:na12878
@PG ID:bwa PN:bwa VN:0.7.17… CL:bwa mem -t 2 -M -R @RG\tID:reads\tSM:NA12878 -o results/na12878.sam …
```

Alignment section:

```
SRR098401.109756285 83  chr10 94760653 60 76M = 94760647 -82 CTAA…    D?@A...
SRR098401.109756285 163 chr10 94760647 60 76M = 94760653  82 ATTA…    ?>@@...
```



www.samformat.info

Alignment Quality Control

How well did our reads align to the reference genome?

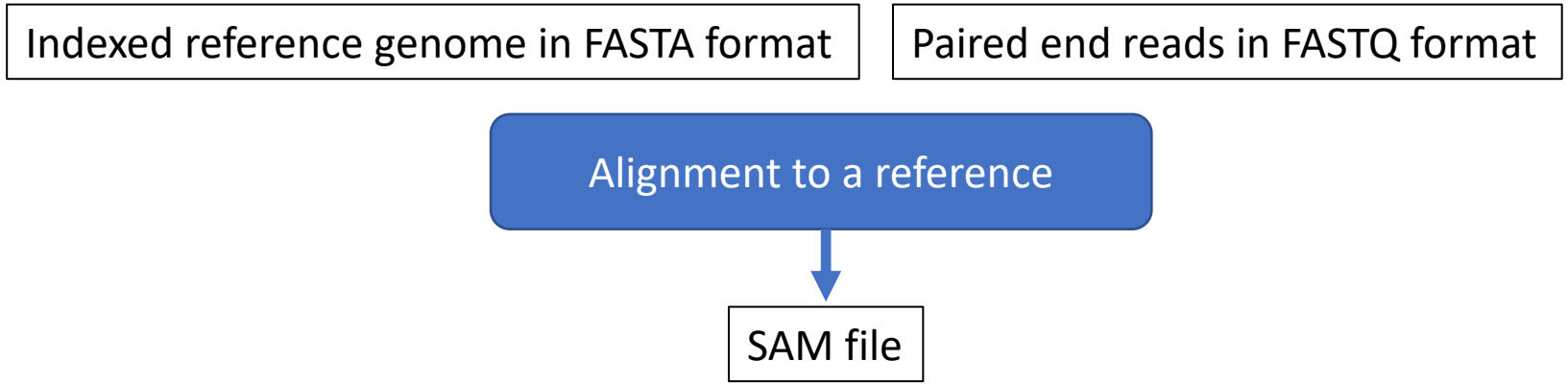
Use a tool called Samtools, summarizes SAM flags:

```
module load samtools/1.9
```

```
samtools flagstat na12878.sam
```

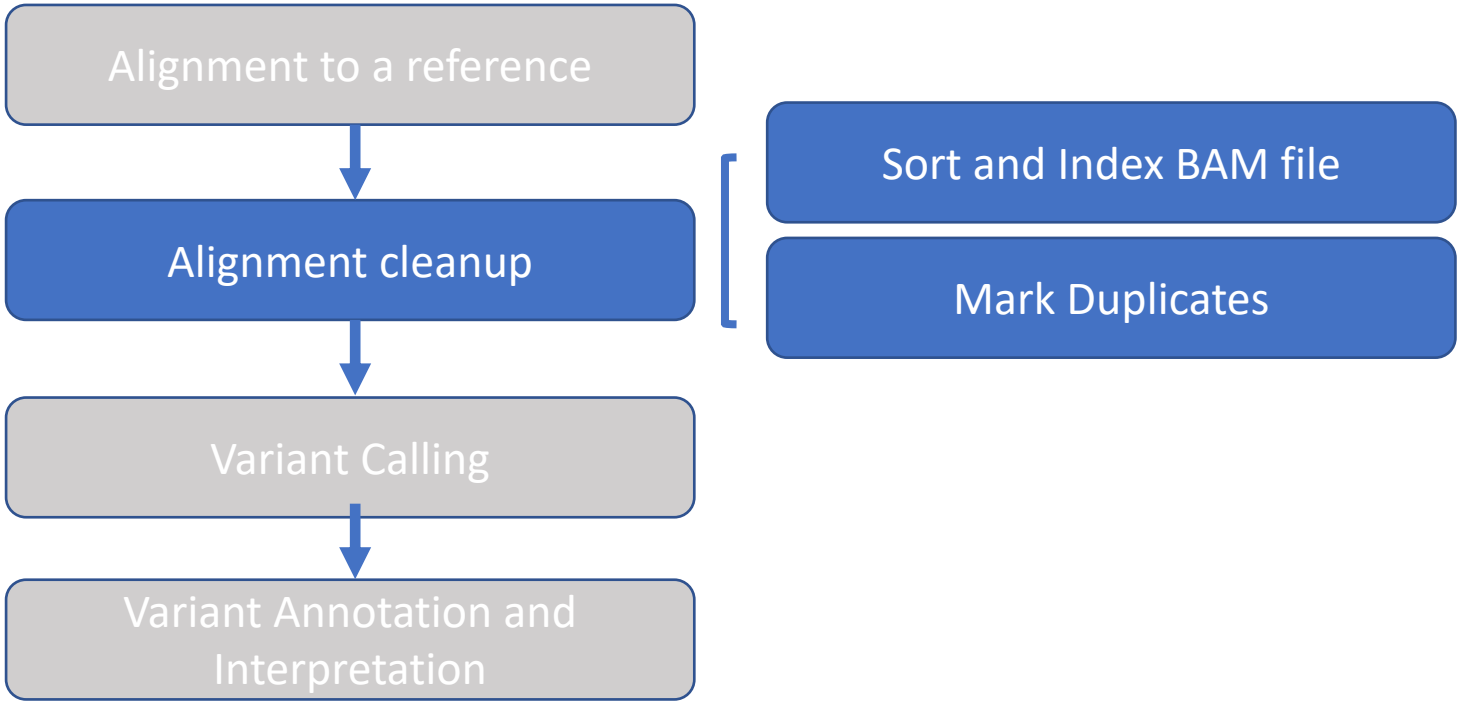
```
9306 + 0 in total (QC-passed reads + QC-failed reads)
2 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
9271 + 0 mapped (99.62% : N/A)
9304 + 0 paired in sequencing
4652 + 0 read1
4652 + 0 read2
9226 + 0 properly paired (99.16% : N/A)
9240 + 0 with itself and mate mapped
29 + 0 singletons (0.31% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

Summary Alignment



https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

Alignment Cleanup



https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

Sort SAM file

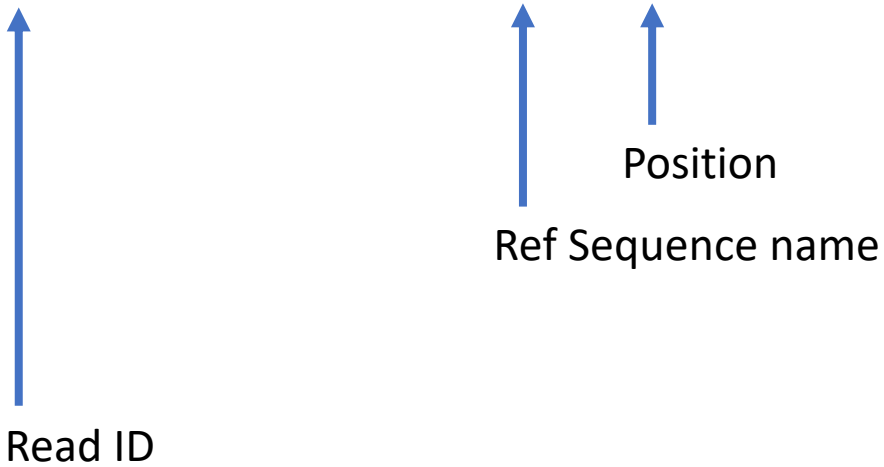
It’s necessary for downstream applications to **sort reads by reference genome coordinate**

Header

```
@SQ     SN:chr10        LN:133797422
@RG     ID:reads        SM:na12878
@PG ID:bwa PN:bwa VN:0.7.17… CL:bwa mem -t 2 -M -R @RG\tID:reads\tSM:NA12878 -o results/na12878.sam …
```

Alignment

```
SRR098401.109756285 83  chr10 94760653 60 76M = 94760647 -82 CTAA…    D?@A...
SRR098401.109756285 163 chr10 94760647 60 76M = 94760653  82 ATTA…    ?>@@...
```



Sort SAM file

It's necessary for downstream applications to **sort reads by reference genome coordinate**

Open another script in our course directory called picard.sh

```
cd ..
nano picard.sh
```

Enter the following text:

```
module load picard/2.8.0

picard SortSam \
INPUT=results/na12878.sam \
OUTPUT=results/na12878.srt.bam \
SORT_ORDER=coordinate
```

We'll use the Picard toolkit for SAM file manipulation

Output will be in BAM file, Binary Alignment Map – a compressed SAM format

Exit nano by typing ^X and follow prompts to save and name the file picard.sh

https://broadinstitute.github.io/picard/

Sort SAM file

Run the script:

```
sh picard.sh
```

```
[Fri Nov 22 15:33:22 EST 2019] picard.sam.SortSam …
…
[Fri Nov 22 15:33:22 EST 2019] picard.sam.SortSam done.
```

Take a look at the results directory:

```
ls results
```

```
na12878.sam
na12878.srt.bam
```

https://broadinstitute.github.io/picard/

Mark Duplicates in BAM file

In the sequencing process, many copies are made of a DNA fragment
The amount of duplication may not be the same for all sequences and can cause **biases** in variant calling
Therefore, we mark the duplicates so the variant caller can ignore them.

Duplicates

Duplicates removed

Mark Duplicates in BAM file

Let's continue editing our script:

```
nano picard.sh
```

Add these lines:

```
printf  'start mark duplicates\n\n'

picard MarkDuplicates \
INPUT=results/na12878.srt.bam \
OUTPUT=results/na12878.srt.markdup.bam \
METRICS_FILE=results/na12878.markdup.txt
```

Print statement to show progress

Run our script:

```
sh picard.sh
```

```
…
starting mark duplicates
[Fri Nov 22 16:03:41 EST 2019]
picard.sam.markduplicates.MarkDuplicates INPUT=…
Runtime.totalMemory()=9186050048
```

https://broadinstitute.github.io/picard/

Mark Duplicates metrics file

Take a look at our metrics file:

```
head results/na12878.markdup.txt
```

LIBRARY	UNPAIRED_READS_EXAMINED	READ_PAIRS_EXAMINED	SECONDARY_OR_SUPPLEMENTARY_RDS	UNMAPPED_READS	UNPAIRED_READ_DUPLICATES
Unknown Library	29	4620	2	35	14

READ_PAIR_DUPLICATES	READ_PAIR_OPTICAL_DUPLICATES	PERCENT_DUPLICATION	ESTIMATED_LIBRARY_SIZE
425	0	0.093214	23546

Normal % duplication for DNA exome sequencing 10-30%

Index your BAM file

Let’s continue editing our script:

```
nano picard.sh
```

Add these lines:

```
printf  'start index\n\n'

picard BuildBamIndex \
INPUT=results/na12878.srt.markdup.bam
```

Run our script:

```
sh picard.sh
```

```
…
starting BAM index
[Fri Nov 22 16:10:19 EST 2019] picard.sam.BuildBamIndex
INPUT…
Runtime.totalMemory()=2027945984
```

Index your BAM file

Let's continue editing our script:

```
nano picard.sh
```

Add these lines:

```
printf  'start index\n\n'

picard BuildBamIndex \
INPUT=results/na12878.srt.markdup.bam
```

Run our script:

```
sh picard.sh
```

```
…
starting BAM index
[Fri Nov 22 16:10:19 EST 2019] picard.sam.BuildBamIndex
INPUT…
Runtime.totalMemory()=2027945984
```

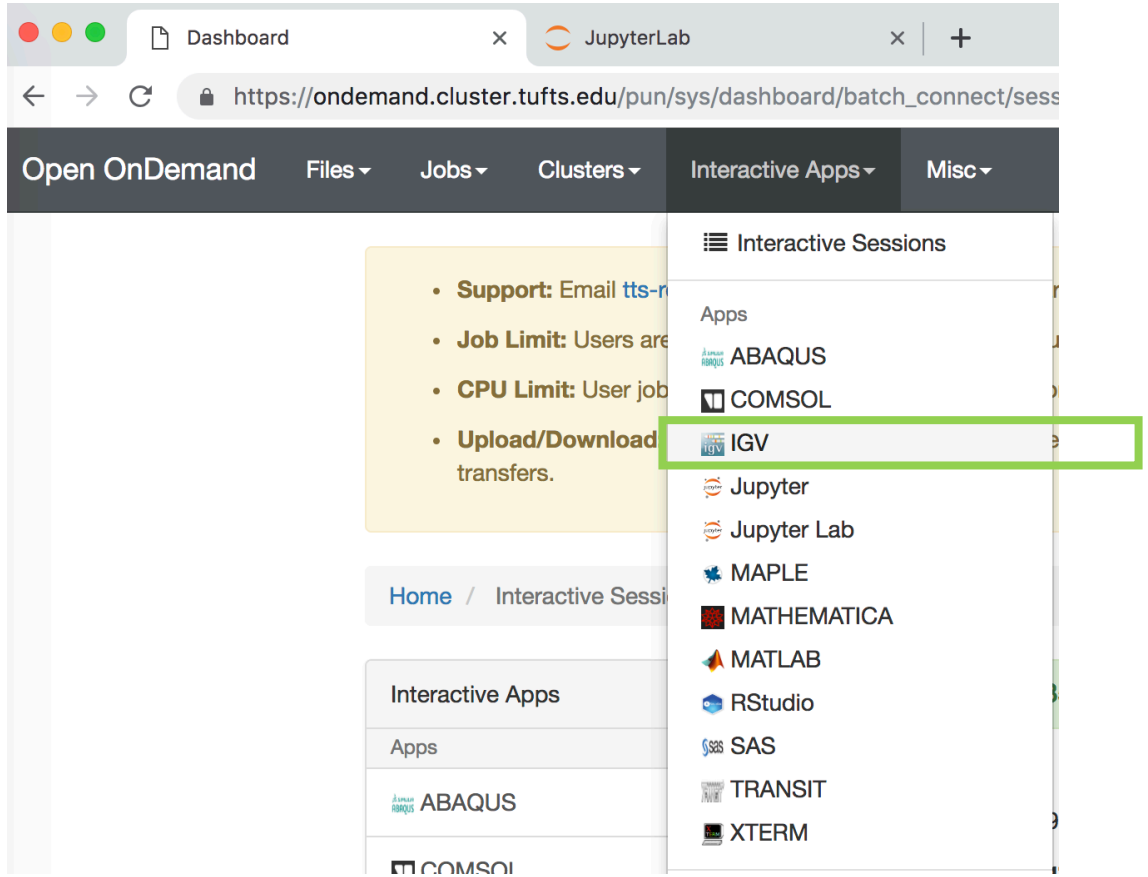
Results directory should now contain:

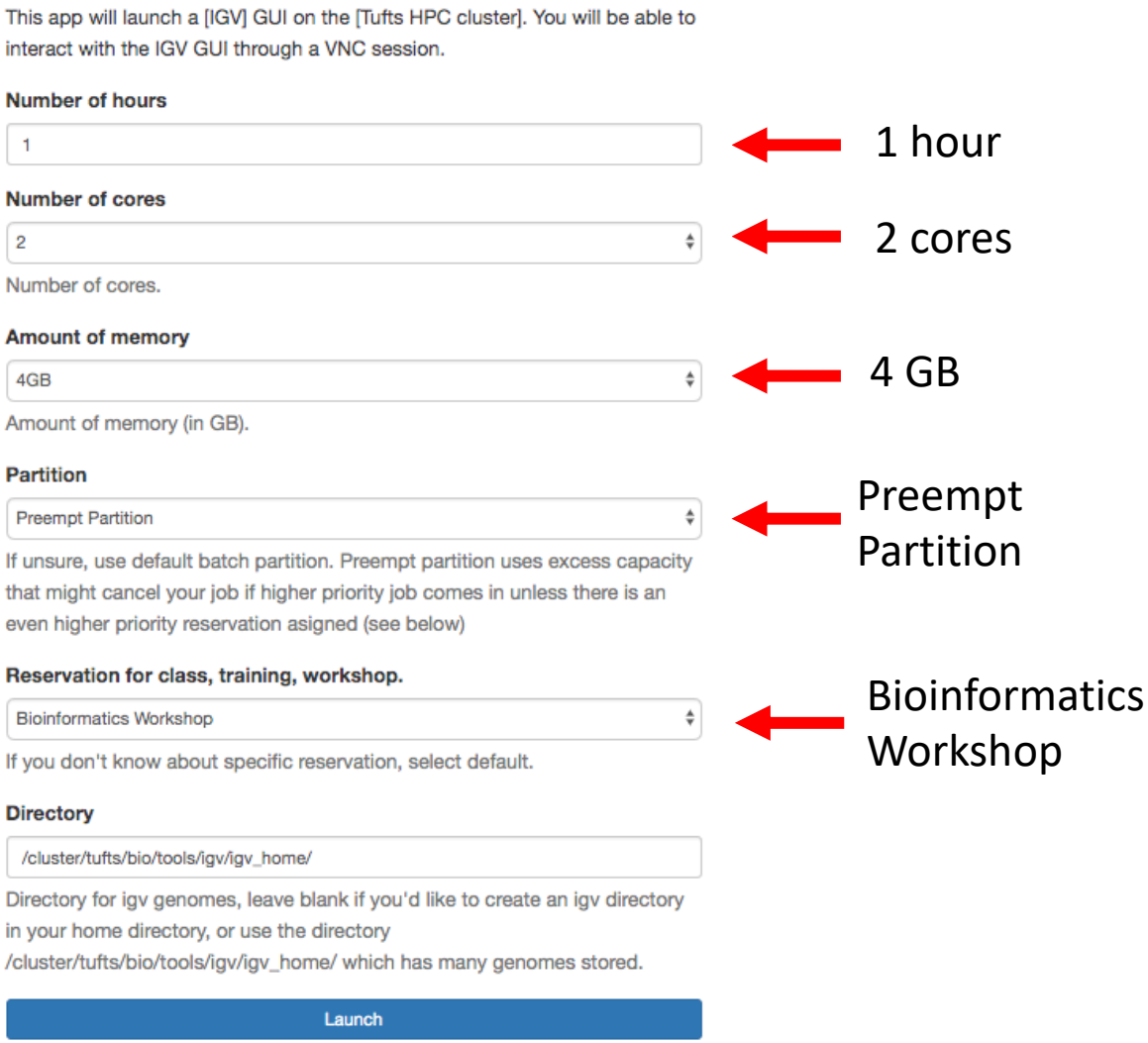
```
ls results
```

na12878.sam
na12878.srt.bam
na12878.srt.markdup.bam
na12878.markdup.txt
na12878.srt.markdup.bai

BAM Visualization with IGV

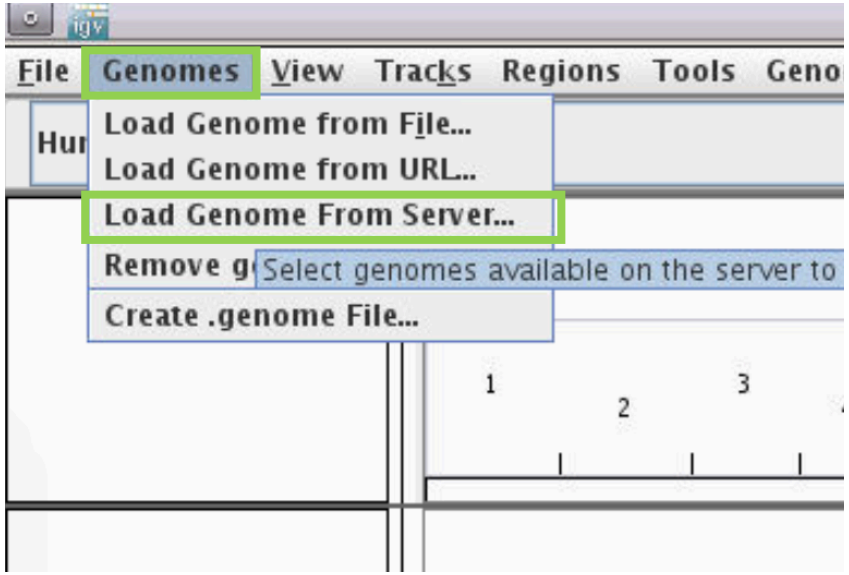
1. Chrome web browser to **ondemand.cluster.tufts.edu** -> login.
2. Choose Interactive Apps->IGV
3. Set parameters, click “Launch”
4. Click “Launch NoVNC in New Tab” when it appears



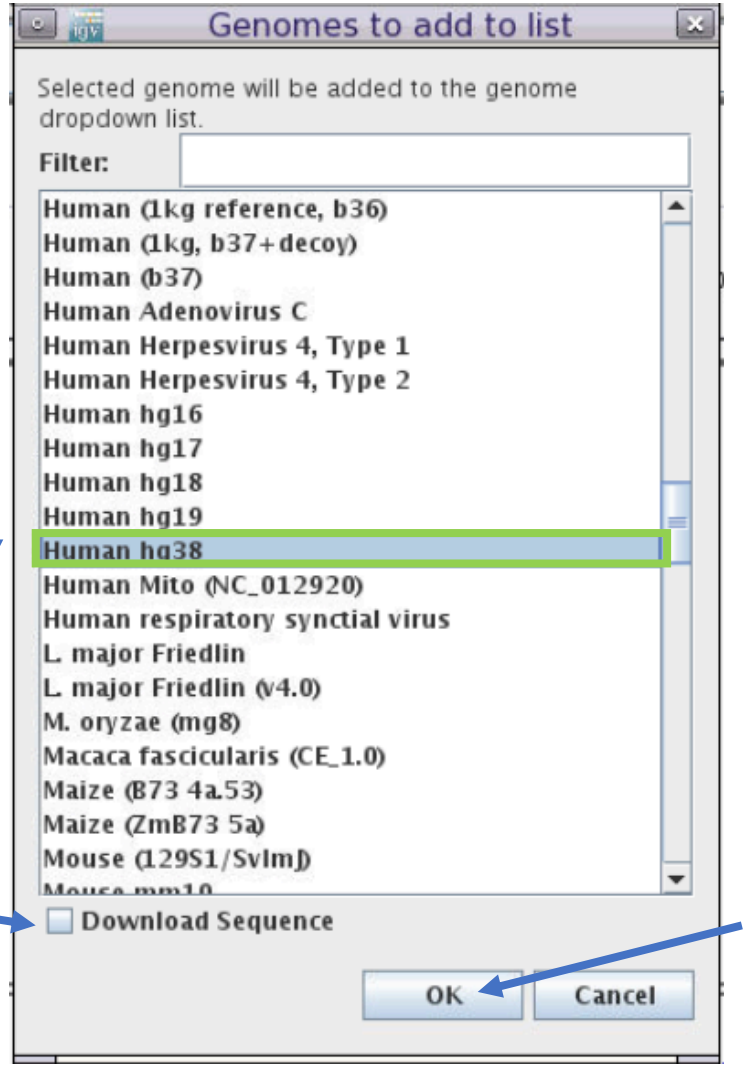


IGV: Load reference genome

1. Choose reference genome by clicking “Genomes” -> “Load Genome From Server…”



2. Scroll down to “Human hg38”

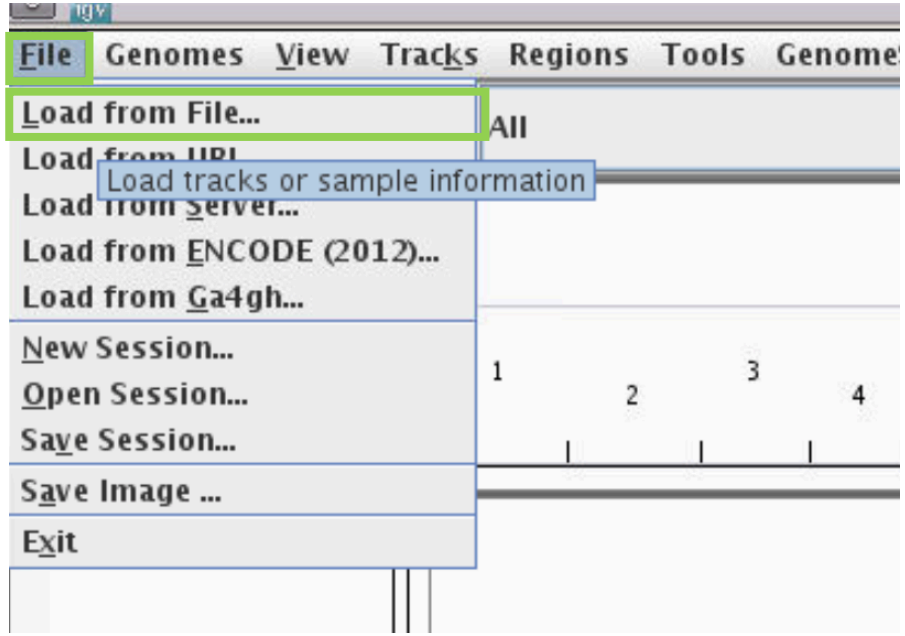


3. **DO NOT** click “Download Sequence”

4. Click “OK”

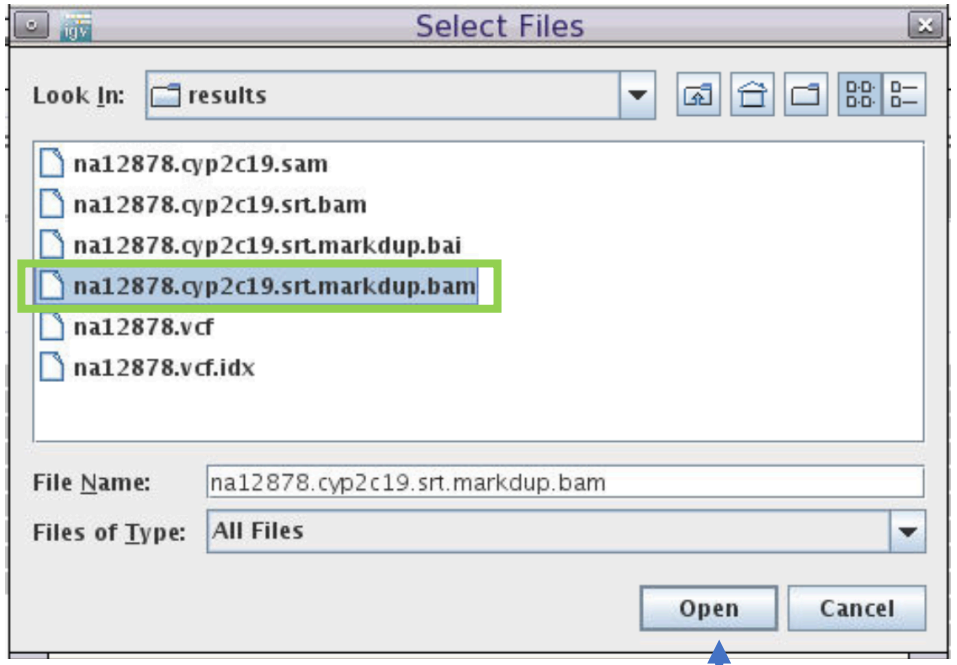
IGV: Load BAM file

1. Choose File -> Load from File->



2. Navigate to results folder in course directory:
E.g. /cluster/home/your-user-name/intro-to-ngs/results

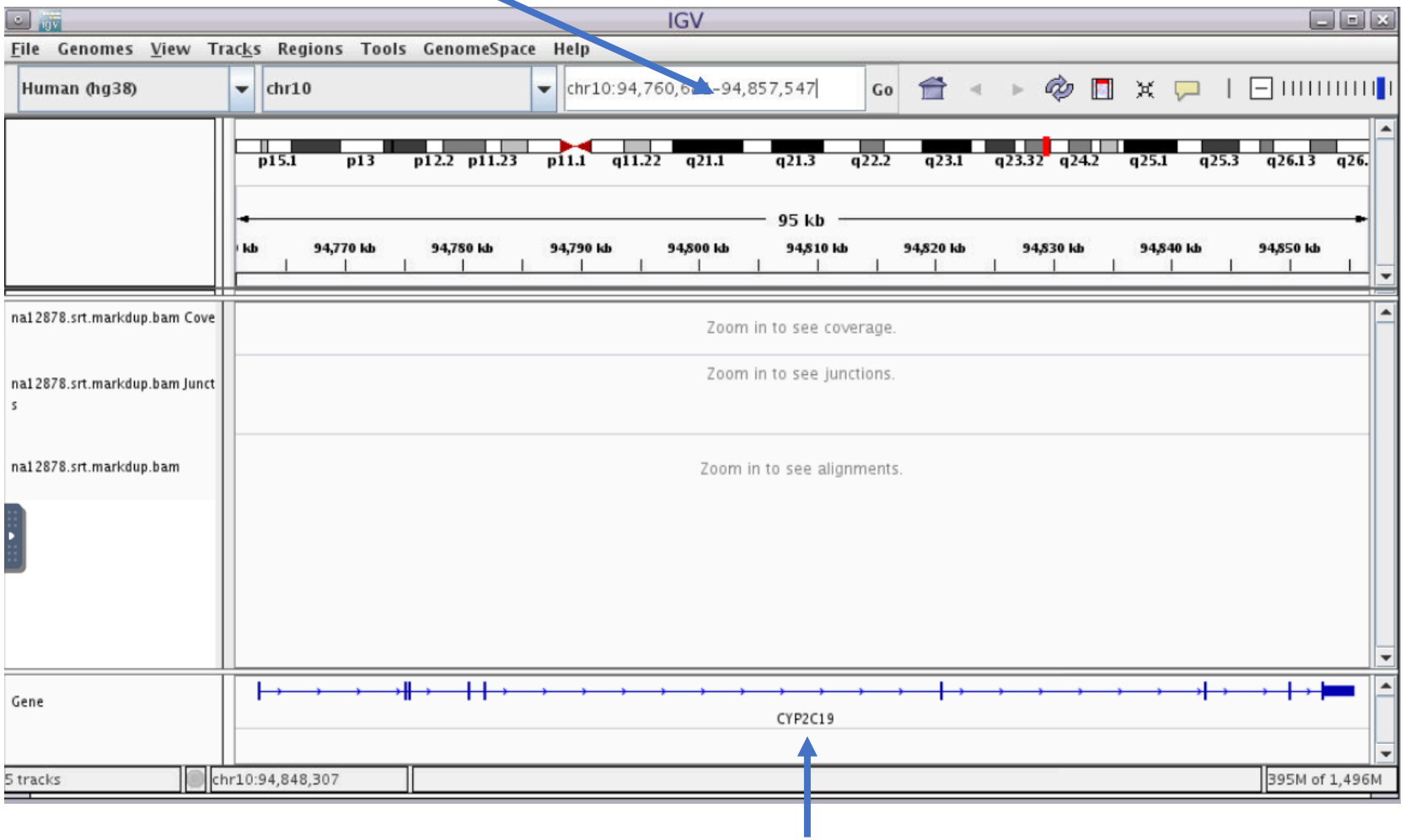
3. Select na12878.srt.markdup.bam



4. Click Open

IGV: Look at gene

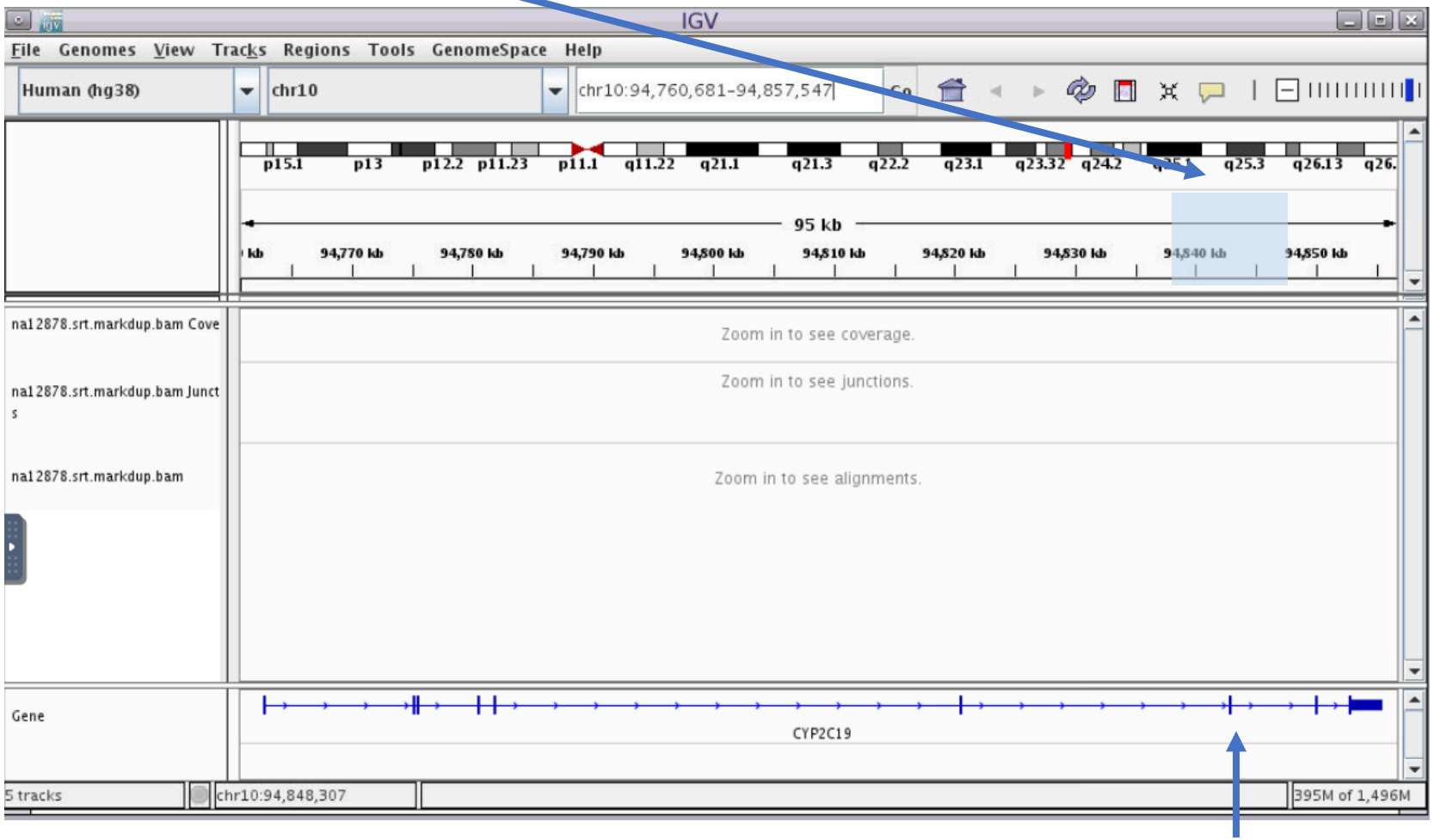
1. In the box type gene name “Cyp2c19” and hit enter



2. You will see the gene model display in the “RefSeq Genes” track, showing vertical bars where exons are located

IGV: zoom in

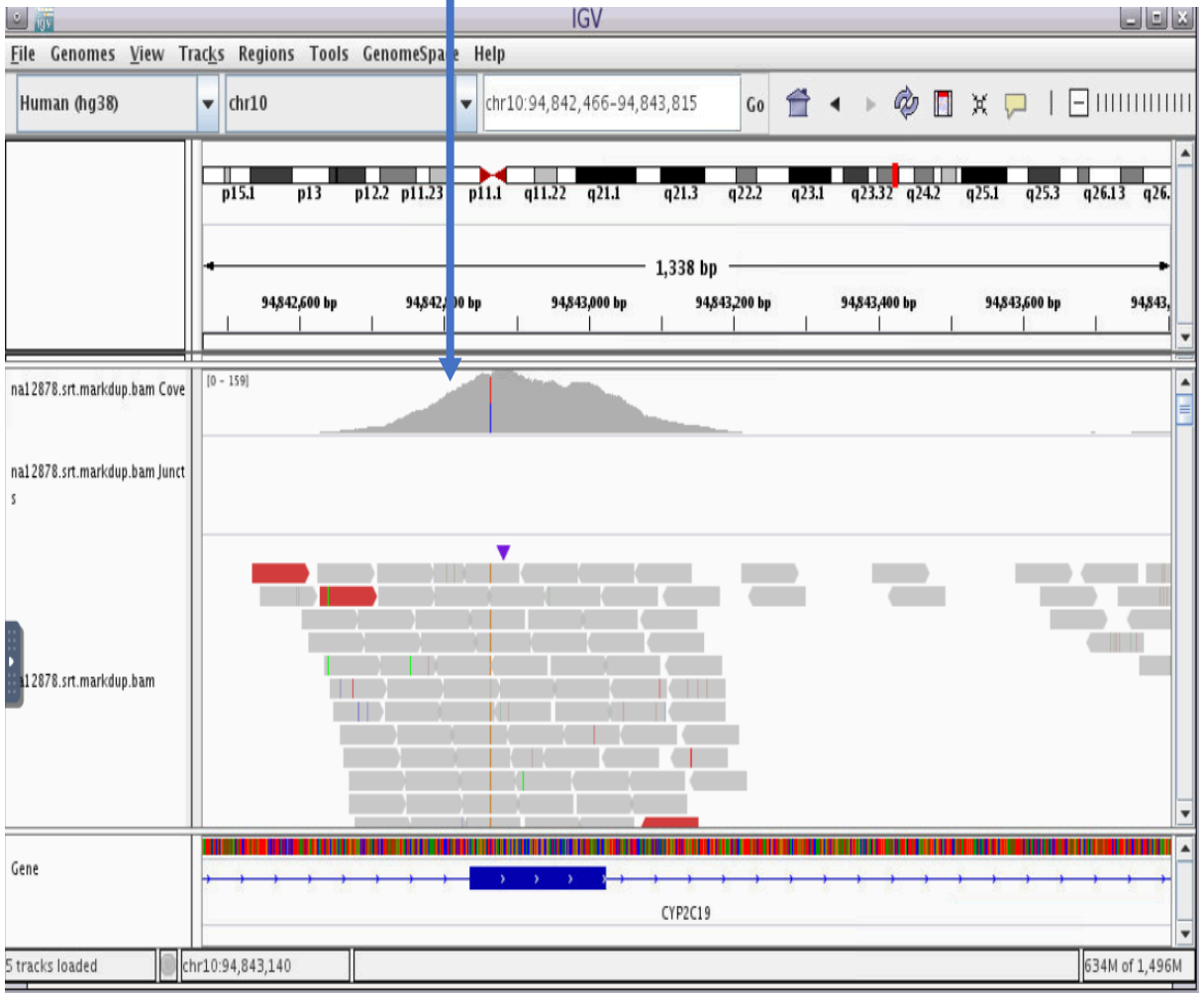
1. Click and drag over a region in the scale bar to zoom in on exon 7



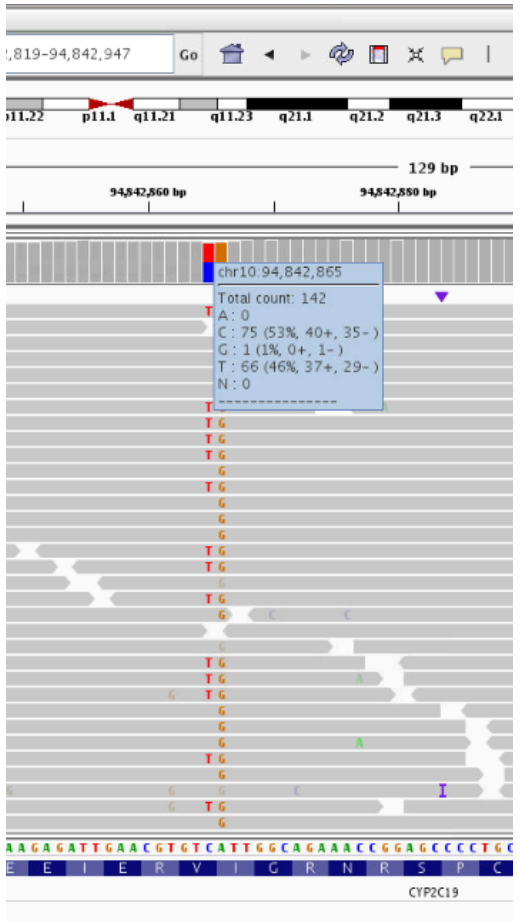
Exon 7: hover with mouse to confirm

IGV: reads pile up around exons

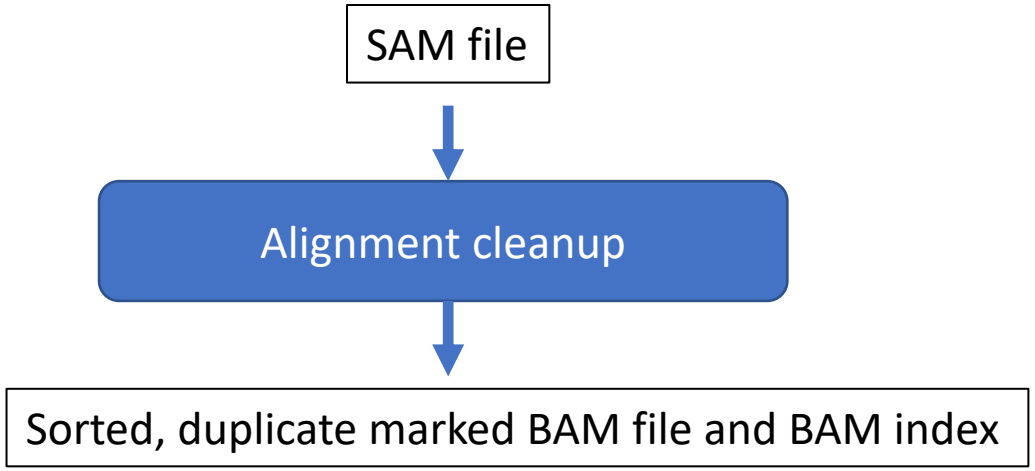
There appears to be a **variant** in this **exon**



Hover for more detail:

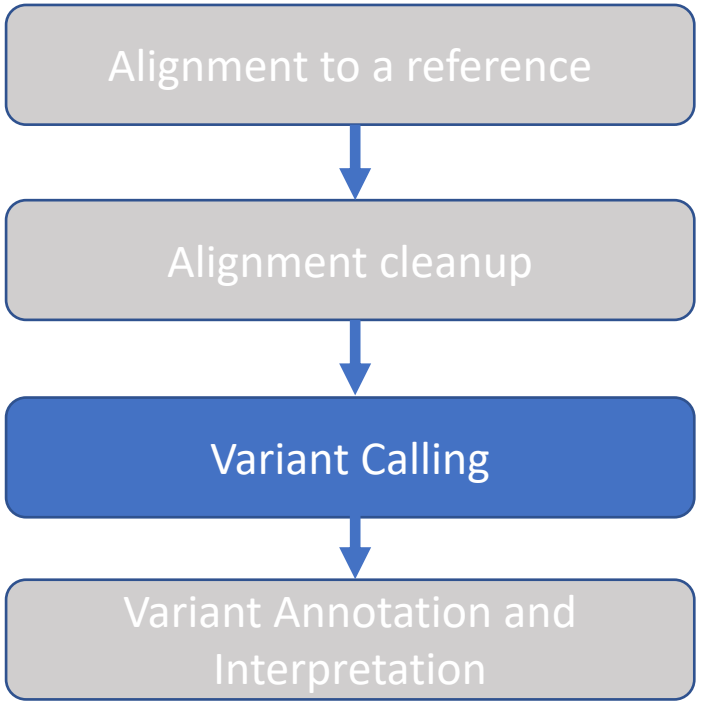


Alignment Cleanup summary



https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

Variant Calling step



https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

Prepare the reference sequence for GATK

Let's open a new script:

```
nano prepare.sh
```

Add these lines:

```
module load samtools/1.9
module load picard/2.8.0

samtools faidx ref_data/chr10.fa

picard CreateSequenceDictionary \
REFERENCE=ref_data/chr10.fa \
OUTPUT=ref_data/chr10.dict
```

Run our script

```
sh prepare.sh
```

```
[Tue Nov 26 15:41:09 EST 2019]
picard.sam.CreateSequenceDictionary
REFERENCE=ref_data/chr10.fa
 …
Runtime.totalMemory()=2058354688
```

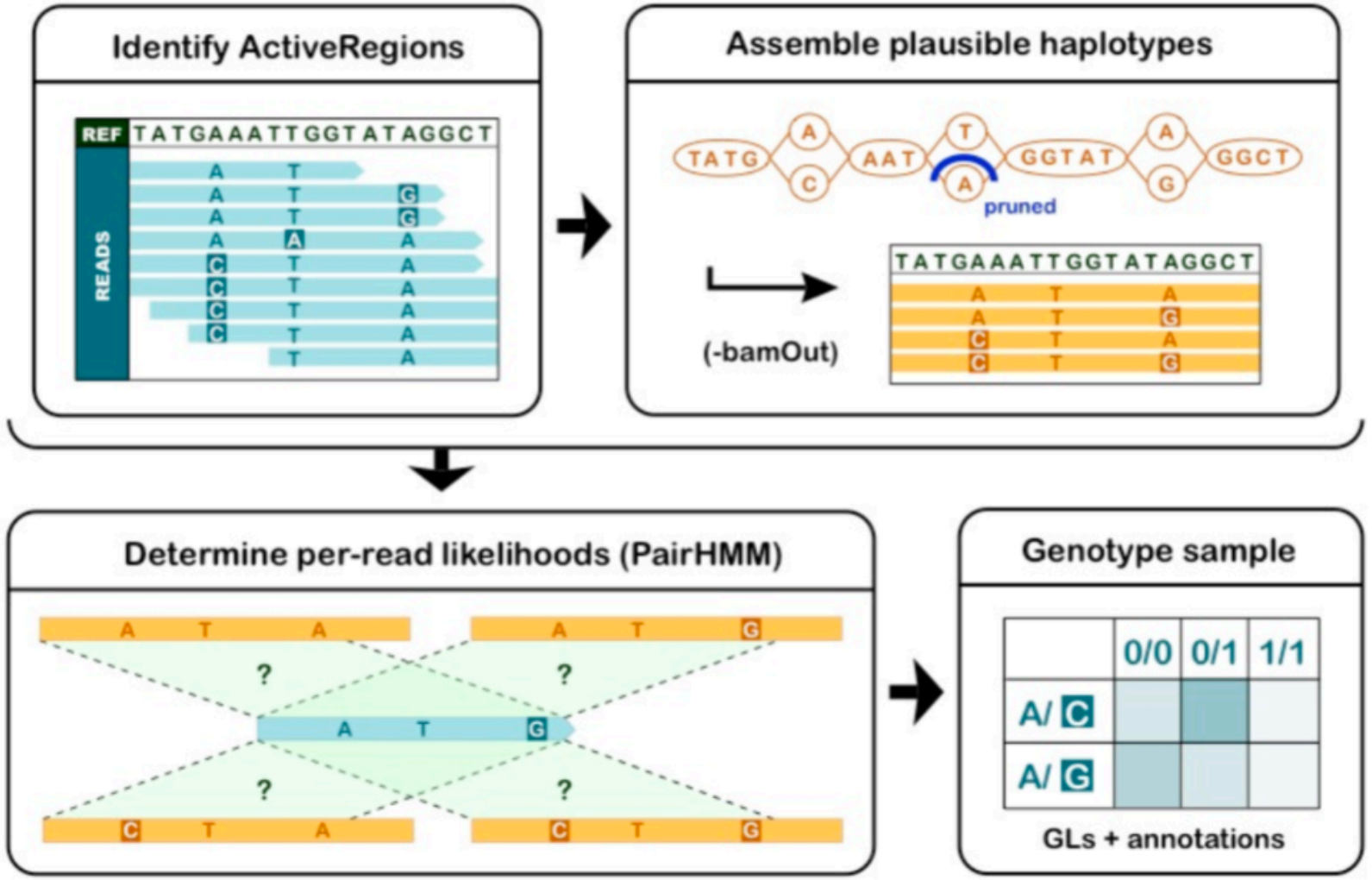
Two new files in ref_data:

chr10.fa.fai
chr10.dict

Variant calling using GATK HaplotypeCaller

Goals:

- Separate true variants from sequencing errors
- Establish which variants co-exist on single DNA strand (haplotype)



https://software.broadinstitute.org/gatk/documentation/tooldocs/3.8-0/org_broadinstitute_gatk_tools_walkers_haplotypecaller_HaplotypeCaller.php

Run GATK on our BAM

Let's see how to run GATK:

```
module load GATK/3.7
```

```
gatk --help
```

```
…
usage: java -jar GenomeAnalysisTK.jar -T <analysis_type> …
…
 haplotypecaller
   HaplotypeCaller          Call germline SNPs and indels via local re-assembly of haplotypes
   HaplotypeResolver        Haplotype-based resolution of variants in separate callsets.
```

Get tool specific help:

```
gatk -T HaplotypeCaller --help
```

Run GATK on our BAM

Let's open up a new script:

```
nano gatk.sh
```

Add these lines:

```
module load GATK/3.7

gatk -T HaplotypeCaller \
-R ref_data/chr10.fa \
-I results/na12878.srt.markdup.bam \
-o results/na12878.vcf
```

Run our script:

```
sh gatk.sh
```

```
INFO  17:17:41,656 HelpFormatter - ----------
INFO  17:17:41,660 HelpFormatter - The Genome Analysis Toolkit
(GATK) v3.7-0-gcfedb67, Compiled 2016/12/12 11:21:18
…
```

Two new files have appeared in our results folder:

```
ls results
```

```
na12878.vcf
na12878.vcf.idx
```

Variant Call Files (VCF)

Take a look at our VCF:

```
cd results
```

```
head na12878.vcf
```

Header lines

```
##fileformat=VCFv4.2
##FILTER=<ID=LowQual,Description="Low quality">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth …
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes …
##GATKCommandLine.HaplotypeCaller=<ID=HaplotypeCaller,...
….
##contig=<ID=chr10,length=135534747>
##reference=file:///cluster/home/tutln01/intro-to-ngs/ref_data/chr10.fa
#CHROM    POS        ID        REF       ALT       QUAL      FILTER    INFO              FORMAT          NA12878
chr10     96521422   .         A         G         60.28     .         AC=2;AF=1.00; ….  GT:AD:DP:GQ:PL  1/1:0,3:3:9:88,9,0
chr10     96522365   .         T         C         1134.77   .         AC=1;AF=0.500;….  GT:AD:DP:GQ:PL  0/1:47,37:84:99:1163,0,1502
```

VCF Quality Control

How did our Variant Calling perform?

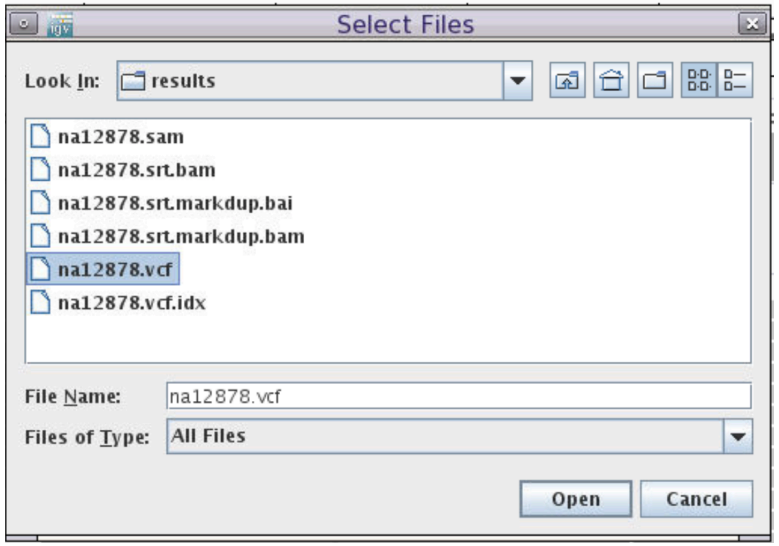
- Transition/Transversion ratio
- Fraction of called variants that are ‘known’, e.g. exist in database dbSNP
- Comparing with NIST NA12878 ‘gold standard’callsets
- …

https://gatkforums.broadinstitute.org/gatk/discussion/6308/evaluating-the-quality-of-a-variant-callset

IGV: Add our VCF to IGV

Go to back to IGV on demand

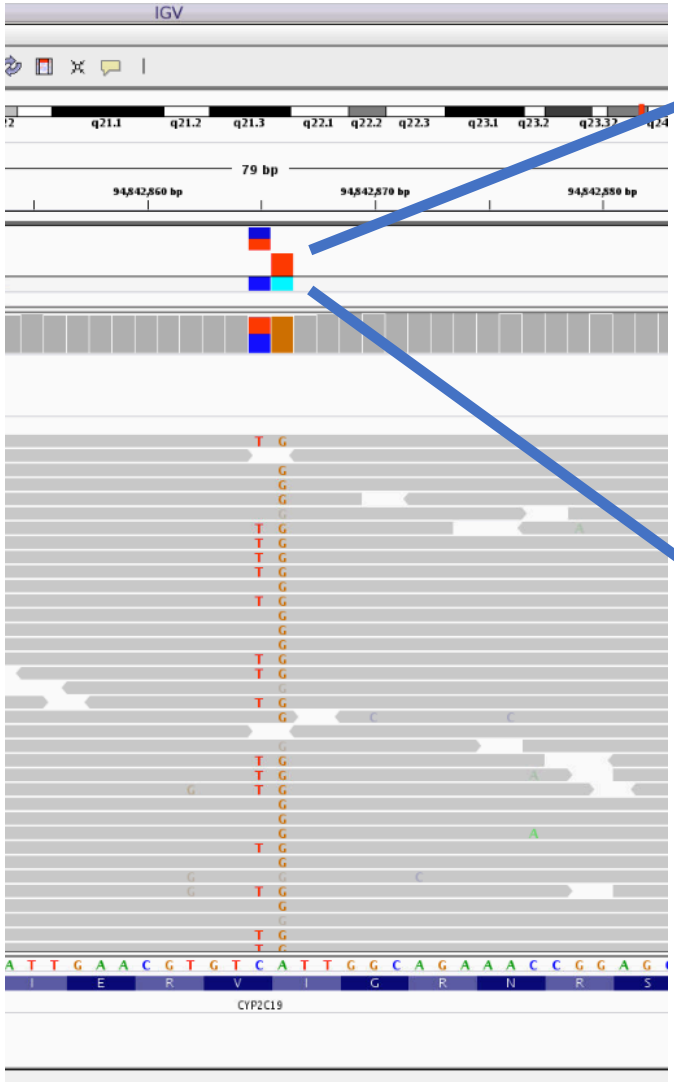
File-> Load from File and select the na12878.vcf



Hover to explore two tracks:

Variant track

Genotype track



Chr: chr10
Position: 94842866
ID: .
Reference: A*
Alternate: G
Qual: 4821.77
Type: SNP
Is Filtered Out: No

Alleles:
Alternate Alleles: G
Allele Count: 2
Total # Alleles: 2
Allele Frequency: 1.0

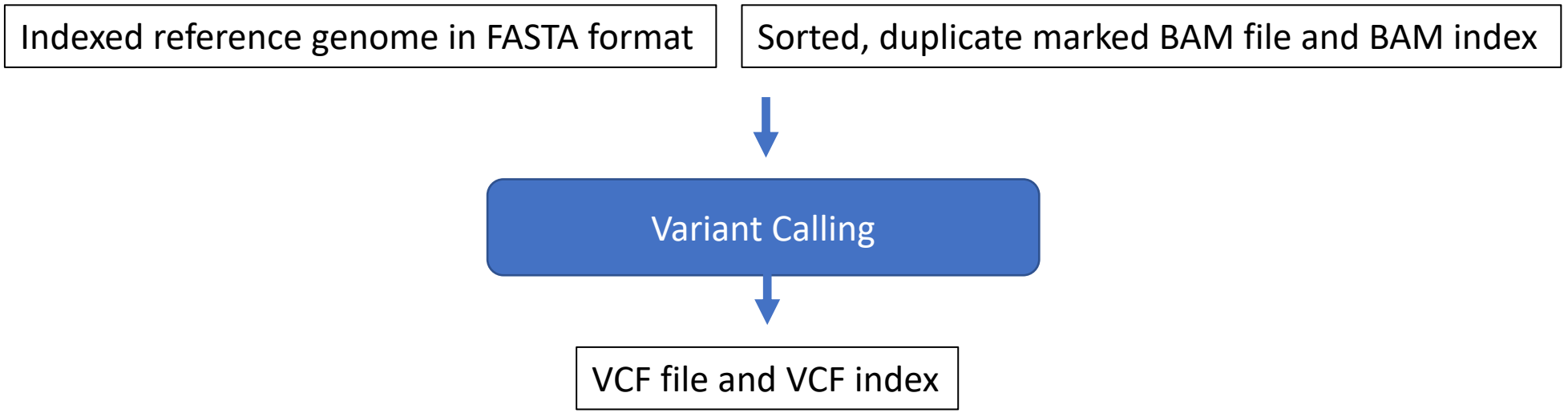
Variant Attributes
Allele Frequency: 1.00
Allele Count: 2
QD: 30.63
Mapping Quality: 59.69
SOR: 0.740
MLEAC: 2
ExcessHet: 3.0103
MLEAF: 1.00
Depth: 133
Total Alleles: 2
FS: 0.000

Chr: chr10
Position: 94842865
ID: .

Genotype Information
Sample: NA12878
Genotype: C/T
Quality: 99
Type: HET
Is Filtered Out: No

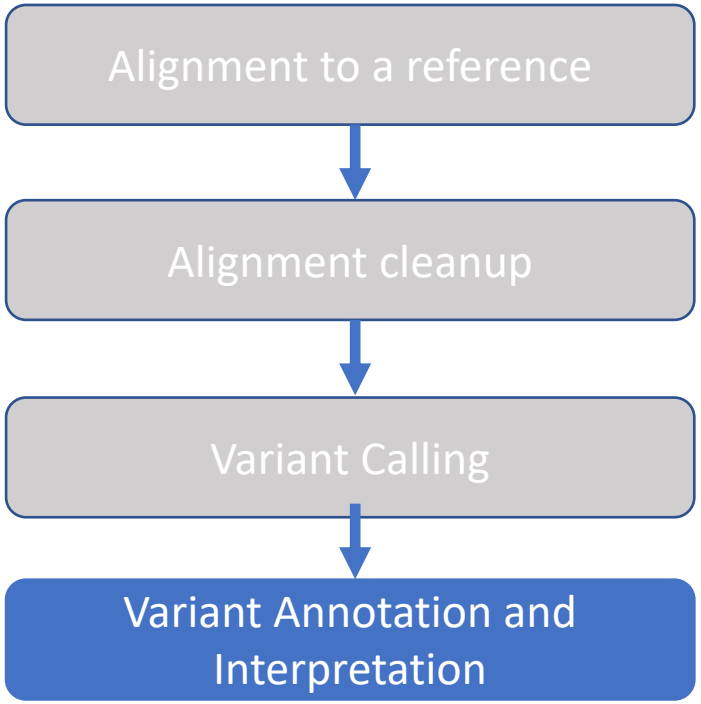
Genotype Attributes
AD: 63,64
Genotype Quality: 99
Depth: 127
PL: 1909,0,1650

Variant Calling Summary



https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

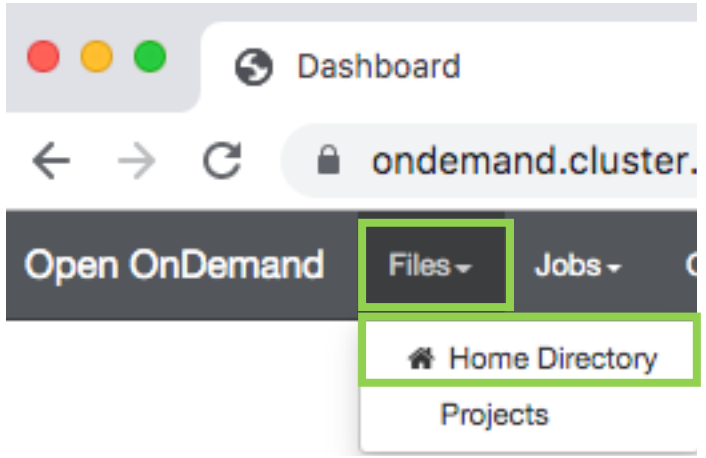
Variant Annotation and Interpretation



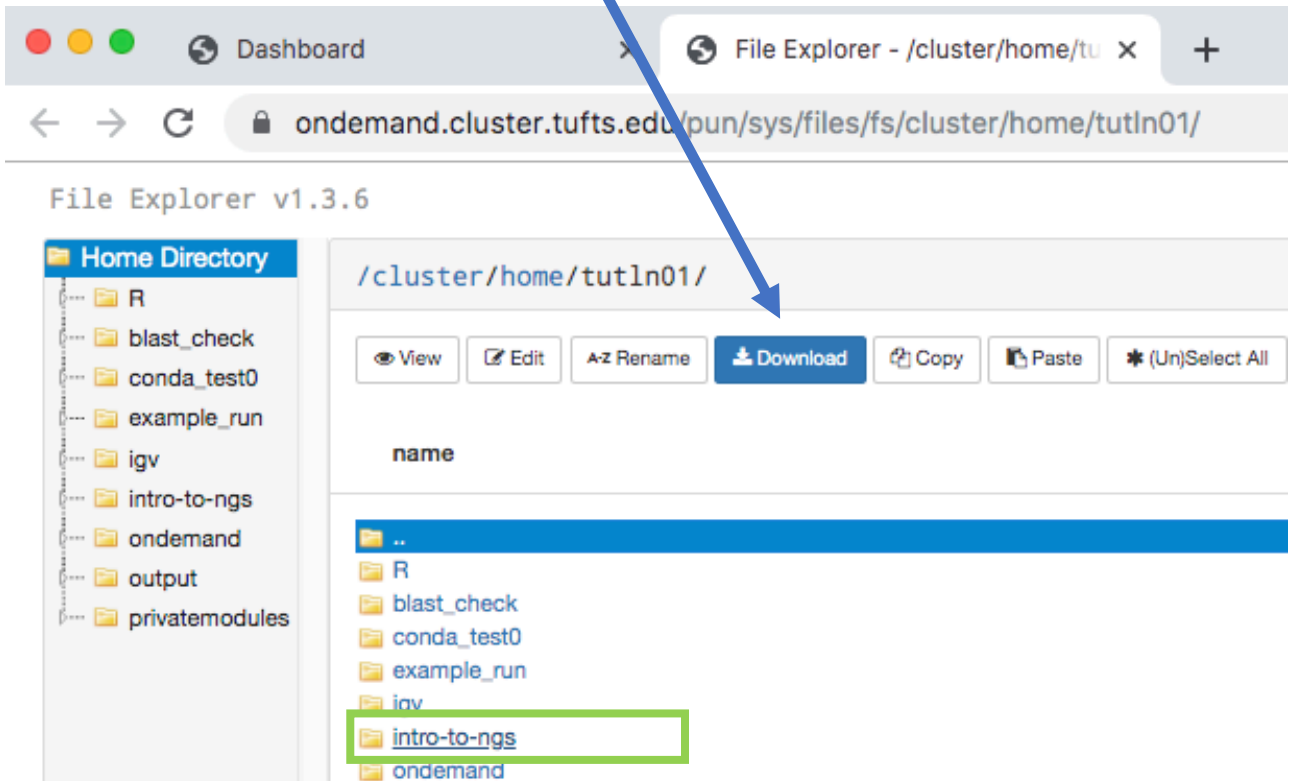
https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

Download the VCF

1. Go back to ondemand.cluster.tufts.edu
2. Click **Files**-> **Home Directory**



3. Select **intro-to-ngs->results->na12878.vcf**
4. Click Download



Variant Effect Predictor (VEP)

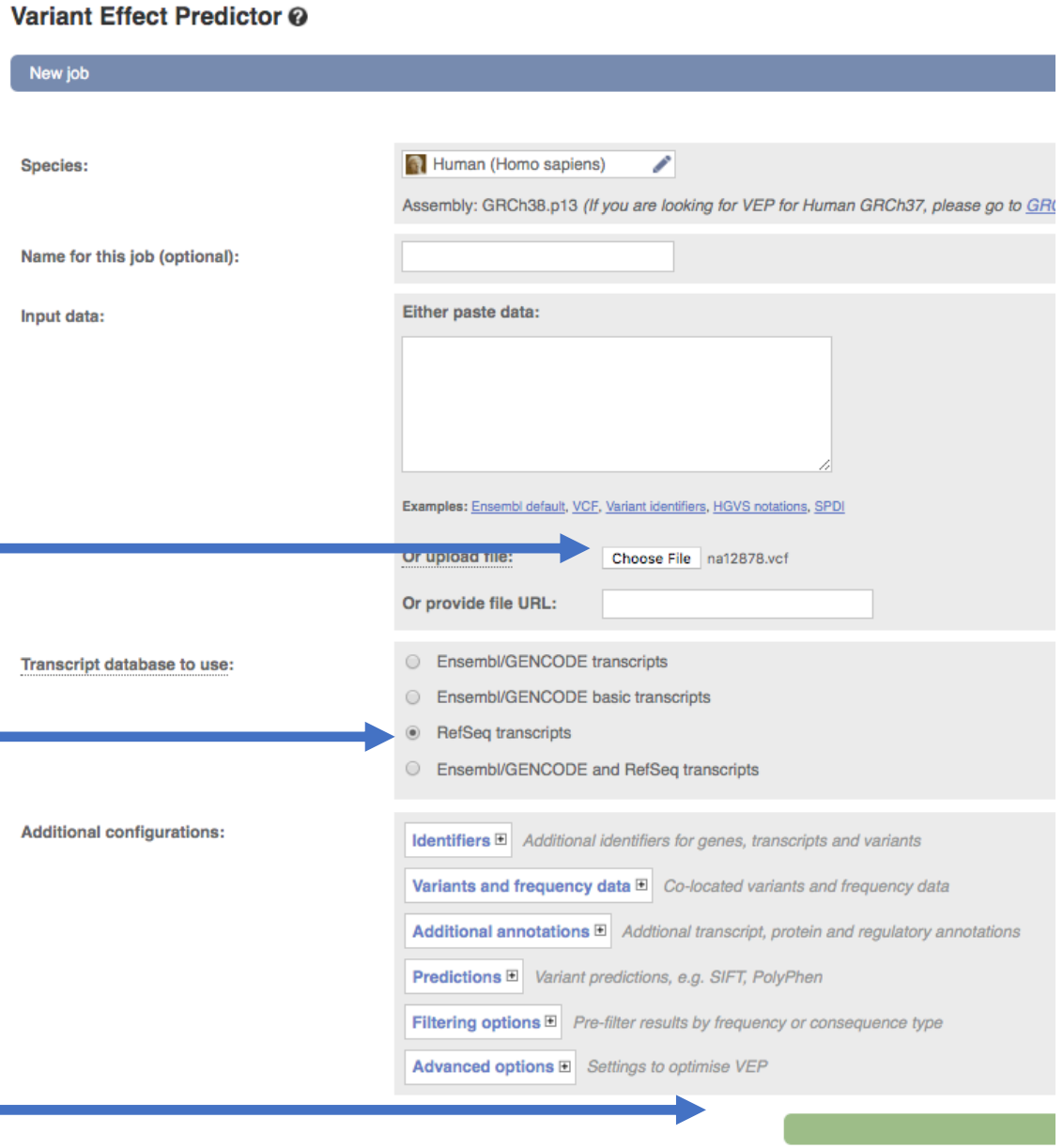
1. Point a web browser to* :
https://useast.ensembl.org/Tools/VEP

* can also be run on HPC command line

2. Choose file na12878.vcf

3. Choose RefSeq transcripts

4. Click Run



Viewing VEP results

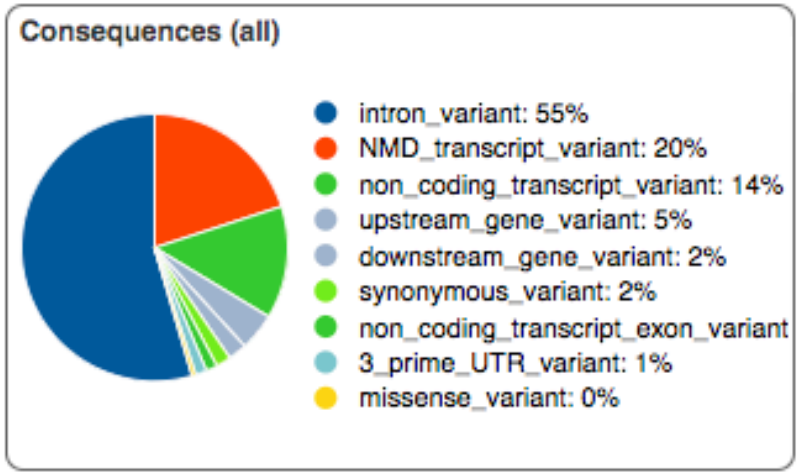
When your job is done, click "**View Results**"

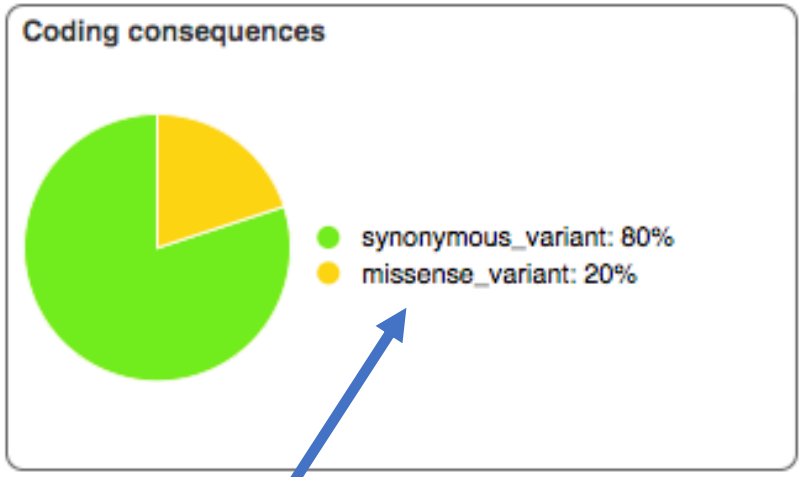
Variant Effect Predictor results

Job details

Summary statistics

Category	Count
Variants processed	44
Variants filtered out	0
Novel / existing variants	4 (9.1) / 40 (90.9)
Overlapped genes	3
Overlapped transcripts	5
Overlapped regulatory features	0





missense_variants change the amino acid
Let's investigate!

Filtering VEP consequences

Under **Filters** choose **Consequence + is + missense_variant** and click **Add**

You should see 1 row – here are a subset of interesting columns:

Location	Allele	Consequence	IMPACT	SYMBOL	BIOTYPE	Amino_acids
10:94842866-94842866	G	missense_variant	MODERATE	CYP2C19	protein_coding	I/V

Existing_variation	SIFT	PolyPhen	AF
rs3758581,CM983294	tolerated(0.38)	benign(0.05)	0.9515

Interpreting a variant

Under **Filters** choose **Consequence + is + missense_variant** and click **Add**

You should see 1 row – here are a subset of interesting columns:

Location	Allele	Consequence	IMPACT	SYMBOL	BIOTYPE	Amino_acids
10:94842866-94842866	G	missense_variant	MODERATE	CYP2C19	protein_coding	I/V

Some examples of how VEP matches transcript consequence to IMPACT:

Consequence	IMPACT
stop_gained	HIGH
missense_variant	MODERATE
synonymous_variant	LOW
upstream_gene_variant	MODIFIER

https://useast.ensembl.org/info/genome/variation/prediction/predicted_data.html

Interpreting a variant

Under **Filters** choose **Consequence + is + missense_variant** and click **Add**

You should see 1 row – here are a subset of interesting columns:

Location	Allele	Consequence	IMPACT	SYMBOL	BIOTYPE	Amino_acids
10:94842866-94842866	G	missense_variant	MODERATE	CYP2C19	protein_coding	I/V

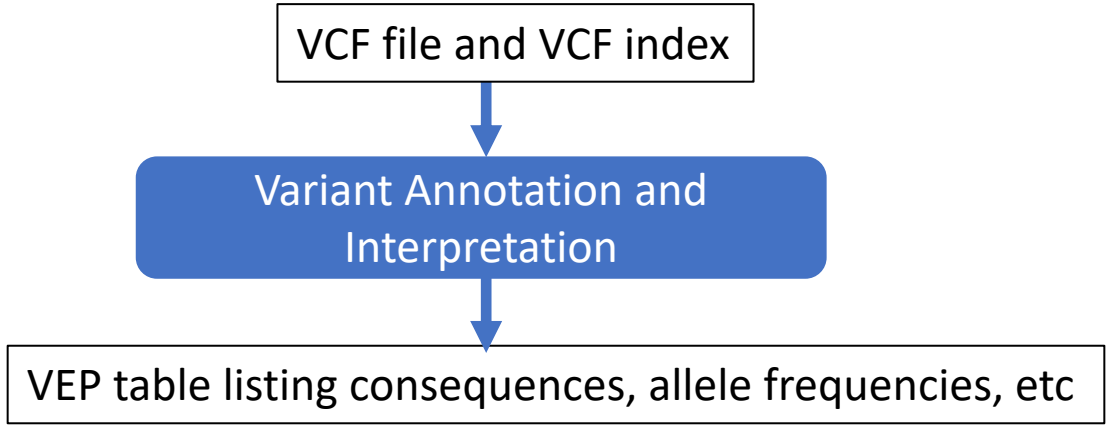
Existing_variation	SIFT	PolyPhen	AF
rs3758581,CM983294	tolerated(0.38)	benign(0.05)	0.9515

Allele Frequency (AF) shows that this substation is common in the global population – which suggests it is not pathogenic.

SIFT, PolyPhen are computational algorithms to predict the effect on protein function – both suggest this protein function will not be altered

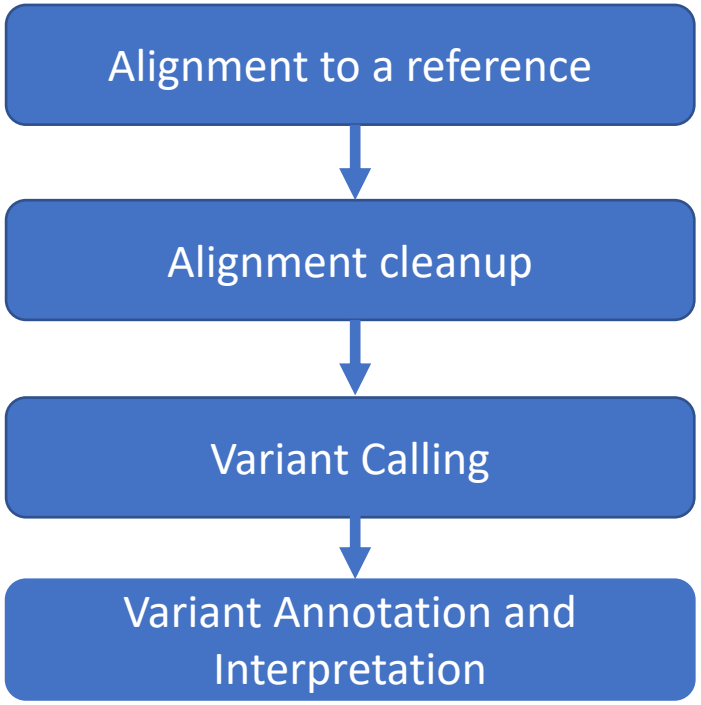
https://useast.ensembl.org/info/genome/variation/prediction/protein_function.html

Annotation Summary

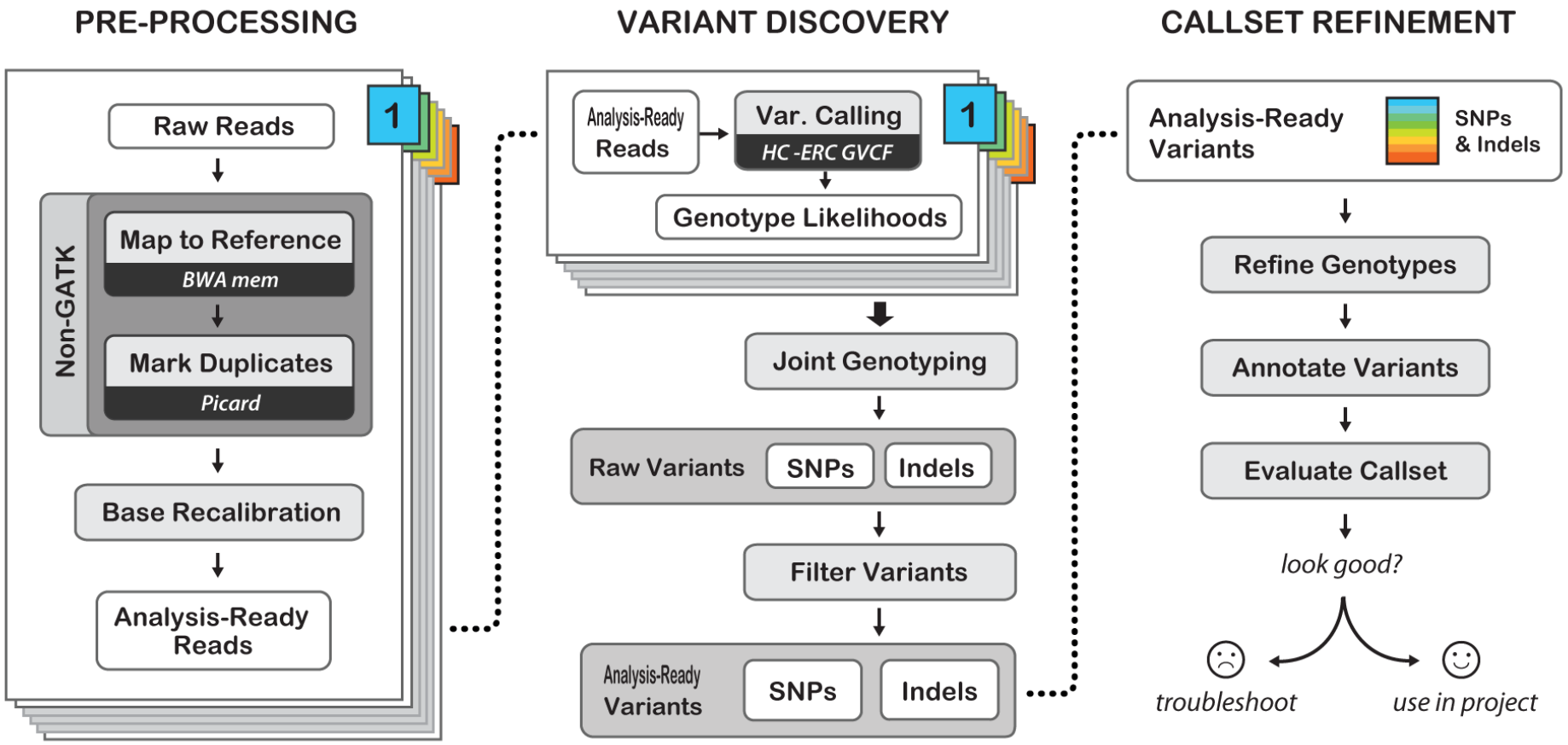


https://github.com/hbctraining/In-depth-NGS-Data-Analysis-Course

Workflow Summary



Variant Calling workflow – GATK best practices



Best Practices for Germline SNPs and Indels in Whole Genomes and Exomes - June 2016

https://software.broadinstitute.org/gatk/best-practices/

Thank you

Especially to:
Shawn Doughty, Research Computing Manager, TTS
Delilah Maloney, High Performance Computing Specialist, TTS
Susi Remondi, Senior Technical Training Specialist, TTS

For more tutorials like these on doing Bioinformatics on the Tufts HPC cluster:
https://sites.tufts.edu/biotools/tutorials/

For more great bioinformatics tutorials:
https://github.com/hbctraining/

For questions on Bioinformatics or the Tufts HPC, contact tts-research@tufts.edu