

Human Genomics and Epigenomics

Practical 1 – 18/01/2021

Practical 2 – 19/01/2021

Practical 3 – 25/01/2021

Practical 4 – 26/01/2021

Prof. Massimo Delledonne
Functional Genomics lab

ALIGNMENT AND VARIANT CALLING

1° Day (3h): Pre-processing of raw reads

- The fastq file
- Quality control of fastq files
- Adapter removing and trimming of fastq files
 - Sickle and scythe
 - Trimmomatic
- Reads alignment:
 - The human reference genome (hg19 and hg38, main differences)
 - The BAM file

2° Day (3h): Alignment

- Alignment of trimmed reads to the reference genome
 - BWA-mem
 - Isaac2 pipeline
- Duplicates removal
- Read Clipping
- Visualization of aligned reads on IGV

ALIGNMENT AND VARIANT CALLING

3° Day (3h): Statistics and Variant Calling

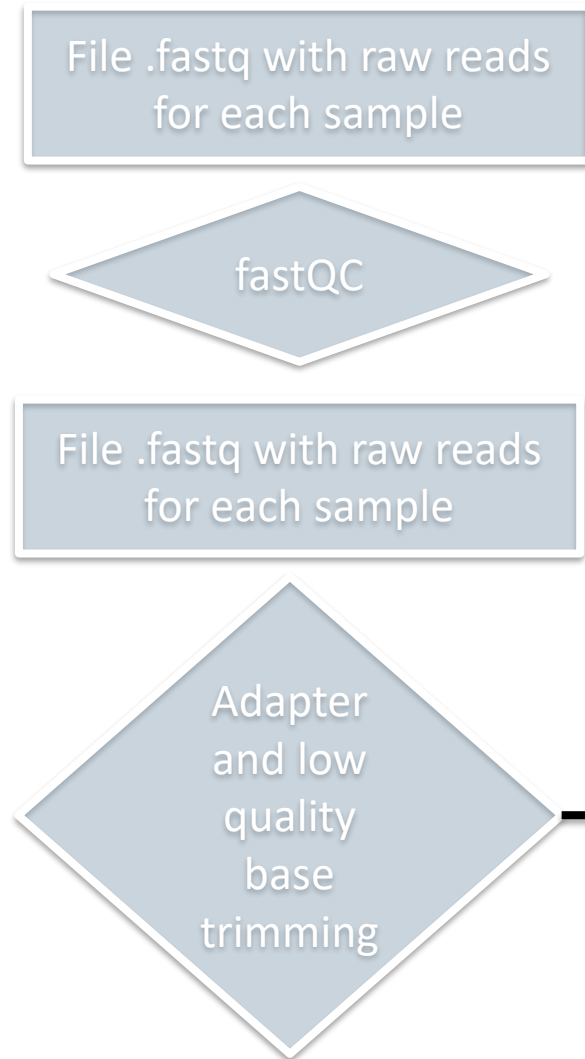
- Statistics on reads alignment: main parameters for the evaluation of NGS data
 - Average coverage and uniformity
 - Fold enrichment (on/near/off target)
 - Genotypability (mapping quality besides coverage)
- Variant calling:
 - The VCF and gVCF files
 - Germline variant calling
 - GATK4 Best practice pipeline

4° Day (3h): Variant Calling

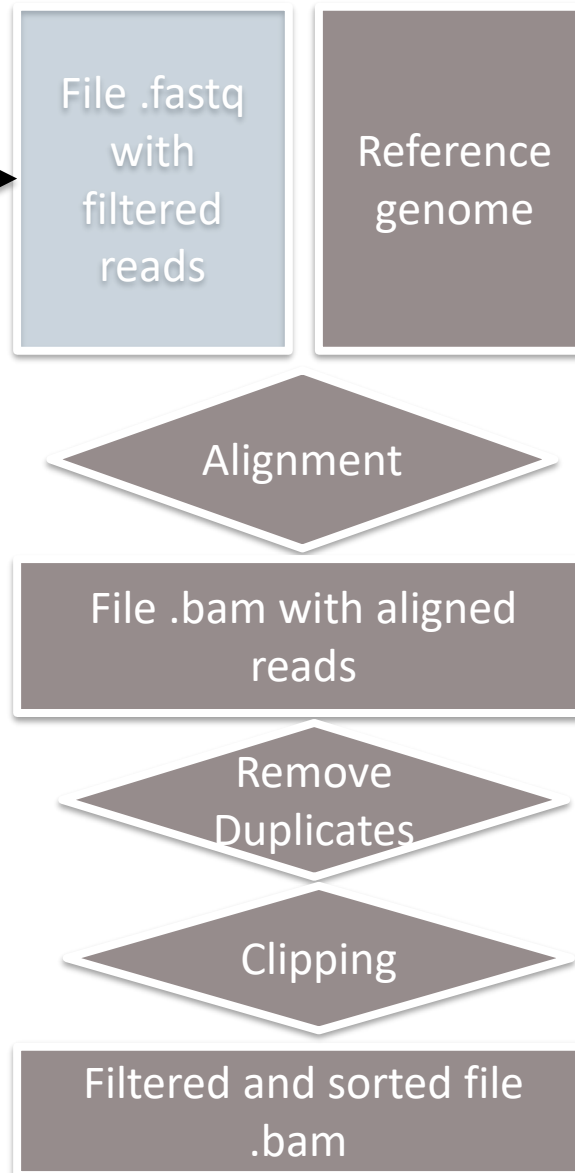
- Germline variant calling
 - GATK4 Best practice pipeline
 - Strelka2
- Visualization of genetic variants on IGV
- CNV detection

Pipeline

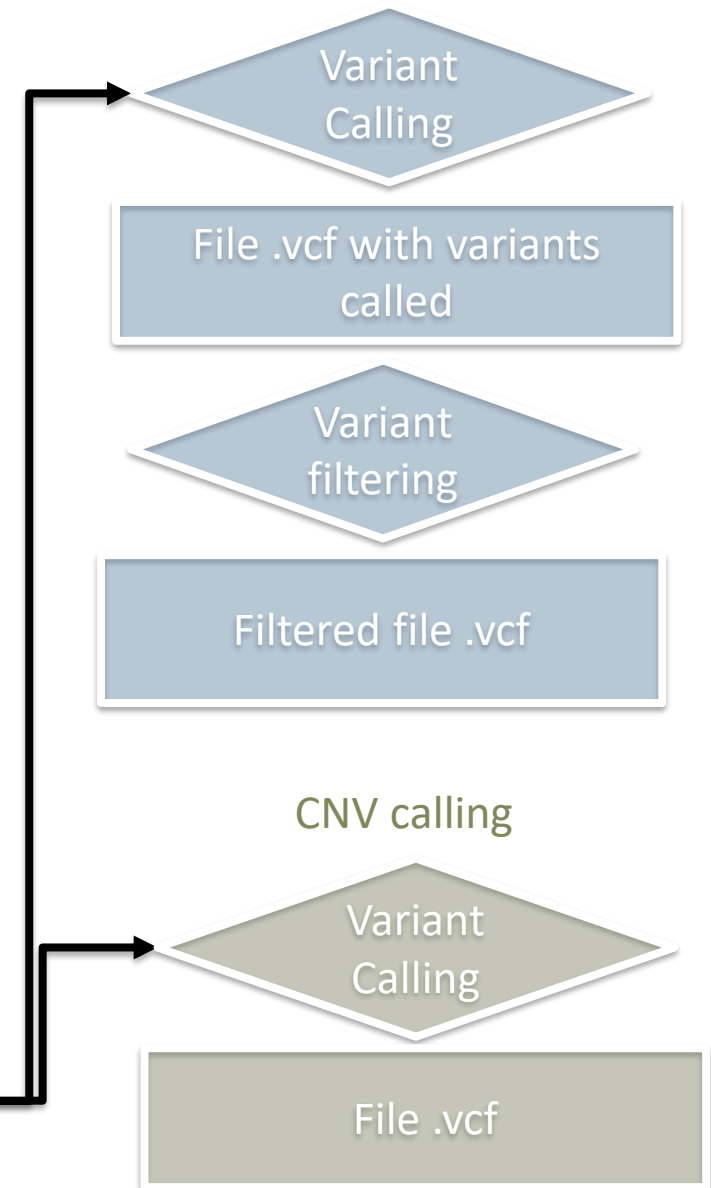
Data QC & Filtering



Alignment



Variant Calling




CNV calling

ALIGNMENT OF TRIMMED READS TO THE REFERENCE GENOME

BWA

- BWA is a software package for mapping low-divergent sequences against a large reference genome.
- Three different algorithms:
 - MEM
 - SW
 - backtrack

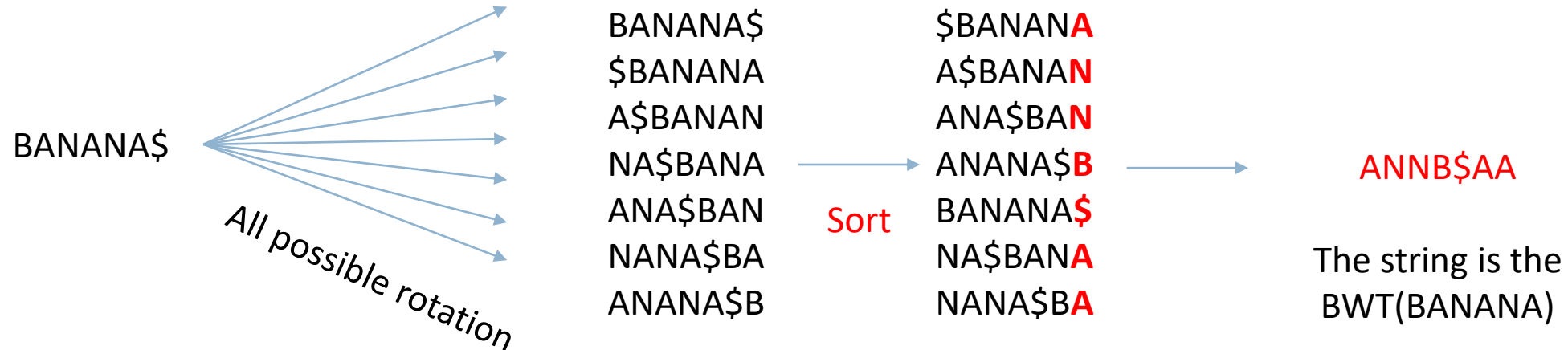
BWA

- BWA is a software package for mapping low-divergent sequences against a large reference genome.
- Three different algorithms:
 - **MEM**  Fast and accurate short read alignment with Burrows-Wheeler transform (2009)
Heng Li, Richard Durbin
DOI: [10.1093/bioinformatics/btp324](https://doi.org/10.1093/bioinformatics/btp324)
 - SW
 - backtrack

BWA-mem

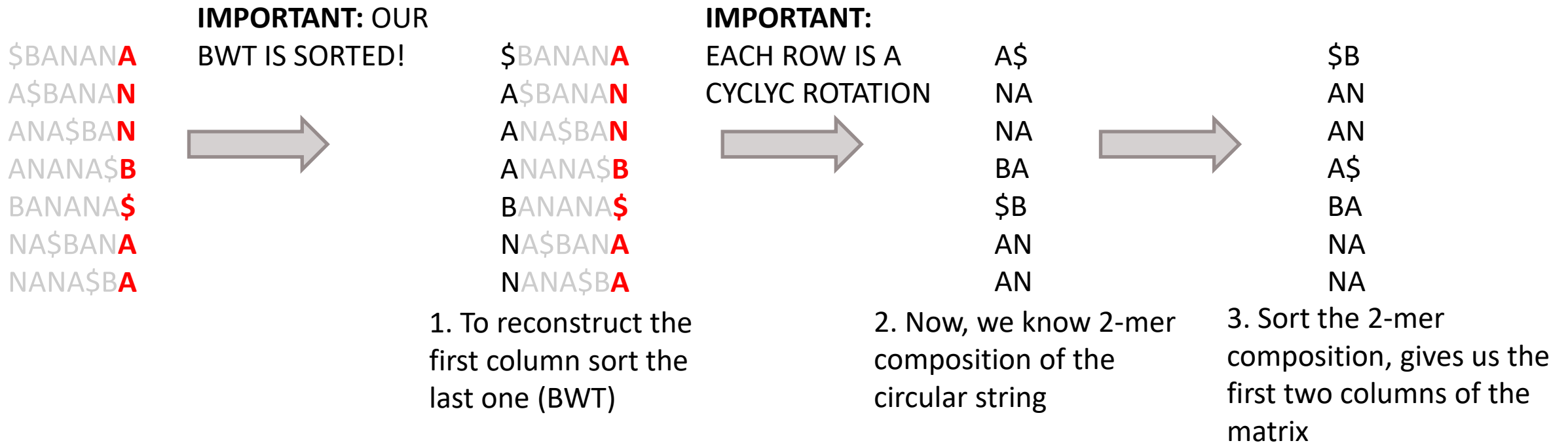
BWA-mem is one of the most popular NGS aligners. Based on the Burrow Wheeler transform algorithm, BWA produces very accurate alignments. **It is used especially for DNA sequencing data: targeting sequencing, WES, WGS. It breaks reads up to 100bp. BWA soft-clip unmapped bases, retaining only reads with minimum 19bp mapped.**

Burrow Wheeler Transform



BWA-mem: Reconstruct «genome»

We have only the BWT and we want to reconstruct our reference sequence from the BWT(BANANA) = **ANNB\$AA**



BWA-mem: Reconstruct «genome»

Now as before...

\$BANANA
A\$BANAN
ANANAS\$
ANANAS\$
BANANAS
NANAS\$
NANAS\$

IMPORTANT:
EACH ROW IS A
CYCLIC ROTATION



A\$B
NA\$
NAN
BAN
\$BA
ANA
ANA



\$BA
A\$B
ANA
ANA
BAN
NA\$
NAN



...and repeat
again, till you
reconstruct the
full matrix

4. Now, we know 3-mer
composition of the
circular string

5. Sort the 3-mer composition,
gives us the first three columns
of the matrix



\$BANANA
A\$BANAN
ANANAS\$
ANANAS\$
BANANAS
NANAS\$
NANAS\$

6. We assume that «\$» is the
lexicographically first, everything
after must be the original sequence

BWA-mem: Reconstruct «genome» faster

For each letter a number indicating the occurrence is given.

We know that «\$» is the starting point.

\$₁BANAN A₁
A₁\$BANAN N₁
A₂NA\$BAN N₂
A₃NANA\$ B₁
B₁ANANA\$ S₁
N₁A\$BAN A₂
N₂ANA\$B A₃

So we look at the character in the last column.



A\$

\$₁BANAN A₁
A₁\$BANAN N₁
A₂NA\$BAN N₂
A₃NANA\$ B₁
B₁ANANA\$ S₁
N₁A\$BAN A₂
N₂ANA\$B A₃

In this case the occurrence of the A is 3, so we go to the corresponding letter in the first column

NA\$

\$₁BANAN A₁
A₁\$BANAN N₁
A₂NA\$BAN N₂
A₃NANA\$ B₁
B₁ANANA\$ S₁
N₁A\$BAN A₂
N₂ANA\$B A₃

ANA\$

\$₁BANAN A₁
A₁\$BANAN N₁
A₂NA\$BAN N₂
A₃NANA\$ B₁
B₁ANANA\$ S₁
N₁A\$BAN A₂
N₂ANA\$B A₃

NANA\$

\$₁BANAN A₁
A₁\$BANAN N₁
A₂NA\$BAN N₂
A₃NANA\$ B₁
B₁ANANA\$ S₁
N₁A\$BAN A₂
N₂ANA\$B A₃

ANANA\$

\$₁BANAN A₁
A₁\$BANAN N₁
A₂NA\$BAN N₂
A₃NANA\$ B₁
B₁ANANA\$ S₁
N₁A\$BAN A₂
N₂ANA\$B A₃

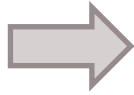
BANANA\$

\$₁BANAN A₁
A₁\$BANAN N₁
A₂NA\$BAN N₂
A₃NANA\$ B₁
B₁ANANA\$ S₁
N₁A\$BAN A₂
N₂ANA\$B A₃

BWA-mem: Search for a pattern

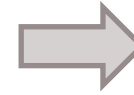
We want to search for «ANA»

\$₁BANAN**A**₁
A₁\$BANAN**N**₁
A₂NA\$BA**N**₂
A₃NANAN**A**\$_{B1}
B₁ANANAN**\$**₁
N₁A\$BAN**A**₂
N₂ANAN**A**\$_{B3}



\$₁BANAN**A**₁
A₁\$BANAN**N**₁
A₂NA\$BA**N**₂
A₃NANAN**A**\$_{B1}
B₁ANANAN**\$**₁
N₁A\$BAN**A**₂
N₂ANAN**A**\$_{B3}

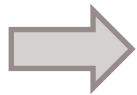
We have two
occurrences but we
don't know where
are the genome!



Suffix Array: each
row corresponds to
the path to the «\$»

1 2 3 4 5 6 7
B A N A N A \$

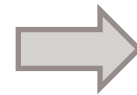
7 \$₁BANAN**A**₁
A₁\$BANAN**N**₁
A₂NA\$BA**N**₂
A₃NANAN**A**\$_{B1}
B₁ANANAN**\$**₁
N₁A\$BAN**A**₂
N₂ANAN**A**\$_{B3}



1 2 3 4 5 6 7
B A N A N A \$

7 \$₁BANAN**A**₁
6 **A₁\$BANAN****N**₁
A₂NA\$BA**N**₂
A₃NANAN**A**\$_{B1}
B₁ANANAN**\$**₁
N₁A\$BAN**A**₂
N₂ANAN**A**\$_{B3}

...



7 \$₁BANAN**A**₁
6 A₁\$BANAN**N**₁
4 A₂NA\$BA**N**₂
2 A₃NANAN**A**\$_{B1}
1 B₁ANANAN**\$**₁
5 N₁A\$BAN**A**₂
3 N₂ANAN**A**\$_{B3}

ANA occurs in
positions 4 and 2!

Connect to server

1. Enter in the server:

a. `ssh lessons@157.27.80.26`

b. Password: `lez2021`

2. Enter in the created folder: `cd HGE_2021/your_name`

Alignment with BWA command

- Align your reads to the reference genome (chr6 hg38):

```
/opt/bwa/bwa mem /home/lessons/HGE_2021/ref/chr6.hg38.fa  
trimmed1.fastq.gz trimmed2.fastq.gz > sample.bwa.sam
```

- Turn your file sam into file bam:

```
samtools view -bT /home/lessons/HGE_2021/ref/chr6.hg38.fa -o sample.bwa.bam  
sample.bwa.sam
```

Alignment with BWA command

- Sort your file:

```
samtools sort sample.bwa.bam -o sample.sorted.bwa.bam
```

- Create index for your bam file:

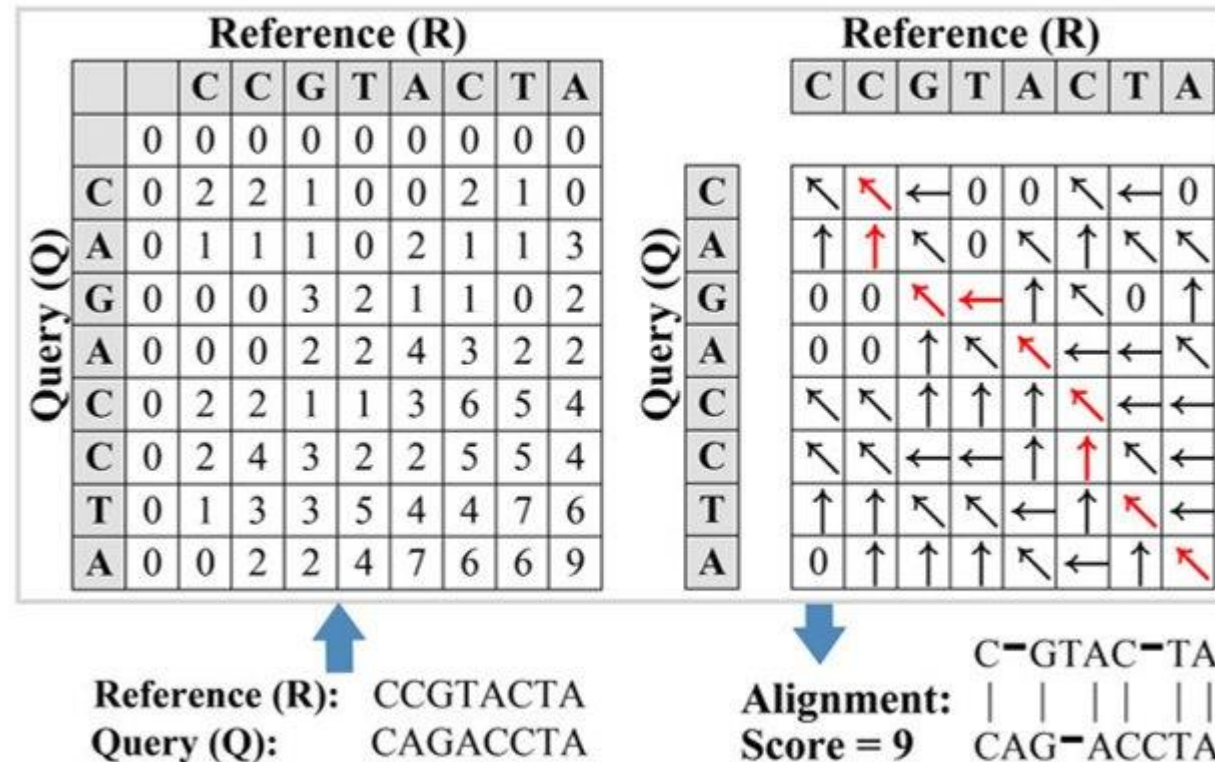
```
samtools index sample.sorted.bwa.bam
```

Isaac4

Isaac4 is a sequence alignment and variant detector tool developed by Illumina based on Smith-Waterman algorithm. Isaac employs a large amount of memory to produce ultrafast alignments and is reportedly 45 times faster than the BWA pipeline. It is used especially for DNA sequencing data: targeting sequencing, WES, WGS. It breaks reads up to 10,000bp. Isaac soft-clip unmapped bases and allows to soft-clip overlapping reads using the option --clip-overlapping 1.

Smith-Waterman algorithm

The Smith-Waterman algorithm compare the reference to the query using local alignment.



match=+2
mismatch=-1
gap=1

Isaac4: construct matrix

We search for the pattern **TCC** in our reference genome sequence **AATCC**

Example:
match= +2
mismatch= -1
gap= -1

		A	A	T	C	C
	0	0	0	0	0	0
T	0					
C	0					
C	0					

1. In the first step, the first row and column are set to 0

		A	A	T	C	C
	0	0	0	0	0	0
T	0					
C	0					
C	0					

2. We want to fill the first cell. To do that we have to check all the possibilities from the 3 cells before.

		A
	0	0
T	0	

↓ Gap: -1
↘ Mismatch: -1
→ Gap: -1

So the max is 1, so the value of the cell is 1.

Isaac4: construct matrix

We search for the pattern **TCC** in our reference genome sequence **AATCC**

Example:
match= +2
mismatch= -1
gap= -1

		A	A	T	C	C
	0	0	0	0	0	0
T	0	1				
C	0					
C	0					

1. In the first step, the first row and column are set to 0

		A	A
	0	0	0
T	0	-1	

↓ Gap: $0 + 1 = -1$

↘ Mismatch: $0 - 1 = -1$

→ Gap: $-1 - 1 = -2$

So the max is 2, so the value of the cell is 2.

		A	A	T	C	C
	0	0	0	0	0	0
T	0	-1	-1	2	1	0
C	0	-1	-2	1	4	3
C	0	-1	-2	0	5	4

Isaac4: reconstruct the path

We search for the pattern **TCC** in our reference genome sequence **AATCC**

		A	A	T	C	C
	0	0	0	0	0	0
T	0	-1	-1	2	1	0
C	0	-1	-2	1	4	3
C	0	-1	-2	0	5	8

Reconstruct starting from the highest value, back up to the start

The local alignment will be:

AATCC

- - TCC

Alignment with Isaac4 command

- Create a folder: `mkdir Fastq`
- Copy fastq files into the folder:

```
cp R1.fastq.gz Fastq/lane1_read1.fastq.gz
```

```
cp R2.fastq.gz Fastq/lane1_read2.fastq.gz
```

- Align your reads to the reference genome (chr6 hg38):

```
/opt/Isaac4/bin/isaac-align -r /home/lessons/HGE_2021/ref/chr6.hg38.fa -b  
$(pwd) --base-calls-format fastq -t Temp -o Aligned --default-adapters  
Standard -m30 --keep-duplicates 0 -j 5 --clip-overlapping 1
```

Alignment command

- Open the **BWA** file:
 `samtools view sample.sorted.bwa.bam | less -S`
- Close the visualization: `q`

Alignment output – BAM file

Header

Body

```
@HD VN:1.3 SO:coordinate
@SQ SN:chr2 LN:243199373
@PG ID:bwa PN:bwa VN:0.7.15-r1140 CL:bwa mem chr2.fasta read1.trimmed.fastq.gz read2.trimmed.fastq.gz
NB500897:75:H5NWWBGXY:3:13608:21224:10540 2147 chr2 12323980 0 57H38M21H = 69870261 57546346 GGCTGAGGTGGGAG
NB500897:78:H5T5FBGXY:4:11401:24478:5392 2131 chr2 25275067 0 108H33M9H = 69650885 44375787 AAAAATAAAAAATAA
NB500897:75:H5NWWBGXY:3:12601:23290:7800 2115 chr2 33141345 0 94H56M = 68269866 35128522 GGGGGGGGGGGGGGGGGGGGGG
NB500897:78:H5T5FBGXY:3:12609:16449:18291 2115 chr2 33141438 0 120H30M = 68547068 35405631 GGGGGGGGGGGGGGGGGGGGGG
NB500897:75:H5NWWBGXY:1:13104:6732:16100 2115 chr2 33141478 0 120H30M = 68547068 35405591 GGGGGGGGGGGGGGGGGGGGGG
NB500897:78:H5T5FBGXY:4:12411:14105:19392 2147 chr2 33141574 0 100H50M = 69871302 36729810 GGGGGGGGGGGGGGGGGGGGGG
NB500897:75:H5NWWBGXY:2:23311:23906:6768 2115 chr2 33141581 0 87H43M = 68269811 35128231 GGGGGGGGGGGGGGGGGGGGGG
NB500897:75:H5NWWBGXY:1:12106:7875:15761 2115 chr2 33141594 0 120H30M = 68547068 35405475 GGGGGGGGGGGGGGGGGGGGGG
NB500897:78:H5T5FBGXY:3:22601:12050:10673 161 chr2 50626928 0 2S64M = 69890571 19263716 AAAAAAAAAAAAAAAAAAAAAA
NB500897:75:H5NWWBGXY:4:12607:26302:14475 81 chr2 52194430 0 78M = 69688660 17494154 ATGTTGGCGAGGCTGGTCTCCA
NB500897:78:H5T5FBGXY:4:23607:18581:7865 163 chr2 58355595 0 26M = 58355665 151 TAGCTGGGATTACAGGTGTGTGCCAC
NB500897:78:H5T5FBGXY:4:23607:18581:7865 83 chr2 58355665 0 4S81M13S = 58355595 -151 CTCACCATGTTGCCAGGCTGG
NB500897:75:H5NWWBGXY:3:13608:21224:10540 2179 chr2 65437823 0 63M53H = 69870261 4432439 ACTATGCTGACCAGGTTGGTTTCAAATTCC
NB500897:75:H5NWWBGXY:2:11103:13114:1410 99 chr2 67999995 60 66M = 68000128 259 CTCACAATAAATTTATTTTTTCAAAGCAG
NB500897:75:H5NWWBGXY:2:11103:13114:1410 147 chr2 68000128 60 126M = 67999995 -259 CATCTAGATAGCTATCTTTCCAGACTTTTC
NB500897:78:H5T5FBGXY:1:21201:19536:13168 99 chr2 68000446 60 150M = 68000452 156 CCTCAGAGTATTAAGACCACATAGTATAT
NB500897:78:H5T5FBGXY:4:23607:24808:1972 99 chr2 68000446 60 150M = 68000452 156 CCTCAGAGTATTAAGACCACATAGTATAT
NB500897:78:H5T5FBGXY:1:21201:19536:13168 147 chr2 68000452 60 150M = 68000446 -156 AGTATTAAGACCACATAGTATATATTTTC
NB500897:78:H5T5FBGXY:4:23607:24808:1972 147 chr2 68000452 60 150M = 68000446 -156 AGTATTAAGACCACATAGTATATATTTTC
NB500897:75:H5NWWBGXY:4:12408:9893:6566 163 chr2 68001096 60 150M = 68001096 150 AGATTCCTGGCTAAATTCACCATTGAAAGAAATTGA
NB500897:75:H5NWWBGXY:4:12408:9893:6566 83 chr2 68001096 60 150M = 68001096 -150 AGATTCCTGGCTAAATTCACCATTGAAAGAAATTGA
NB500897:78:H5T5FBGXY:2:12107:7865:18634 99 chr2 68001412 60 150M = 68001549 287 TTCCTAACTAAATACTGACTAGAACAGTGA
NB500897:78:H5T5FBGXY:2:12107:7865:18634 147 chr2 68001549 60 150M = 68001412 -287 CCCAACTCCTGCCACTCTAGCCACATCAAG
NB500897:78:H5T5FBGXY:1:12109:14488:19284 99 chr2 68001927 60 150M = 68002195 418 ACGTAGTAGAAATTCACAGAATACTTGTA
NB500897:78:H5T5FBGXY:1:12305:16077:11340 163 chr2 68001976 60 150M = 68002289 462 AATTCCTCACCAGCTTCAGCAGCTTAAGGA
NB500897:75:H5NWWBGXY:1:22202:17880:17856 99 chr2 68001995 60 150M = 68002066 221 CAGCTTAAGGATAAAGAATCTTGCATCTAA
```

Read ID

Mapping
Tag

Mapping
position

Mapping quality

CIGAR
string

Pair
information

Cigar

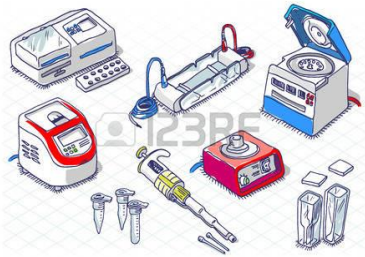
Op	Description
M	alignment match (can be a sequence match or mismatch)
I	insertion to the reference
D	deletion from the reference
N	skipped region from the reference
S	soft clipping (clipped sequences present in SEQ)
H	hard clipping (clipped sequences NOT present in SEQ)
P	padding (silent deletion from padded reference)
=	sequence match
X	sequence mismatch

DUPLICATES REMOVAL

Library
preparation



Bioinformatic
analysis



Sequencing



Data QC

Alignment

Variant
calling

Variant
annotation

Variant
prioritization

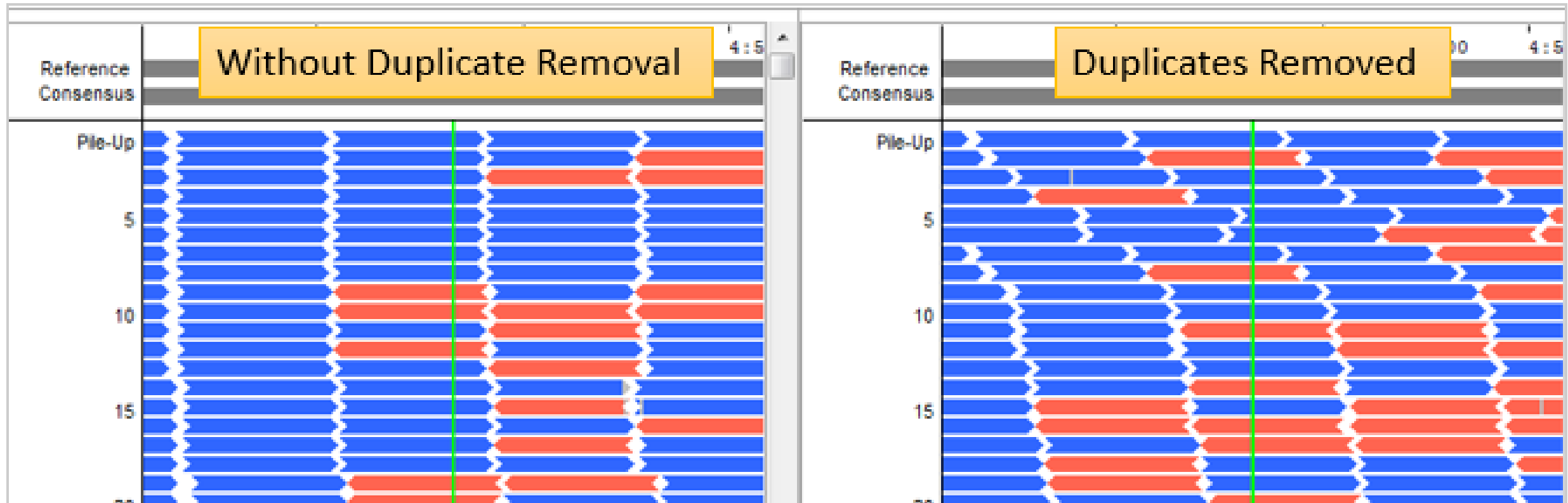
Picard Mark Duplicates Algorithm

Picard by GATK:

The MarkDuplicates tool works by comparing sequences in the 5 prime positions of both reads and read-pairs in a SAM/BAM file. After duplicate reads are collected, the tool differentiates the primary and duplicate reads using an algorithm that ranks reads by the sums of their base-quality scores (default method).



Picard Mark Duplicates



Mark Duplicates

1. Mark Duplicates with picard:

```
java -jar /opt/picard.jar MarkDuplicates  
INPUT=sample.sorted.bwa.bam  
OUTPUT=sample.sorted.dedup.bwa.bam  
REMOVE_DUPLICATES=false METRICS_FILE=duplicates.txt
```

2. Open the output file: `less -S duplicates.txt`

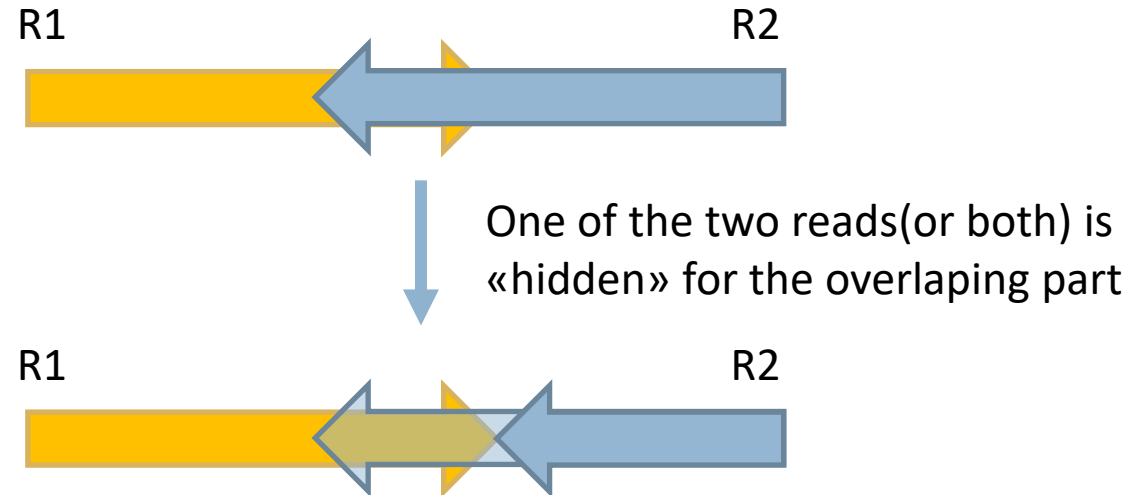
```
## htsjdk.samtools.metrics.StringHeader  
# MarkDuplicates INPUT=[sample.sorted.bwa.bam] OUTPUT=sample.sorted.dedup.bwa.bam METRICS_FILE=duplicates.txt REMOVE_DUPLICATES=false  
## htsjdk.samtools.metrics.StringHeader  
# Started on: Wed Jan 13 14:22:59 CET 2021  
  
## METRICS CLASS      picard.sam.DuplicationMetrics  
LIBRARY UNPAIRED_READS_EXAMINED READ_PAIRS_EXAMINED SECONDARY_OR_SUPPLEMENTARY_READS UNMAPPED_READS UNPAIRED_READ_DUPLICATES  
Unknown Library 1305 2220416 32743 1355 553 211540 16339 0,095367 11697388
```

Percentage of duplicates

READ CLIPPING

BamUtils clipping

Bamutil: The clipOverlap option of bamUtil **only soft-clips** overlapping read pairs. When two mates overlap, **this tool will clip the record's whose clipped region would have the lowest average quality.**



BamUtils clipping command

1. Soft-clip overlapping reads with BamUtils:

```
/opt/bamUtil/bin/bam clipOverlap --in sample.sorted.dedup.bwa.bam --out sample.sorted.dedup.clipped.bwa.bamUtils.bam
```

2. Create index:

```
samtools index sample.sorted.dedup.clipped.bwa.bamUtils.bam
```

View bam **before** soft-clipping:

```
samtools view sample.sorted.dedup.bwa.bam | less -S
```

MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	99	chr6	60065	60	151M	=	60223	309
MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	147	chr6	60223	60	151M	=	60065	-309
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	99	chr6	61798	60	151M	=	61908	261
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	147	chr6	61908	60	151M	=	61798	-261
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	163	chr6	62841	60	151M	=	63114	424
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	82	chr6	63114	60	151M	=	62841	-424

View bam **after** soft-clipping:

```
samtools view sample.sorted.dedup.clipped.bwa.bamUtils.bam | less -S
```

MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	99	chr6	60065	60	151M	=	60223	309
MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	147	chr6	60223	60	151M	=	60065	-309
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	99	chr6	61798	60	110M41S	=	61908	261
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	147	chr6	61908	60	151M	=	61798	-261
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	163	chr6	62841	60	151M	=	63114	424
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	82	chr6	63114	60	151M	=	62841	-424

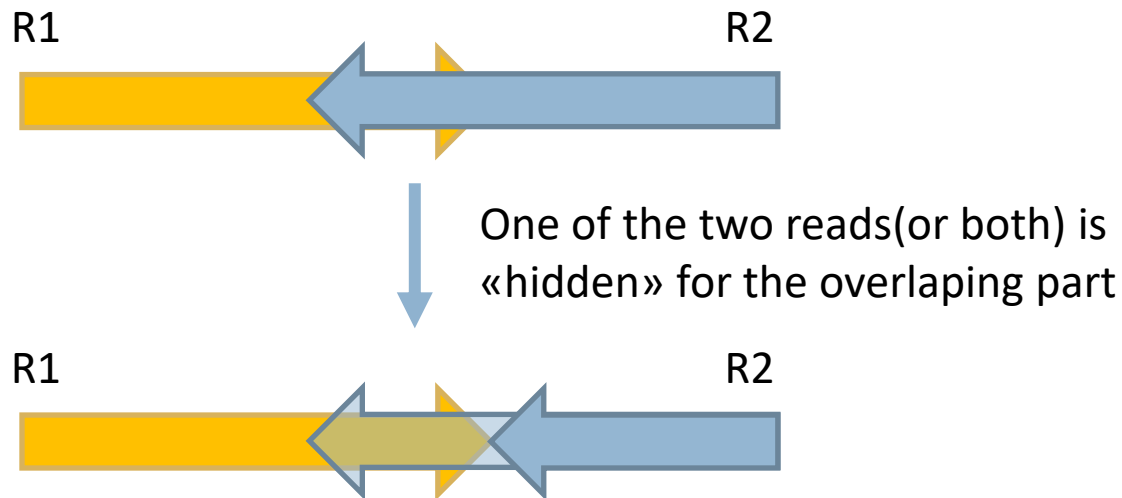
CIGAR string has been
changed, 41 bases
were soft-clipped

fgBio clipping

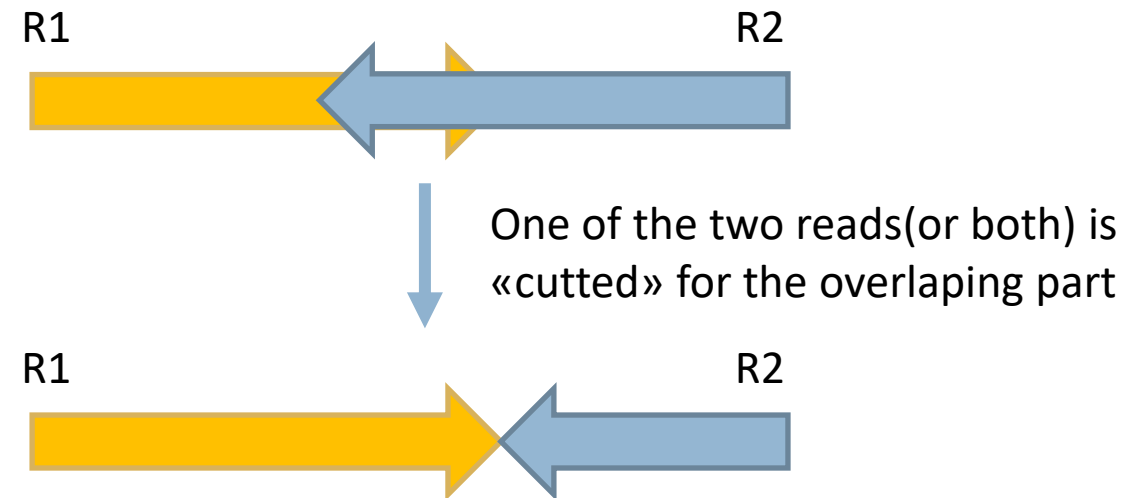
[Fgbio ClipBam](#): Clips overlapping read pairs from the same fragment. **Three clipping modes** are supported:

- **Soft**: soft-clip the bases and qualities.
- **SoftWithMask**: soft-clip and mask the bases and qualities (make bases Ns and qualities the minimum).
- **Hard (default)**: hard-clip the bases and qualities.

Soft-clipping



Hard-clipping



fgBio clipping command

1. Hard-clip overlapping reads with BamUtils:

```
java -jar /opt/fgbio-1.1.0.jar ClipBam -i sample.sorted.dedup.bwa.bam -o sample.sorted.dedup.bwa.fgbio.bam -r  
../ref/chr6.hg38.fa -c Hard --clip-overlapping-reads true
```

View bam **before** hard-clipping:

```
samtools view sample.sorted.dedup.bwa.bam | less -S
```

MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	99	chr6	60065	60	151M	=	60223	309
MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	147	chr6	60223	60	151M	=	60065	-309
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	99	chr6	61798	60	151M	=	61908	261
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	147	chr6	61908	60	151M	=	61798	-261
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	163	chr6	62841	60	151M	=	63114	424
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	99	chr6	63114	60	151M	=	62841	-424

View bam **after** hard-clipping:

```
samtools view sample.sorted.dedup.clipped.bwa.fgbio.bam | less -S
```

MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	99	chr6	60065	60	151M	=	60223	309
MG01HX01:853:HWY5YCCXY:5:2120:11627:31336	147	chr6	60223	60	151M	=	60065	-309
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	99	chr6	61798	60	131M20H	=	61929	261
MG01HX01:853:HWY5YCCXY:3:2215:26839:21649	147	chr6	61929	60	21H130M	=	61798	-261
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	163	chr6	62841	60	151M	=	63114	424
MG01HX01:853:HWY5YCCXY:6:1104:18213:52731	99	chr6	63114	60	151M	=	62841	-424

CIGAR string has been
changed, 41 bases were
soft-clipped from the two
pairs

Insert size

1. Calculate the insert size using picard:

```
java -jar /opt/picard.jar CollectInsertSizeMetrics I=sample.sorted.dedup.clipped.bwa.bamUtils.bam  
H=sample.sorted.dedup.clipped.bwa.bamUtils.hist.pdf O=sample.sorted.dedup.clipped.bwa.bamUtils.output AS=true  
VALIDATION_STRINGENCY=SILENT
```

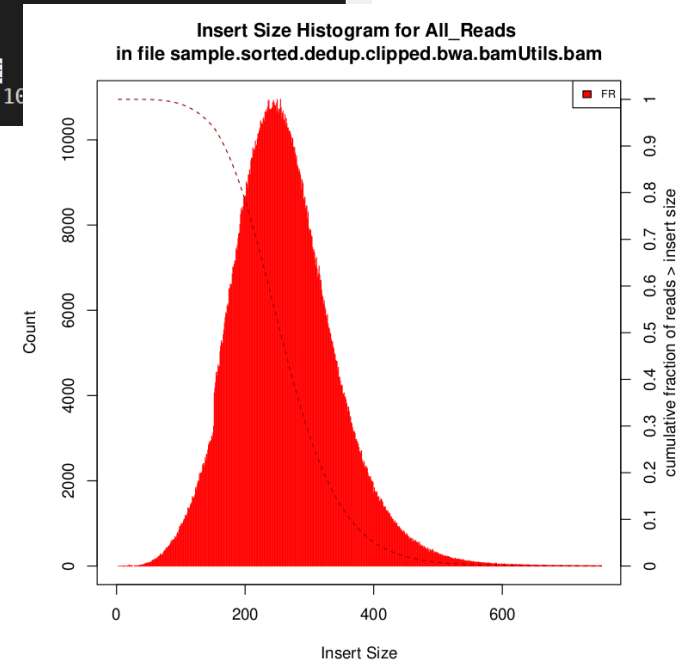
2. Check the results:

```
less -S sample.sorted.dedup.clipped.bwa.bamUtils.output
```

```
## htsjdk.samtools.metrics.StringHeader  
# CollectInsertSizeMetrics HISTOGRAM_FILE=sample.sorted.dedup.clipped.bwa.bamUtils.hist.pdf INPUT=sample.sorted.dedup.clipped.bwa.bamUtils.bam OUTPUT=sampl  
## htsjdk.samtools.metrics.StringHeader  
# Started on: Wed Jan 13 15:24:07 CET 2021  
  
## METRICS CLASS picard.analysis.InsertSizeMetrics  
MEDIAN_INSERT_SIZE MODE_INSERT_SIZE MEDIAN_ABSOLUTE_DEVIATION MIN_INSERT_SIZE MAX_INSERT_SIZE MEAN_INSERT_SIZE  
254 255 50 3 170533062 261,069854 79,617853 1977539 FR 19 39 57 79 16
```

3. Check the on histogram:

```
evince sample.sorted.dedup.clipped.bwa.bamUtils.hist.pdf
```



VISUALIZATION OF ALIGNED READS ON IGV



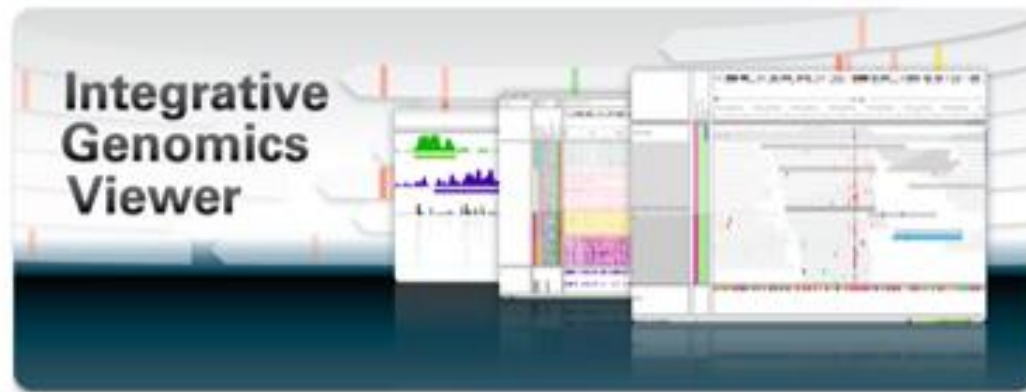
- Home
- Downloads
- Documents
 - IGV User Guide
 - Tutorial Videos
 - File Formats
 - Hosted Genomes
 - FAQ
 - Release Notes
 - Credits
- Contact

Search website

search

© 2013-2018
Broad Institute
and the Regents of the
University of California

Home



Overview

The **Integrative Genomics Viewer (IGV)** is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations.

IGV is available in multiple forms, including:

- the original **IGV** - a Java desktop application,
- IGV-Web** - a web application,
- igv.js** - a JavaScript component that can be embedded in web pages (for developers)

This site is focused on the IGV desktop application. See <https://igv.org> for links to all forms of IGV.

Download IGV



Download the IGV desktop application and igvtools.

Note that the IGV-Web application at <https://igv.org/app> runs in a web browser and requires no downloads. Click on the Help link in the app for more information.

Citing IGV

To cite your use of IGV in your publication, please reference one or more of:

James T. Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, Jill P. Mesirov. [Integrative Genomics Viewer \(IGV\): high-performance genomics data visualization and exploration. *Briefings in Functional Genomics and Proteomics* 29, 24-26 \(2011\). \(Free PMC article \[here\]\(#\)\).](#)

Helga Thorvaldsdóttir, James T. Robinson, Jill P. Mesirov. [Integrative Genomics Viewer \(IGV\): high-performance genomics data visualization and exploration. *Briefings in Functional Genomics and Proteomics* 14, 178-192 \(2013\).](#)

James T. Robinson, Helga Thorvaldsdóttir, Aaron M. Wenger, Ahmet Zehir, Jill P. Mesirov. [Variant Review with the Integrative Genomics Viewer \(IGV\). *Cancer Research* 77\(21\):31-34 \(2017\).](#)

Funding

Development of IGV has been supported by funding from the [National Cancer Institute \(NCI\)](#) of the [National Institutes of Health](#), the [Informatics Technology for Cancer Research \(ITCR\)](#) of the NCI, and the [Star Cancer Consortium](#).

IGV participates in the [GenomeSpace](#) initiative, which is funded by the [National Human Genome Research Institute](#).

IGV

<https://software.broadinstitute.org/software/igv/>

Download IGV

Install IGV 2.8.x

See the [Release Notes](#) for what's new in each release.



IGV Mac App

Download and unzip the Mac App Archive, then double-click the IGV application to run it. You can move the app to the *Applications* folder, or anywhere else.

MacOS Catalina users: We sign our Mac App as a trusted Apple developer, but it is not yet notarized by Apple (a new requirement in Catalina). To run it, right-click on the downloaded IGV app; select "Open" from the menu; and click the "Open" button in the window that pops up. After that, double-clicking on the app will also work.



IGV for Windows

Download and run the installer.
An IGV shortcut will be created on the Desktop; double-click it to run the application.



IGV for Linux

Download and unzip the Archive.
See the downloaded *readme.txt* for further instructions.



IGV and igvtools to run on the command line (all platforms)

Download and unzip the Archive. **Requires Java 11.**
See the downloaded *readme.txt* and *igvtools_readme.txt* for further instructions.

Download the bam and the bai

- Download the bam file and the index file on your pc:
- Open new terminal:

`cd Desktop/HGE_2021`

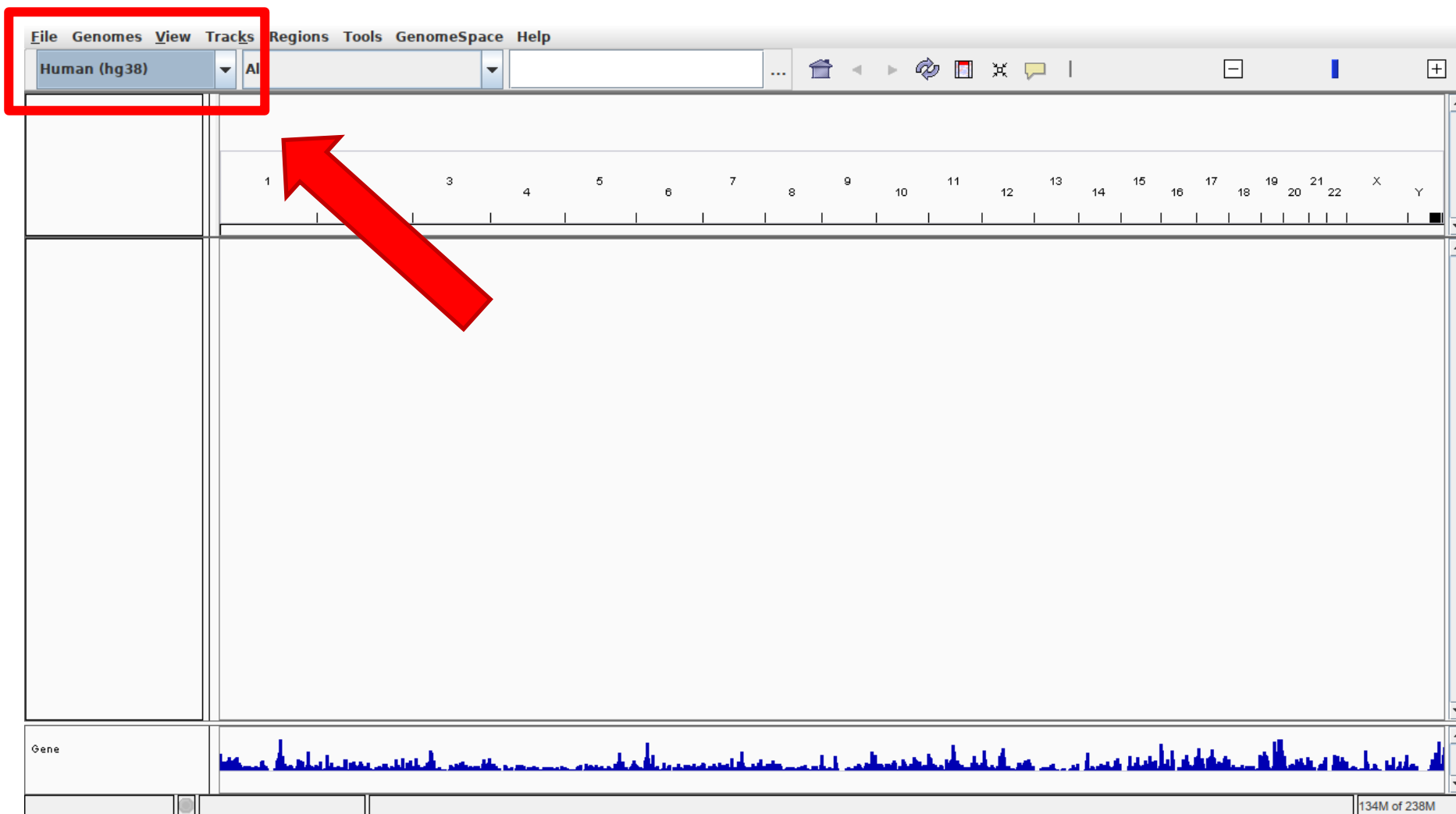
`rsync -auv lessons@157.27.80.26:/home/lessons/HGE_2021/your_name/sample.sorted.dedup.clipped.bwa.bamUtils.ba* .`

`rsync -auv lessons@157.27.80.26:/home/lessons/HGE_2021/denise/sample.sorted.dedup.clipped.bwa.fgbio.ba* .`

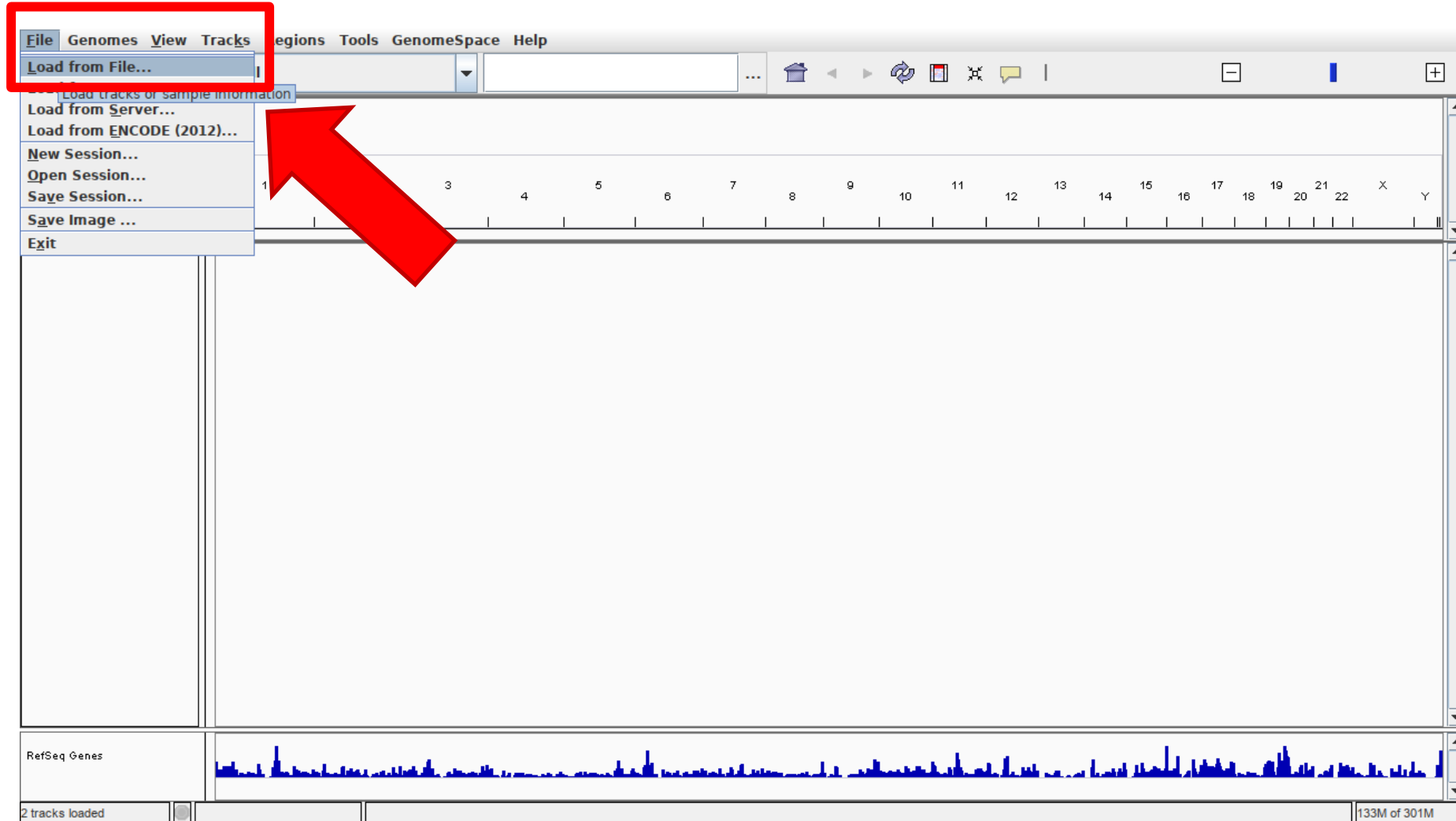
Password: `lez2021`

- Check if you have downloaded: `ls`
- Open IGV
`./igv.sh` for Ubuntu

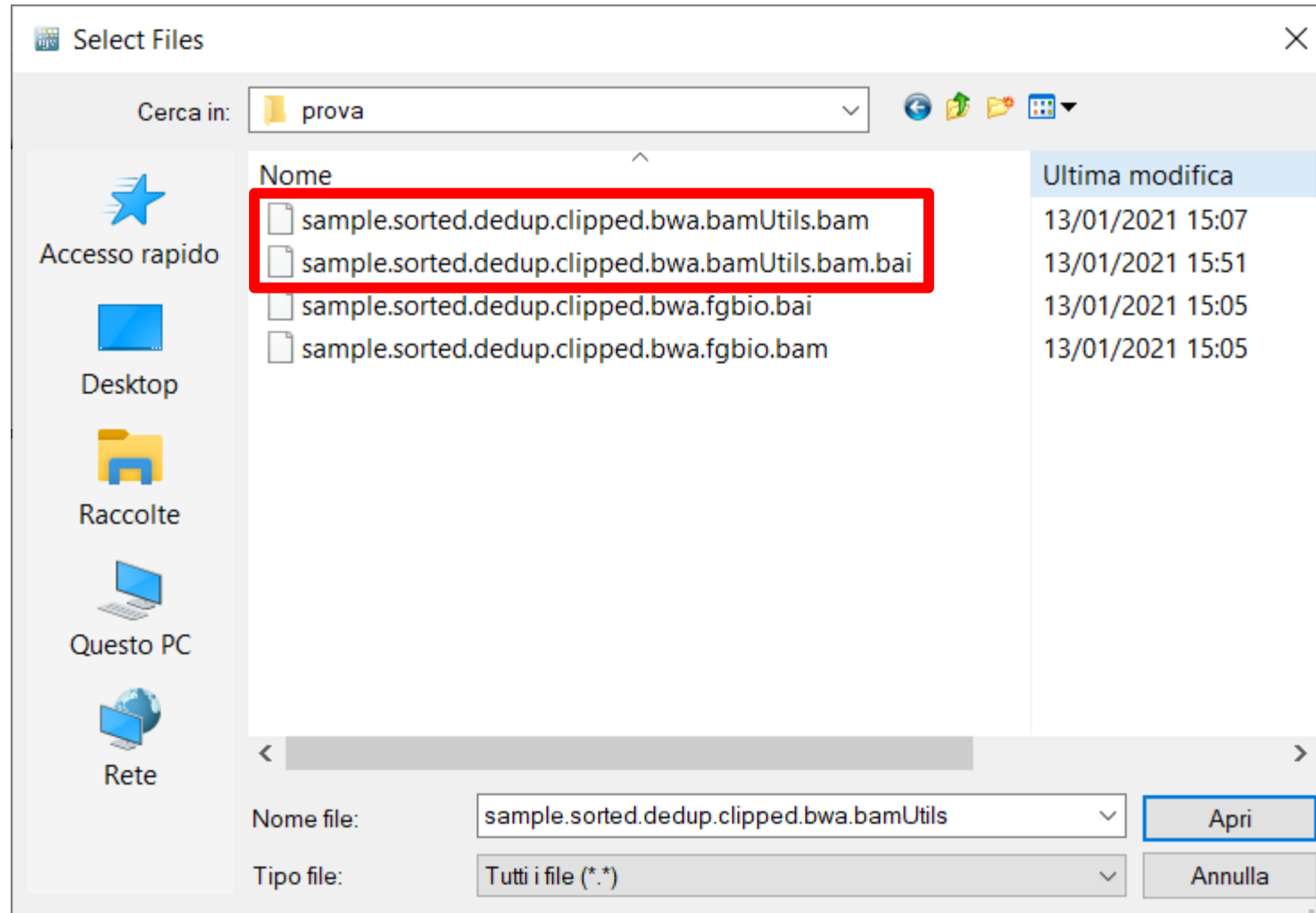
Choose the right genome



Upload the bam



Go into the folder «HGE_2021»,
choose the file bam and open it



Search a specific region

The screenshot displays a genomic browser interface with a search bar at the top containing the text "chr6:289,015-307,482". A red rectangle highlights the search bar, and a large red arrow points from the text "We search the region: chr6:289,015-307,482" to the search bar. The interface includes a menu bar with options like File, Genomes, View, Tracks, Regions, Tools, GenomeSpace, and Help. Below the menu bar, there are dropdown menus for "Human (hg38)" and "All". The main area shows a genomic track with chromosomes 1 through 22, X, and Y. The track is labeled "sample.sorted.bam Coverage" and "sample.sorted.bam". The bottom track is labeled "Gene". The status bar at the bottom indicates "4 tracks" and "154M of 291M".

Human (hg38) All chr6:289,015-307,482 Go

1 2 3 4 5 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 X Y

sample.sorted.bam Coverage

sample.sorted.bam

Gene

4 tracks 154M of 291M

We search the region:
chr6:289,015-307,482

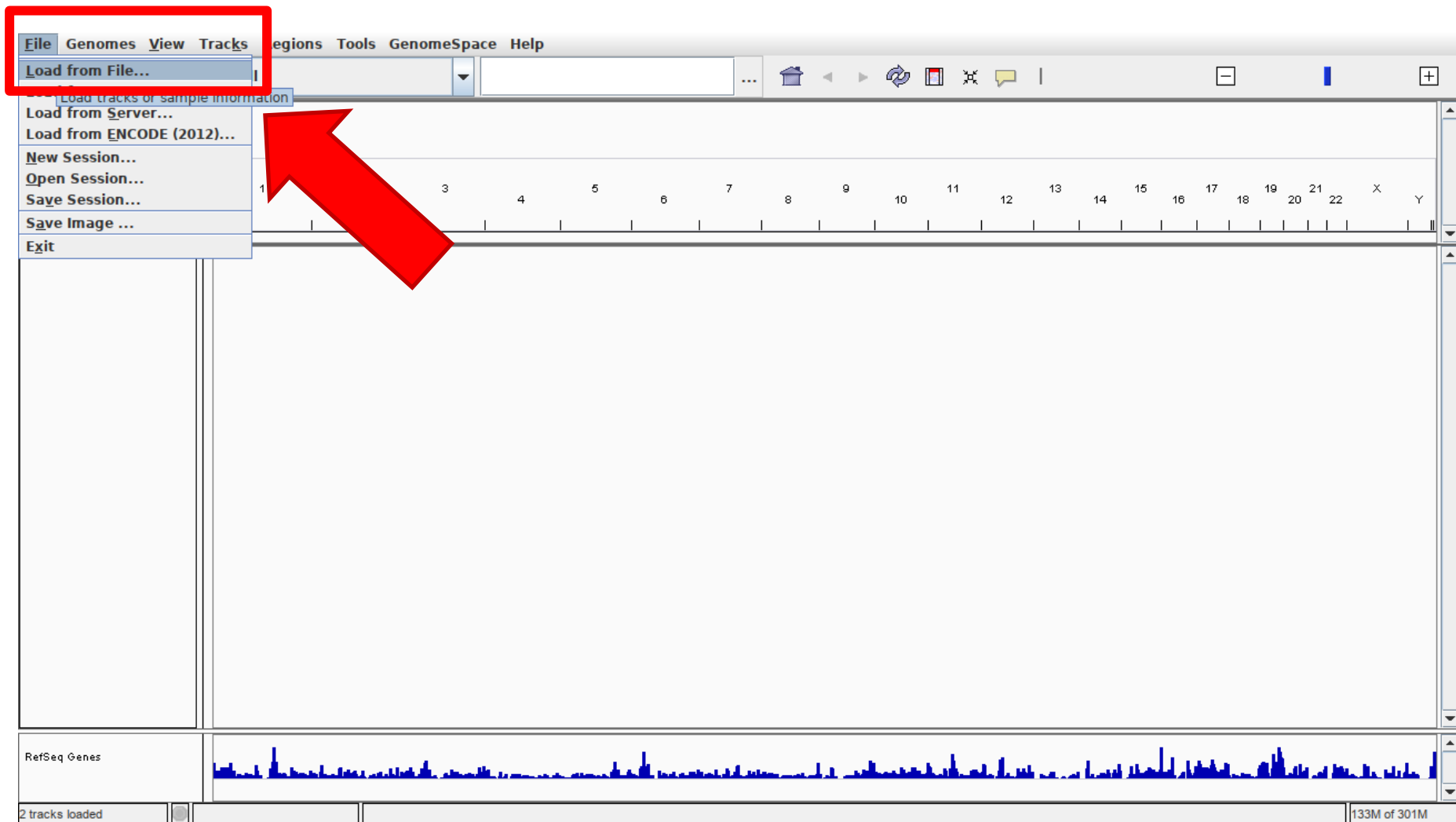
Results



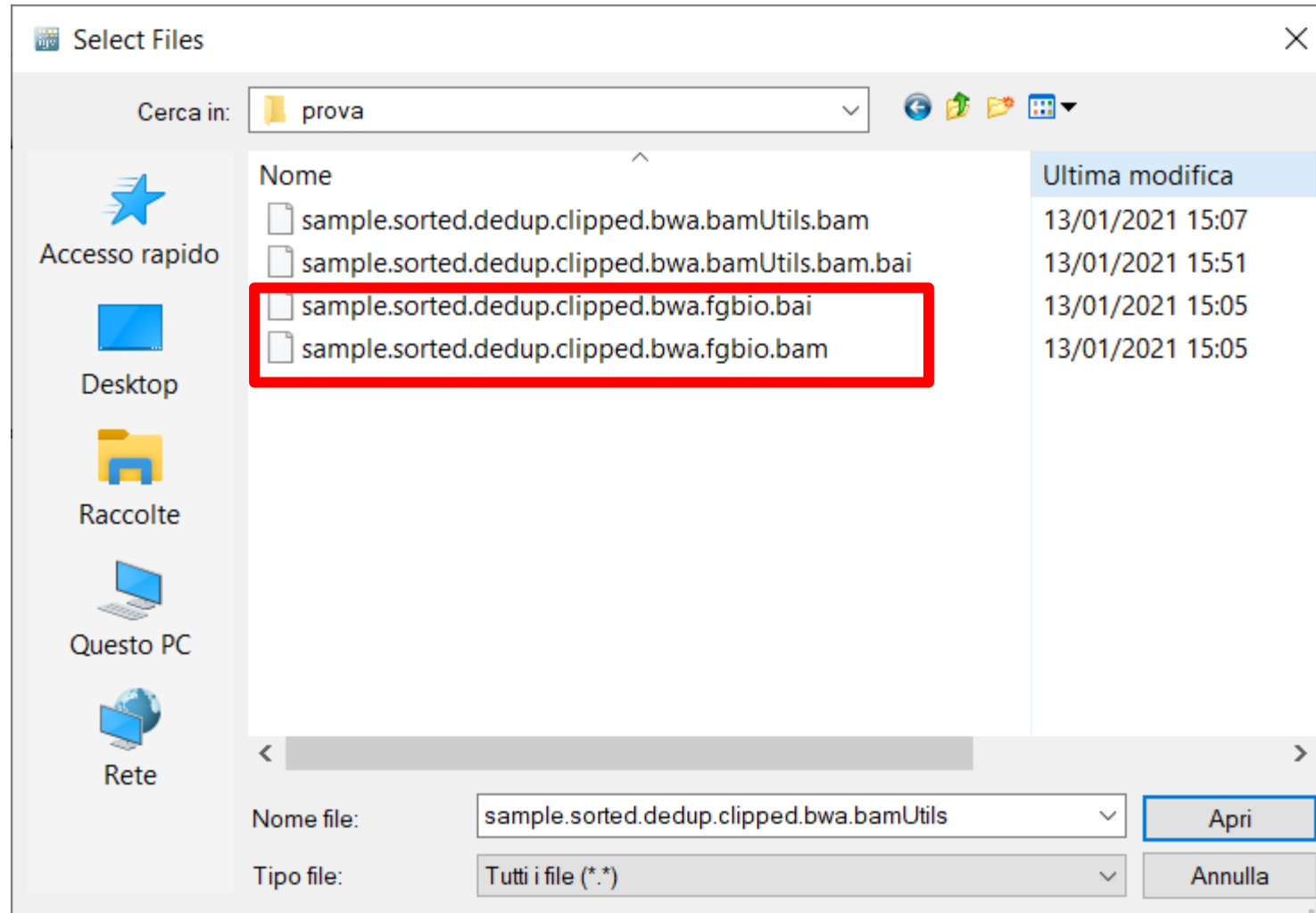
Difference between genome and exome sequencing



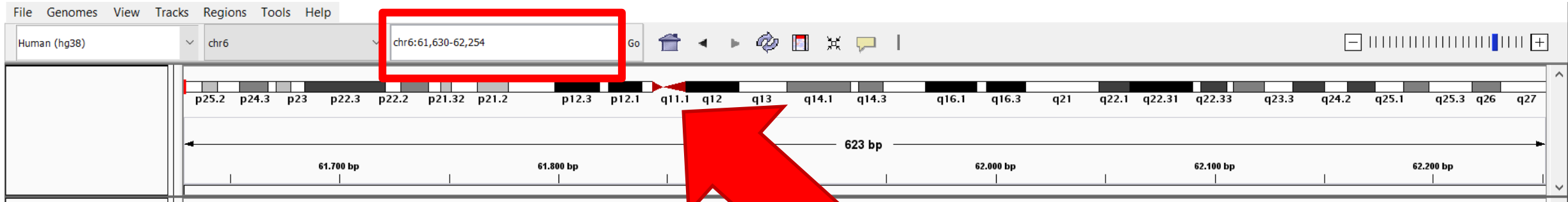
Upload the second bam



Go into the folder "HGS12020",
choose the file bam and open it

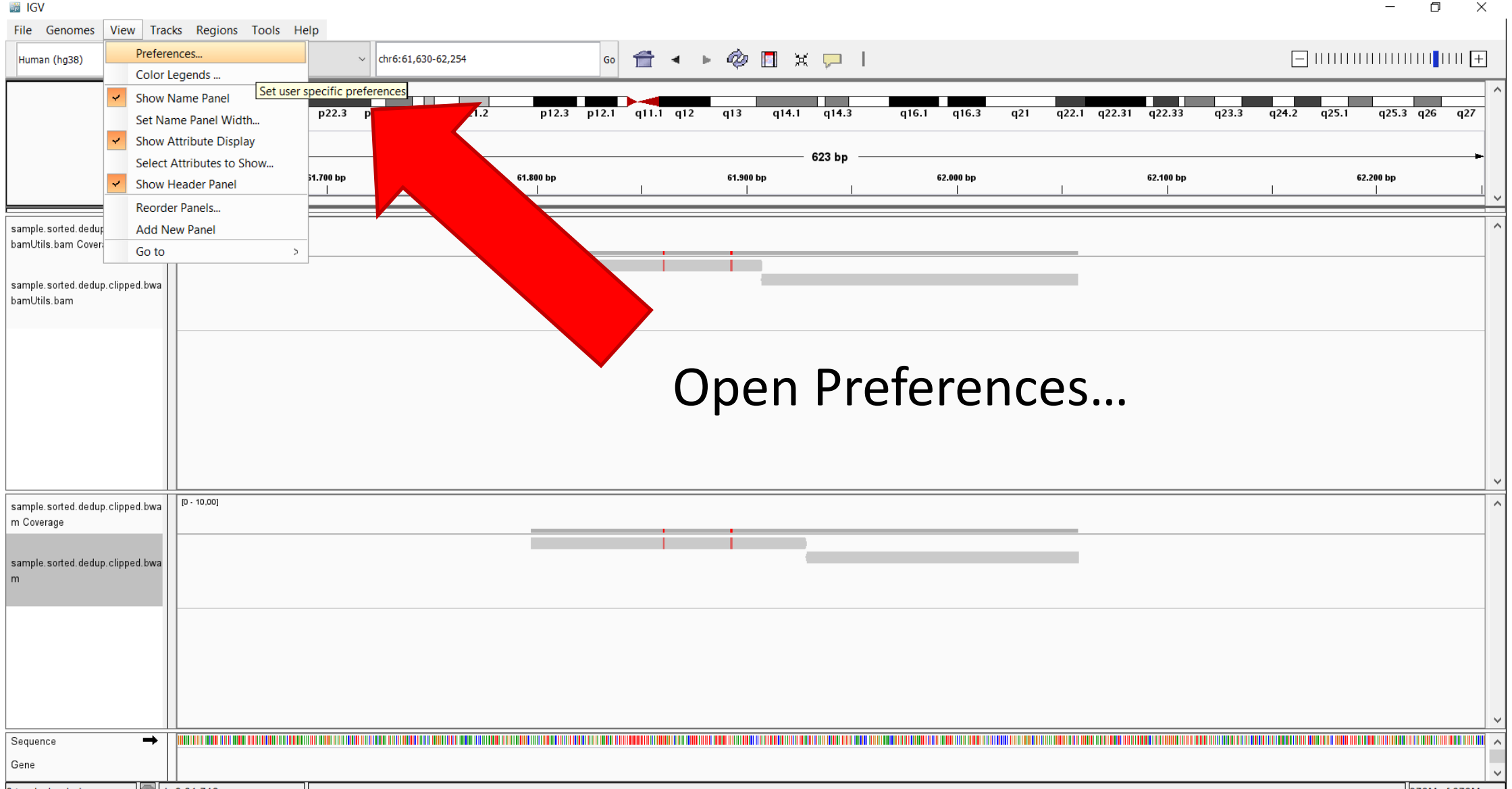


Search a specific region



We search the region:
chr6:61,630-62,254

Check differences between soft and hard clipping



IGV

File Genomes View Tracks Regions Tools Help

Human (hg38) Preferences... Color Legends ... chr6:61,630-62,254 Go

Set user specific preferences

- ☒ Show Name Panel
- Set Name Panel Width...
- ☒ Show Attribute Display
- Select Attributes to Show...
- ☒ Show Header Panel
- Reorder Panels...
- Add New Panel
- Go to >

sample.sorted.dedup.bam bamUtils.bam Coverage

sample.sorted.dedup.clipped.bwa bamUtils.bam

sample.sorted.dedup.clipped.bwa m Coverage

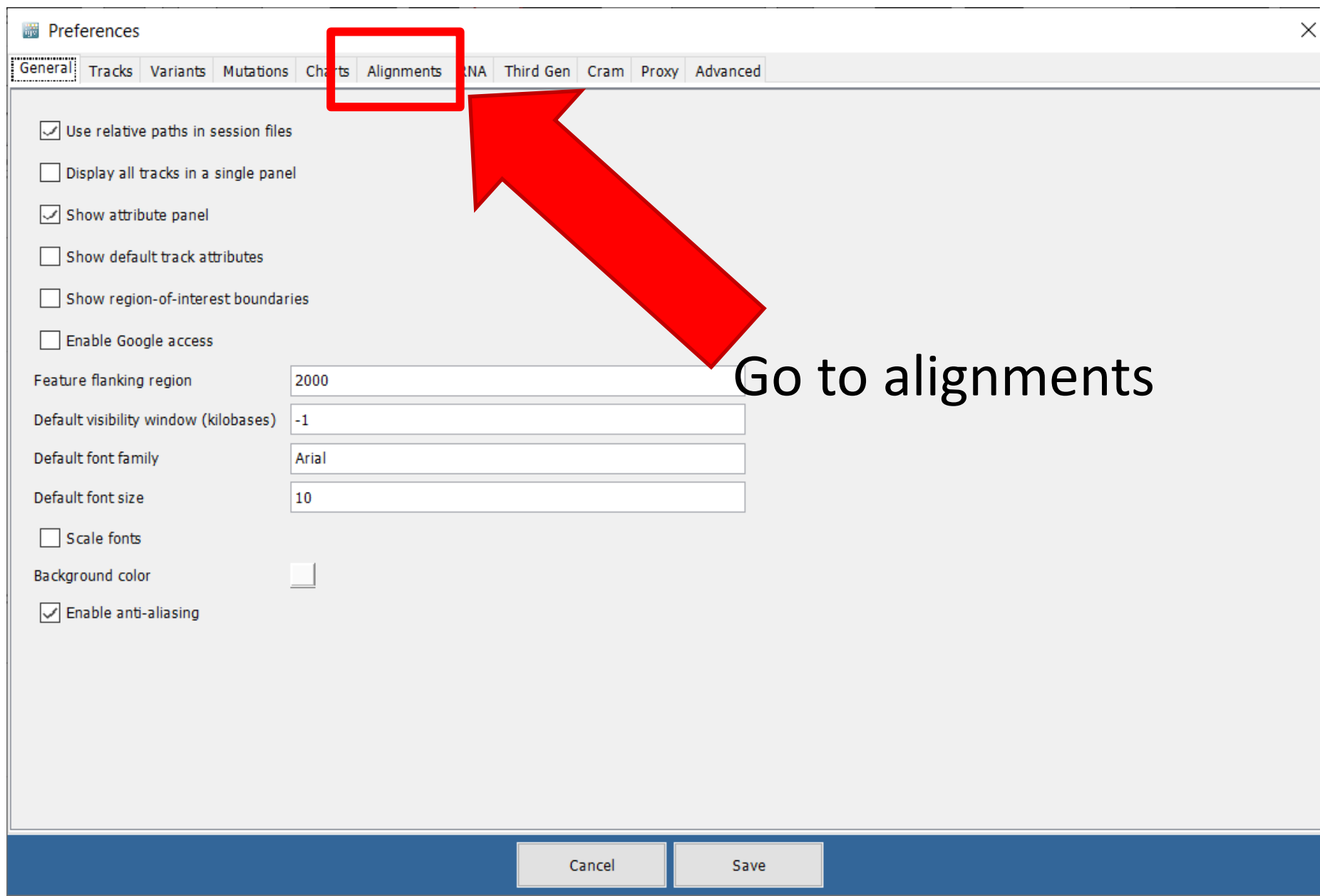
sample.sorted.dedup.clipped.bwa m

Sequence →

Gene

Open Preferences...

Check differences between soft and hard clipping



Check differences between soft and hard clipping

Preferences

General Tracks Variants Mutations Charts **Alignments** RNA Third Gen Cram Proxy Advanced

Linking tag: READNAME

☐ Filter duplicate reads

☒ Filter vendor failed reads

☐ Filter secondary alignments

☒ Filter supplementary alignments

☐ Flag unmapped pairs

☐ Show center line

Hidden SAM tags: SA,MD,XA,RG

☐ Show soft-clipped bases

Maximum soft clip size (reference sequence is extended by this amount): 1000

☐ Filter alignments by read group

URL or path to read group filter file:

Coverage Track Options

Coverage allele-fraction threshold: 0.2f

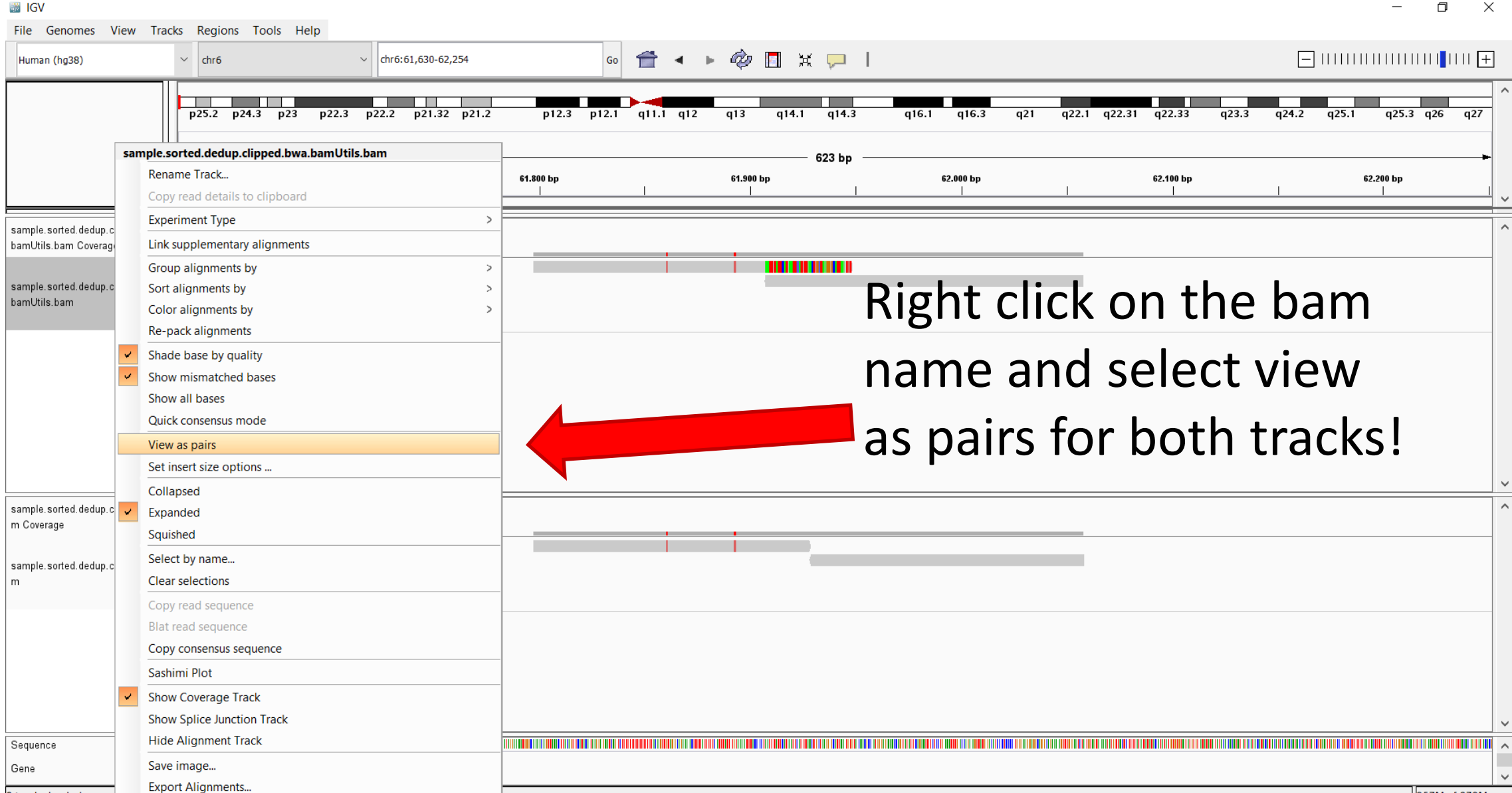
☒ Quality weight allele fraction

Splice Junction Track

Cancel Save

Select «show soft-clipped bases» and save

Check differences between soft and hard clipping



Check differences between soft and hard clipping

IGV

File Genomes View Tracks Regions Tools Help

Human (hg38) chr6 chr6:61,630-62,254 Go

sample.sorted.dedup.clipped.bwa.bamUtils.bam

- Rename Track...
- Copy read details to clipboard
- Experiment Type >
- Link supplementary alignments
- Group alignments by >
- Sort alignments by >
- Color alignments by >
 - ☒ no color
 - insert size
 - pair orientation
 - insert size and pair orientation
 - read strand**
 - first-of-pair strand
 - read group
 - sample
 - library
 - movie
 - ZMW
 - tag
 - bisulfite mode >
- Re-pack alignments
- ☒ Shade base by quality
- ☒ Show mismatched bases
- Show all bases
- Quick consensus mode
- ☒ View as pairs
- Set insert size options ...
- Collapsed
- ☒ Expanded
- Squished
- Select by name...
- Clear selections
- Copy read sequence
- Blat read sequence
- Copy consensus sequence
- Sashimi Plot
- ☒ Show Coverage Track
- Show Splice Junction Track
- Hide Alignment Track
- Save image...
- Export Alignments...
- Export track names...

Right click on the bam name,
go to «color alignments by»
and select «read strand»

360M of 673M

Results differences between soft and hard clipping

