IGSR - Alignment of Next-Generation sequencing data

This part of the course covers the alignment or mapping of the Next generation sequencing (NGS) data generated from a given organism for which a reference sequence for the genome already exists.

In this course, we will use the genomic data generated for the *Oryza sativa* Japonica (rice) species in a project called the 3000 Rice genomes project, which is an international effort to resequence a core collection of >3000 rice accessions from 92 countries. We will use the re-sequencing data generated in this project to identify genomic variants in this plant species.

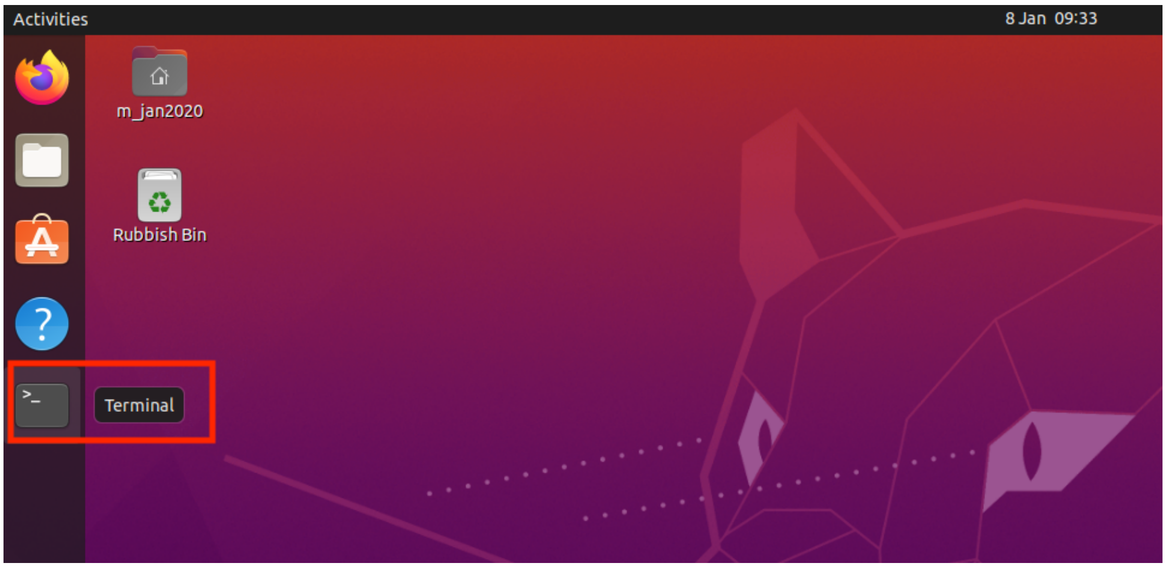
The data from the 3000 Rice genomes project is openly available at the European Nucleotide Archive ENA. It is accessible using the PRJEB6180 study id.

Log in the Ubuntu virtual machine

```
    user: m_jan2020
    pwd: m_jan2020
```

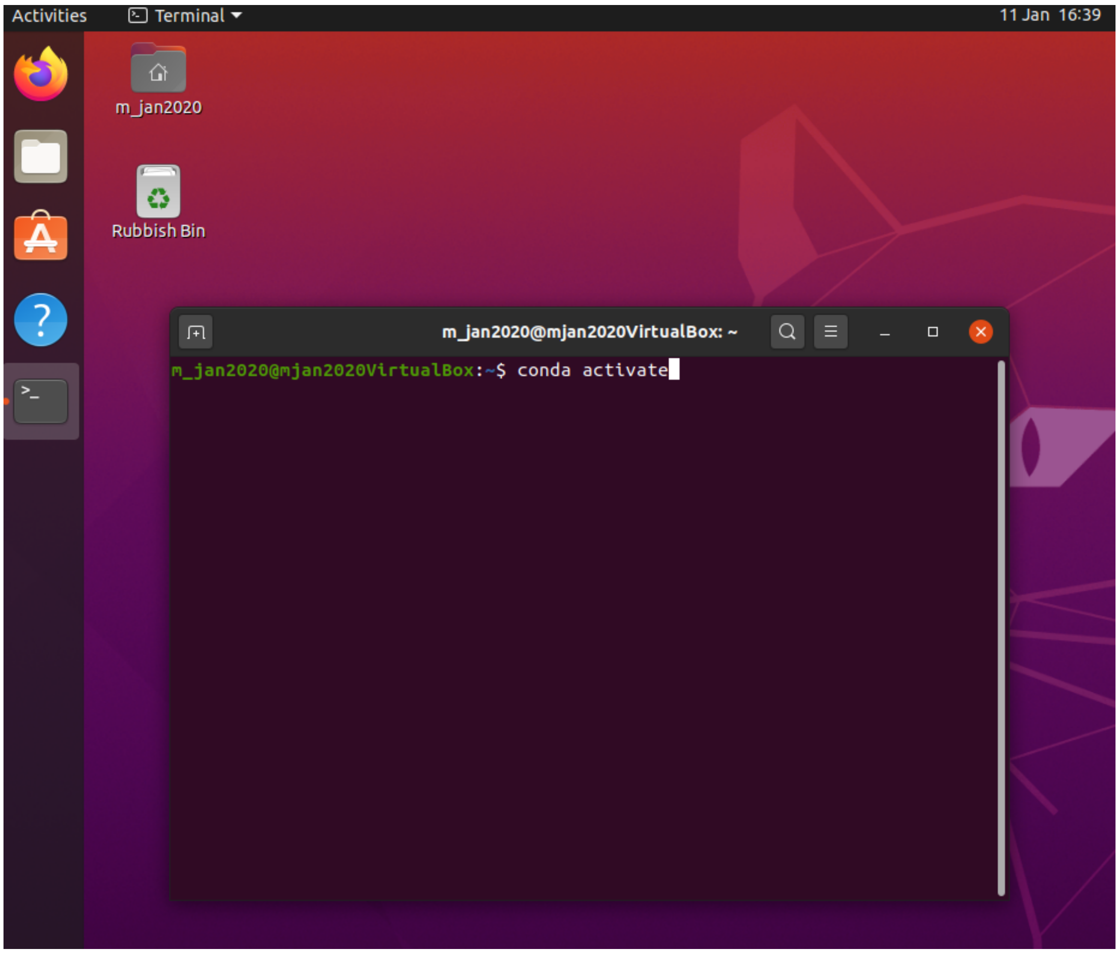
Open a Terminal window in the Ubuntu virtual machine

A terminal in Linux is an interface we can use to enter text commands, it will be the interface we will use to run most of the bioinformatics tools shown in this course. To open a terminal click on the `Terminal` icon you see in your screen:



Activate the conda environment used in this course

Conda is a package and environment management system that we have used in this course to install all the bioinformatics programs.
To start using Conda you need first to activate the course environment by entering the following command in the terminal window you have just opened:



```
m_jan2020@mjan2020VirtualBox:~$ conda activate
```

Unix commands used in this course

Most of the bioinformatics tools used for genomic data analysis run in Unix/Linux, so it is recommended to have a basic knowledge of the commands used to move around the different directories in your system. You should also know how to list the contents of a directory or how to print the contents of a given text file.
Here are some of the basic commands we will use during this course:

- Print the current working directory with the pwd command

```
m_jan2020@mjan2020VirtualBox:~$ pwd
/home/m_jan2020
```

- Change directory (cd)

```
# go to the alignment dir
m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/alignment/

# move up one folder
m_jan2020@mjan2020VirtualBox:~$ cd ../

# now, check where you are
m_jan2020@mjan2020VirtualBox:~$ pwd
/home/m_jan2020/course

# get back to original location
m_jan2020@mjan2020VirtualBox:~$ cd
m_jan2020@mjan2020VirtualBox:~$ pwd
/home/m_jan2020
```

- Listing the directory contents (ls)

```
# go to the following directory
m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/data

# basic listing
m_jan2020@mjan2020VirtualBox:~/course/data$ ls
SAMEA2569438.chr10_1.fastq.gz  SAMEA2569438.chr10_2.fastq.gz

# ls with -l (long-format) and -h (human readable)
m_jan2020@mjan2020VirtualBox:~/course/data$ ls -ls
total 23M
```

```
    -rw-rw-r-- 1 m_jan2020 m_jan2020 11M Nov 26 17:43 SAMEA2569438.chr10_1.fastq.gz
    -rw-rw-r-- 1 m_jan2020 m_jan2020 12M Nov 26 17:43 SAMEA2569438.chr10_2.fastq.gz
```

- Open a file to see its contents using the (`less`) UNIX command:

```
    # go to the following directory
    m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/data

    # enter the `less` command followed by the file name you want to open:
    m_jan2020@mjan2020VirtualBox:~/course/data$ less SAMEA2569438.chr10_2.fastq.gz
    # press Ctrl+F to go forward one window
    # press Ctrl+B to go back one window
    # press 'q' if you want to exit
```

- Now, make (`less`) to print out the line number information by doing:

```
    m_jan2020@mjan2020VirtualBox:~/course/data$ less -N SAMEA2569438.chr10_2.fastq.gz
```

- Output redirection:
  In Linux, output redirection means that we can redirect the STDOUT of a given command to a file. For this, we use the (`>`) symbol.
  Example:

```
    # go to the following directory
    m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/data

    # the output of 'ls' will be redirected to the file named 'listing'
    m_jan2020@mjan2020VirtualBox:~/course/data$ ls -l > listing

    # examine the contents of 'listing'
     m_jan2020@mjan2020VirtualBox:~/course/data$ less listing
```

- Piping
  It is a form of redirection that transfers the standard output of one command/program to another command/program for further processing. The different commands in the pipe are connected using the pipe character (`|`). In this way we can combine 2 or more commands. Pipes are unidirectional, i.e. data flows from left to right.
  Examples:

```
    # go to the following directory
    m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/data

    # count the number of files in a directory
    m_jan2020@mjan2020VirtualBox:~$ ls | wc -l
```

Sequencing data for sample SAMEA2569438

In this course we are going to align 2 files in the FASTQ format to the rice reference genome. Each of the files contain the forward and the reverse reads from the paired-end sequencing experiment. These reads have been generated for the sample with id (`SAMEA2569438`) of the 3000 Rice genomes project (see here).
The first thing we are going to do is check the quality of the sequencing reads.

Quality control of the FASTQ files

For this, we will use FASTQC with our FASTQ files.
First, move to the directory containing the FASTQ files:

```
    m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/data
```

And check that the `FASTQ` files are there:

```
    m_jan2020@mjan2020VirtualBox:~/course/data$ ls -lh
    total 23M
    -rw-rw-r-- 1 m_jan2020 m_jan2020 183 Nov 26 17:44 cmd
    -rw-rw-r-- 1 m_jan2020 m_jan2020 11M Nov 26 17:43 SAMEA2569438.chr10_1.fastq.gz
    -rw-rw-r-- 1 m_jan2020 m_jan2020 12M Nov 26 17:43 SAMEA2569438.chr10_2.fastq.gz
```

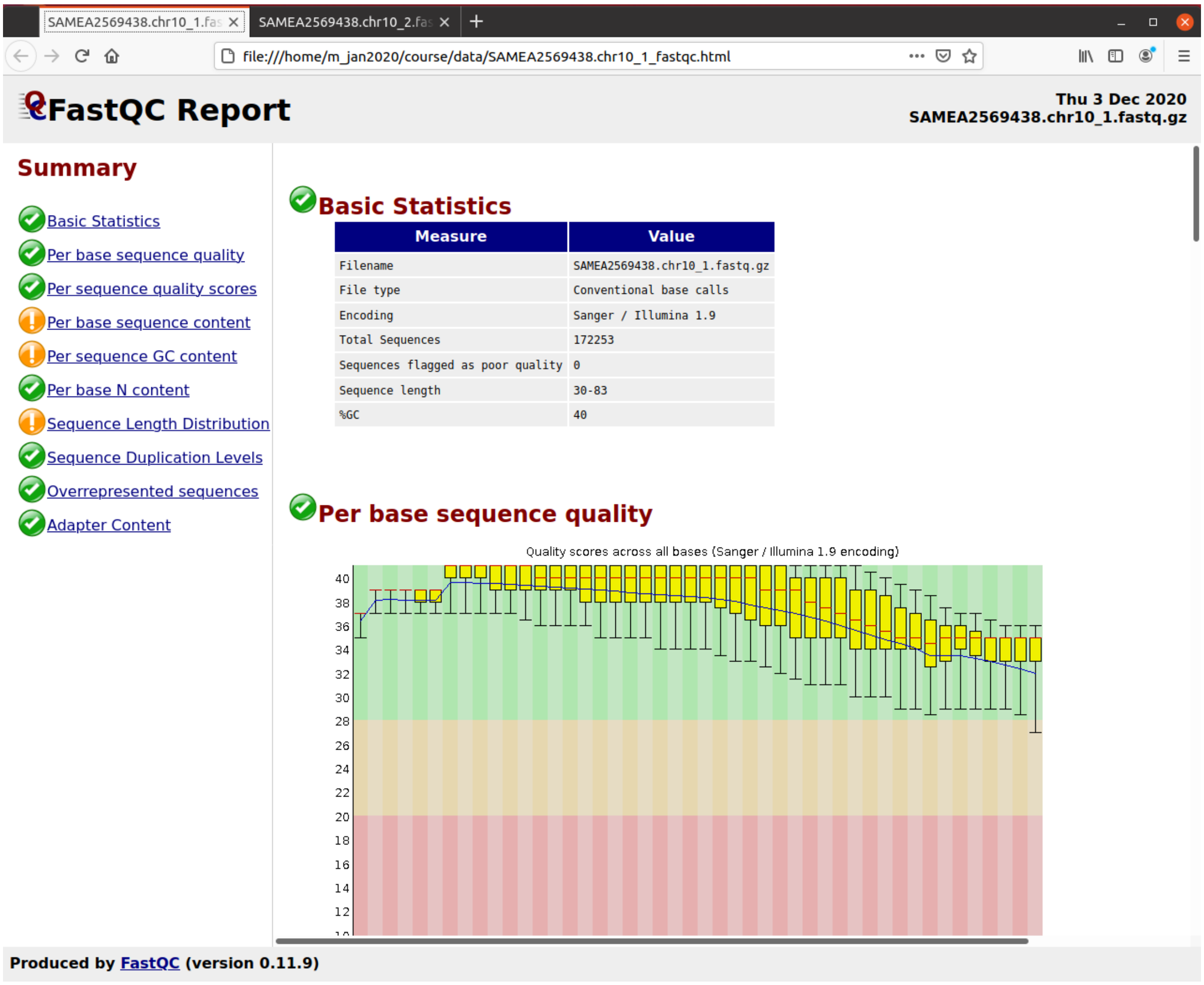
You can now run `FASTQC`:

```
    m_jan2020@mjan2020VirtualBox:~/course/data$ fastqc SAMEA2569438.chr10_1.fastq.gz SAMEA2569438.chr10_2.fastq.gz
```

And then open the two `html` files using `firefox`:

```
m_jan2020@mjan2020VirtualBox:~/course/data$ firefox SAMEA2569438.chr10_1_fastqc.html SAMEA2569438.chr10_2_fastqc.html
```

You should see something similar to:


SAMEA2569438.chr10_1.fas
SAMEA2569438.chr10_2.fas
file:///home/m_jan2020/course/data/SAMEA2569438.chr10_1_fastqc.html
FastQC Report
Thu 3 Dec 2020
SAMEA2569438.chr10_1.fastq.gz
Summary
Basic Statistics
Per base sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Basic Statistics
Per base sequence quality
Quality scores across all bases (Sanger / Illumina 1.9 encoding)
Produced by FastQC (version 0.11.9)

Measure	Value
Filename	SAMEA2569438.chr10_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	172253
Sequences flagged as poor quality	0
Sequence length	30-83
%GC	40

Alignment with BWA

Preparing the reference genome for Samtools

Samtools is going to be used in this course to manipulate the alignments generated by bwa, and Samtools needs the reference genome used for generating the alignments to be indexed. For this, move to the folder where we have prepared a reference file in FASTA format only for chromosome 10:

```
m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/reference
```

And now let's create a Samtools index file for this FASTA.
Samtools is going to be used a lot during this course and an index is necessary for quickly extracting a subsequence from the reference sequence:

```
samtools faidx Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
```

Check that you have a new file with the `.fai` suffix

```
m_jan2020@mjan2020VirtualBox:~/course/reference$ ls -lh
total 23M
-rw-rw-r-- 1 m_jan2020 m_jan2020 406K Nov 26 17:59 Oryza_sativa.IRGSP-1.0.48.chr10.gtf.gz
-rw-rw-r-- 1 m_jan2020 m_jan2020  23M Nov 23 11:23 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
-rw-rw-r-- 1 m_jan2020 m_jan2020   21 Dec  3 12:39 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.fai
```

Building an index for the reference genome

In this course we are going to use BWA for aligning the short reads to the reference. There are other reading mapping tools, but BWA is one of the most popular and with which we have the most experience in our group.
This tool requires a preprocessing step consisting on building an index for the same chromosome 10 FASTA sequence that has been mentioned previously.

To build this index, move to the directory that contains the FASTA file:

```
    m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/reference/
```

And enter the `bwa index` command in your terminal:

```
    m_jan2020@mjan2020VirtualBox:~$ bwa index Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
```

Building the index takes a while, but the good thing is that this index can be reused for different alignment runs.
Once `bwa index` has finished, check that the new files have been generated:

```
    m_jan2020@mjan2020VirtualBox:~/course/reference$ ls -lh
    -rw-rw-r-- 1 m_jan2020 m_jan2020  23M Nov 23 11:23 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
    -rw-rw-r-- 1 m_jan2020 m_jan2020  877 Dec  3 14:26 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.amb
    -rw-rw-r-- 1 m_jan2020 m_jan2020   89 Dec  3 14:26 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.ann
    -rw-rw-r-- 1 m_jan2020 m_jan2020  23M Dec  3 14:26 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.bwt
    -rw-rw-r-- 1 m_jan2020 m_jan2020   21 Dec  3 12:39 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.fai
    -rw-rw-r-- 1 m_jan2020 m_jan2020 5.6M Dec  3 14:26 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.pac
    -rw-rw-r-- 1 m_jan2020 m_jan2020  12M Dec  3 14:26 Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.sa
```

BWA mem

The BWA mapper has different execution options, this can be verified by entering the following in your terminal:

```
m_jan2020@mjan2020VirtualBox:~$ bwa
```

You will see something very similar to:

```
Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.17-r1188
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index         index sequences in the FASTA format
     mem           BWA-MEM algorithm
     fastmap       identify super-maximal exact matches
     pemerge       merge overlapping paired ends (EXPERIMENTAL)
     aln           gapped/ungapped alignment
     samse         generate alignment (single ended)
     sampe         generate alignment (paired ended)
     bwasw         BWA-SW for long queries

     shm           manage indices in shared memory
     fa2pac        convert FASTA to PAC format
     pac2bwt       generate BWT from PAC
     pac2bwtgen    alternative algorithm for generating BWT
     bwtupdate     update .bwt to the new format
     bwt2sa        generate SA from BWT and Occ

Note: To use BWA, you need to first index the genome with `bwa index'.
  There are three alignment algorithms in BWA: `mem', `bwasw', and
  `aln/samse/sampe'. If you are not sure which to use, try `bwa mem'
  first. Please `man ./bwa.1' for the manual.
```

It is widely accepted that the alignment option used for reads that are 70 bp to 1Mb in length is `BWA mem`, as it is more accurate and faster than the other alignment options included with `BWA`.

To run `BWA`, first you need to move to the directory where the output will be generated:

```
    m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/alignment
```

Now, enter the following to get an alignment in the BAM format:

```
    m_jan2020@mjan2020VirtualBox:~$ bwa mem /home/m_jan2020/course/reference/Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
/home/m_jan2020/course/data/SAMEA2569438.chr10_1.fastq.gz /home/m_jan2020/course/data/SAMEA2569438.chr10_2.fastq.gz |samtools view
-b -o SAMEA2569438.chr10.bam
```

You can verify that `BWA` is running by using the `top` Unix command. For this, first check what your username is on your machine:

```
m_jan2020@mjan2020VirtualBox:~$ whoami
m_jan2020
```

Now, you can use the `top` command to obtain a summary of the processes that user `m_jan2020` is running in the system:

```
top -u m_jan2020
```

You should see something similar to what I obtain in my system:

```
top - 17:10:34 up  2:43,  1 user,  load average: 1.57, 0.69, 0.70
Tasks: 180 total,   1 running, 179 sleeping,   0 stopped,   0 zombie
%Cpu(s): 96.3 us,  3.7 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    981.3 total,     68.3 free,    549.9 used,    363.1 buff/cache
MiB Swap:    923.3 total,    370.3 free,    553.0 used.    272.0 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
58337 m_jan2020    20   0  265416 165348   2832 S  92.0  16.5   1:20.36 bwa
4401 m_jan2020    20   0   10872   2868   1792 S   0.0   0.3   0:00.65 bash
58338 m_jan2020    20   0   24204   9796   8320 S   0.0   1.0   0:04.89 samtools
```

If you want to obtain an alignment file in the `SAM` format enter the following:

```
    m_jan2020@mjan2020VirtualBox:~$ bwa mem /home/m_jan2020/course/reference/Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
/home/m_jan2020/course/data/SAMEA2569438.chr10_1.fastq.gz /home/m_jan2020/course/data/SAMEA2569438.chr10_2.fastq.gz -o
SAMEA2569438.chr10.sam
```

Finally, if you want to generate an alignment in the CRAM format, do the following:

```
    m_jan2020@mjan2020VirtualBox:~$ bwa mem /home/m_jan2020/course/reference/Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
/home/m_jan2020/course/data/SAMEA2569438.chr10_1.fastq.gz /home/m_jan2020/course/data/SAMEA2569438.chr10_2.fastq.gz |samtools view
-C -T /home/m_jan2020/course/reference/Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa > SAMEA2569438.cram
```

The SAM alignment format

The Sequence Alignment Format (SAM) is a text-based format (check the Wikipedia entry for more information), used for representing the alignment of the short reads in the FASTQ files to the reference sequence. This format is not compressed and it is preferable to convert it to its binary equivalent (BAM) or to its ultra-compressed equivalent called CRAM to facilitate its handling and storage.

Compare the diferent file sizes for each of the alignment files generated in the previous section by listing the directory contents:

```
m_jan2020@mjan2020VirtualBox:~$ ls -lh
```

And you get:

```
total 135M
drwxrwxr-x 3 m_jan2020 m_jan2020 4.0K Nov 26 17:52 postprocessing
-rw-rw-r-- 1 m_jan2020 m_jan2020  32M Dec  3 14:46 SAMEA2569438.chr10.bam
-rw-rw-r-- 1 m_jan2020 m_jan2020  91M Dec  3 14:48 SAMEA2569438.chr10.sam
-rw-rw-r-- 1 m_jan2020 m_jan2020  13M Dec  3 14:50 SAMEA2569438.cram
```

The alignment files in the `SAM` format (click here for the format specification) start with an optional header section followed by the alignment lines.

All header lines start with '@' and are used to represent different metadata elements such as the reference version and the chromosome sequence IDs used in the alignment, the technology used to generate the sequence data, if the alignment file is sorted or not, etc ...

The alignment lines are characterized by having 11 mandatory fields:

Col	Field	Type	Brief description
1	QNAME	String	Query template NAME
2	FLAG	int	bitwise FLAG
3	RNAME	String	References sequence NAME
4	POS	int	1- based leftmost mapping POSition
5	MAPQ	int	Mapping quality
6	CIGAR	String	CIGAR string
7	RNEXT	String	Ref. name of the mate/next read
8	PNEXT	int	Position of the mate/next read

Col	Field	Type	Brief description
9	TLEN	int	observed template length
10	SEQ	String	segment sequence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33

Samtools

Samtools is a command line tool used to interact and manipulate the SAM-related alignment files.

You can practise the commands that are more relevant for this course by going to the directory where `bwa` was run:

```
m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/alignment
```

And entering the following commands:

- print the alignment to stdout

  ```
  m_jan2020@mjan2020VirtualBox:~$ samtools view SAMEA2569438.chr10.bam | less
  ```

- print the header section

  ```
   m_jan2020@mjan2020VirtualBox:~$ samtools view -H SAMEA2569438.chr10.bam
  ```

- And if you want to get the alignments in a particular genomic region:

  ```
   m_jan2020@mjan2020VirtualBox:~$ samtools view SAMEA2569438.chr10.bam 10:10000000-10001000
  [main_samview] random alignment retrieval only works for indexed BAM or CRAM files.
  ```

This does not work because first you need to sort and build an index for the `bam` file.
To do this, we are going to move to a different folder to keep things in order:

```
 m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/alignment/postprocessing

 m_jan2020@mjan2020VirtualBox:~$ samtools sort ../SAMEA2569438.chr10.bam -o SAMEA2569438.chr10.sorted.bam -O BAM
```

Note the `../` in the command, this means that the `bam` file is one directory up.

Now, build an index for the new sorted `bam` file:

```
 m_jan2020@mjan2020VirtualBox:~$ samtools index SAMEA2569438.chr10.sorted.bam
```

Finally, repeat the `samtools view` command to print a particular region:

```
 m_jan2020@mjan2020VirtualBox:~$ samtools view SAMEA2569438.chr10.sorted.bam 10:10000000-10001000
```

- Some basic stats on the alignment

You can use the `flagstat` command for this:

```
m_jan2020@mjan2020VirtualBox:~$ samtools flagstat SAMEA2569438.chr10.sorted.bam
```

And you get:

```
344702 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
196 + 0 supplementary
0 + 0 duplicates
342947 + 0 mapped (99.49% : N/A)
344506 + 0 paired in sequencing
172253 + 0 read1
172253 + 0 read2
332224 + 0 properly paired (96.43% : N/A)
340996 + 0 with itself and mate mapped
1755 + 0 singletons (0.51% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

The explanation of this report can be found in the samtools stats manual page.

- Print the base coverage per position

For this, we can use `samtools mpileup`. The output of this command is explained here.
You can run `samtools mpileup` by doing:

```
    m_jan2020@mjan2020VirtualBox:~$ samtools mpileup -f /home/m_jan2020/course/reference/Oryza_sativa.IRGSP-
1.0.dna.toplevel.chr10.fa SAMEA2569438.chr10.sorted.bam |less
```

Now let's try a command combination that is more complex to print all positions having a coverage >=10:

```
    m_jan2020@mjan2020VirtualBox:~$ samtools mpileup -f /home/m_jan2020/course/reference/Oryza_sativa.IRGSP-
1.0.dna.toplevel.chr10.fa SAMEA2569438.chr10.sorted.bam | awk '{if ($4>=20) print}' | less
```

Alignment post-processing

The alignment file in the BAM format needs a series of post-processing steps that are required for variant discovery. The different steps that are shown here will produce an analysis-ready BAM file that can be used in the following section of this course.

Adding metadata to the alignment file

BWA generates a BAM file without metadata information about the experimental design that has been used during the sequencing, this is why we need to manually add this metadata so it can be used during the variant calling analysis described in the variant calling section of this course. To do this, we are going to use Picard AddOrReplaceReadGroups in the following way:

```
    m_jan2020@mjan2020VirtualBox:~$ picard AddOrReplaceReadGroups I=SAMEA2569438.chr10.sorted.bam RGSM=SAMEA2569438
RGLB=SAMEA2569438 RGPL=ILLUMINA O=SAMEA2569438.chr10.sorted.reheaded.bam RGPU=SAMEA2569438
```

This command will add information about the sample id that has been sequenced, which sequencing platform has been used and also information about the sequencing library id. You can verify that the SAM header contains this metadata information by doing:

```
    m_jan2020@mjan2020VirtualBox:~$ samtools view -H SAMEA2569438.chr10.sorted.reheaded.bam
```

And you will see the new metadata in the line starting with `@RG`:

```
    @HD     VN:1.6  SO:unsorted
    @SQ     SN:10   LN:23207287
    @RG     ID:1    LB:SAMEA2569438 PL:ILLUMINA     SM:SAMEA2569438 PU:SAMEA2569438
    @PG     ID:bwa  PN:bwa  VN:0.7.17-r1188 CL:bwa mem Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa SAMEA2569438.chr10_1.fastq.gz
SAMEA2569438.chr10_2.fastq.gz
```

MarkDuplicates

The alignment file we have generated using `BWA` may contain duplicate reads. These reads are originated in the PCR amplification step during the library preparation and might produce an over-representation of identical reads from the same DNA fragment. These duplicate reads must be identified and marked so they can be correctly handled by the variant calling tool. There are multiple tools available to handle these duplicates, in this course we will use Picard MarkDuplicates, which is one of the most reliable.

```
     m_jan2020@mjan2020VirtualBox:~$ picard MarkDuplicates -Xms1g I=SAMEA2569438.chr10.sorted.reheaded.bam
O=SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.bam M=SAMEA2569438.chr10.metrics.txt
```

We use the `M` option so that `MarkDuplicates` generates a text file with metrics on the number of reads duplicates. This file is named `SAMEA2569438.chr10.metrics.txt` in this case.

Let's open this metrics file

```
     m_jan2020@mjan2020VirtualBox:~$ less SAMEA2569438.chr10.metrics.txt
```

The first part of the file is the most relevant for us:

```
    ## METRICS CLASS        picard.sam.DuplicationMetrics
    LIBRARY UNPAIRED_READS_EXAMINED READ_PAIRS_EXAMINED     SECONDARY_OR_SUPPLEMENTARY_RDS  UNMAPPED_READS
UNPAIRED_READ_DUPLICATES        READ_PAIR_DUPLICATES    READ_PAIR_OPTICAL_DUPLICATES    PERCENT_DUPLICATION
ESTIMATED_LIBRARY_SIZE
    SAMEA2569438    1755    170498  196     1755    133     5827    0       0.034389        2437223
    ...
```

Finally, we print the reads that are duplicates using `samtools view` in combination with the bitwise FLAG value in the second column that selects the PCR duplicates:

```
     m_jan2020@mjan2020VirtualBox:~$ samtools view -f 1024 SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.bam |less
```

Viewing the aligned reads using IGV

The Integrative Genomics Viewer (IGV) is a useful interactive tool to explore visually the genomic data resulting from your analysis. We are going to use it in this section of the course to display the alignments we have generated.
In this example, we will fetch the alignments for a specific region in chromosome 10.

Selecting the alignments for a specific genomic sub-region

First, move to the following directory to select the alignments falling in a given sub-region from the file generated in the previous section (`SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.bam`):

```
    m_jan2020@mjan2020VirtualBox:~$ cd /home/m_jan2020/course/alignment/aln_visualization
```

And again, we will use `samtools view` together with the region we want to extract. Remember that this command requires an indexed BAM file. So first we have to build an index for the file:

```
    m_jan2020@mjan2020VirtualBox:~$ samtools index
/home/m_jan2020/course/alignment/postprocessing/SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.bam
```

Then we will select all the alignments in the region with coordinates `10:10000000-11000000` that are correct, which means that both members of the read pair are mapped correctly. For this, we use the SAM bitwise flag = 2 and `samtools view`:

```
    m_jan2020@mjan2020VirtualBox:~$ samtools view -f 2
/home/m_jan2020/course/alignment/postprocessing/SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.bam 10:10000000-11000000 -b -o
SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam
```

And now we need to build an index for the new BAM, as IGV needs it to quickly retrieve the aligments to display:

```
    m_jan2020@mjan2020VirtualBox:~$ samtools index SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam
```

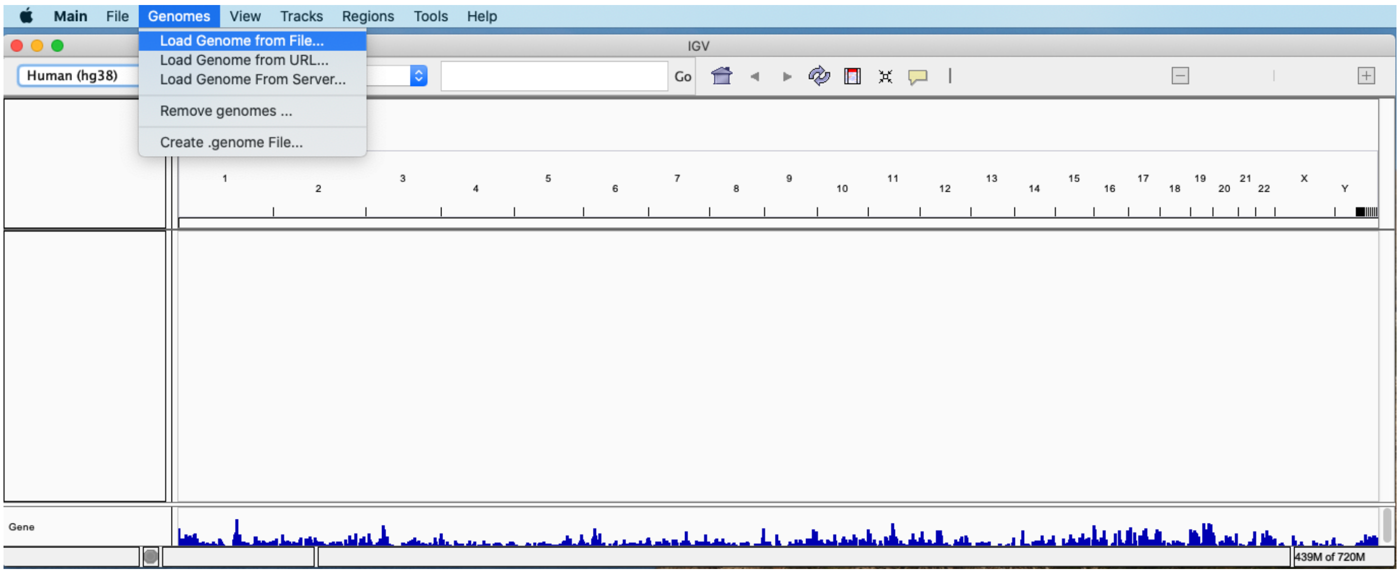
Using IGV with the new alignment file

In this section we will use IGV to explore the alignments in the file created in the previous section named `SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam`.

Open `IGV` by going to your terminal and entering:

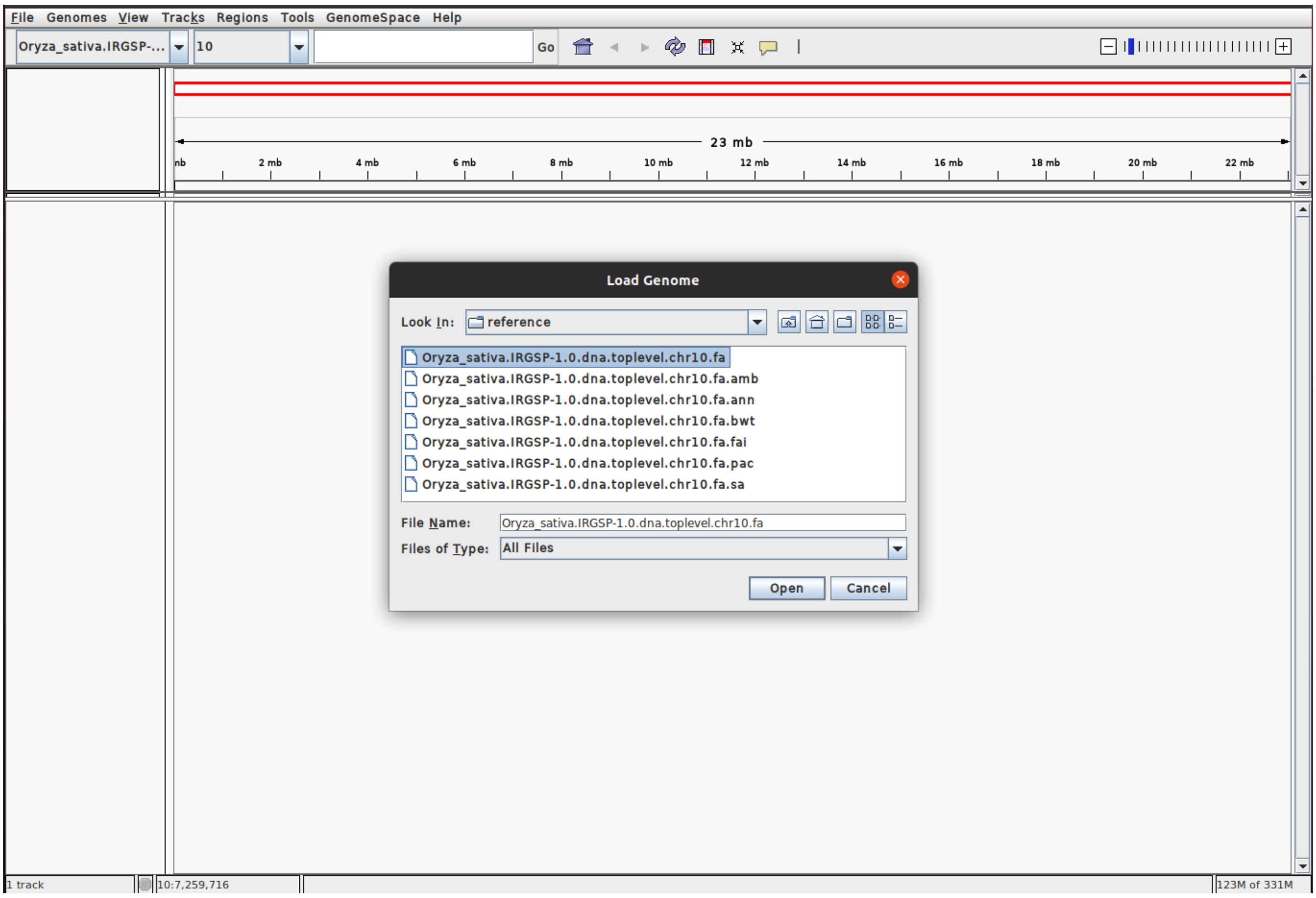
```
    igv
```

Once `IGV` is open, you will need to load the FASTA file containing the chromosome 10 sequence for rice, as this sequence is not included by default in `IGV`:

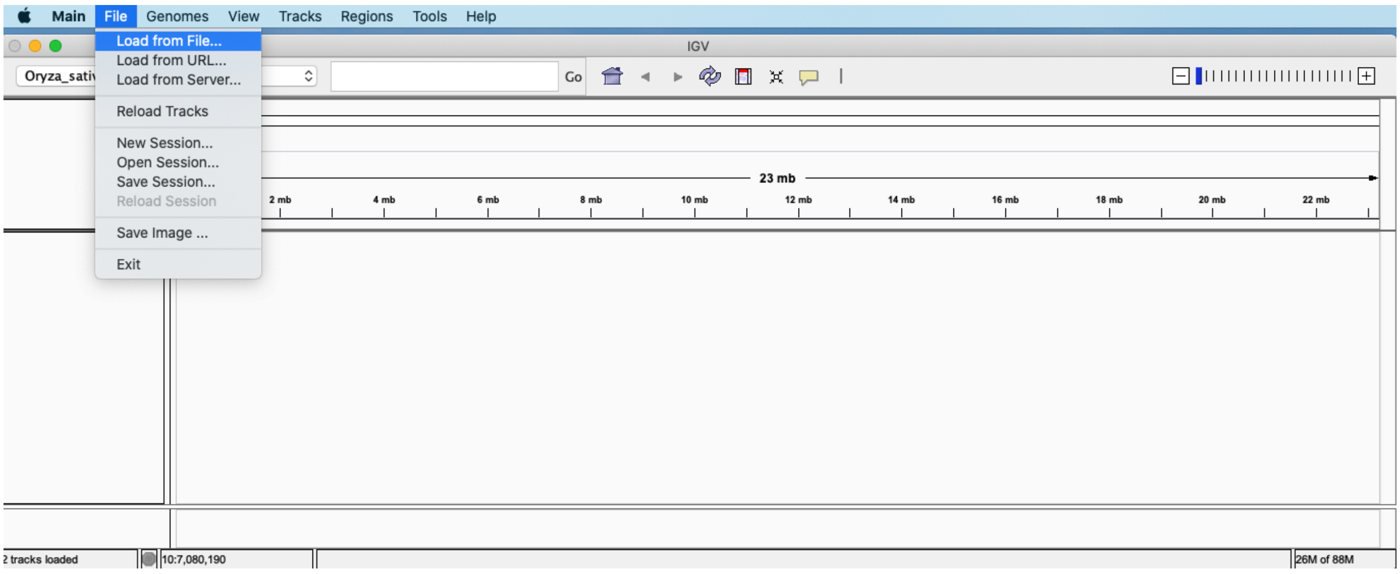


Look for you file by going to the folder named (`m_jan2020`) and clicking on the different folders until you find the FASTA file named `Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa`:

```
    course->reference->Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
```

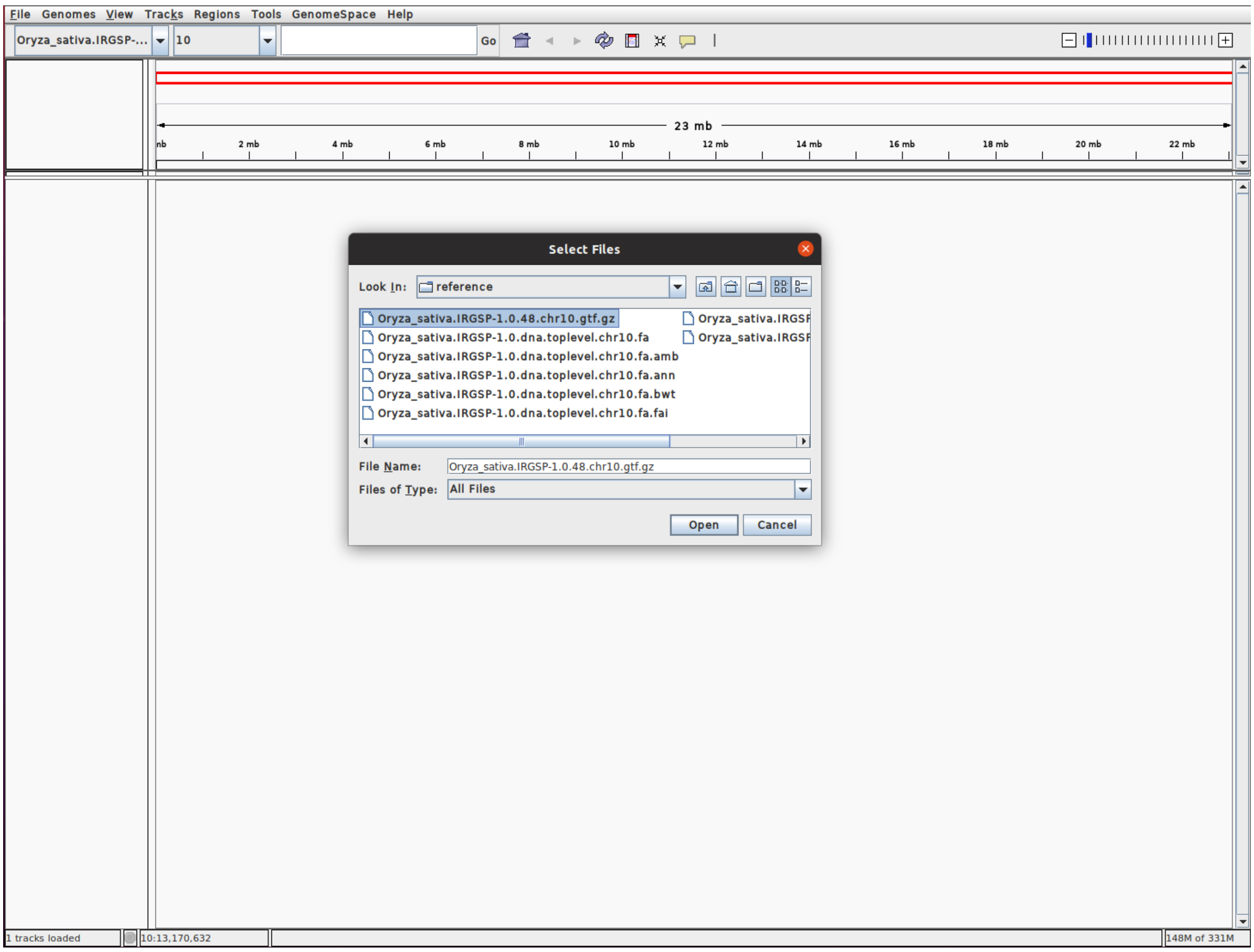

File Genomes View Tracks Regions Tools GenomeSpace Help
Oryza_sativa.IRGSP-...
10
Go
23 mb
Load Genome
Look In: reference
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.amb
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.ann
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.bwt
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.fai
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.pac
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.sa
File Name: Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
Files of Type: All Files
Open
Cancel
1 track
10:7,259,716
123M of 331M

Now, load the GTF file containing the rice gene annotations for chromosome 10:



The GTF file can be found by going to the folder named (`m_jan2020`) and clicking on the different folders until you find the GTF file named `Oryza_sativa.IRGSP-1.0.48.chr10.gtf.gz`:

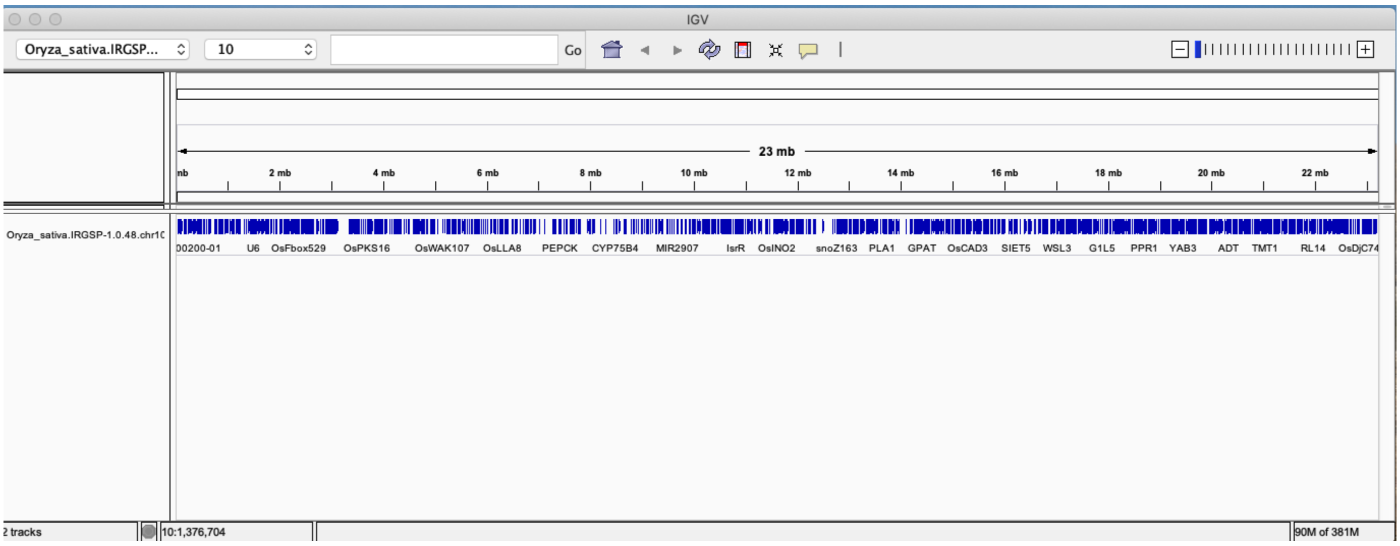
```
course->reference->Oryza_sativa.IRGSP-1.0.48.chr10.gtf.gz
```



Oryza_sativa.IRGSP-1.0.48.chr10.gtf.gz
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.amb
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.ann
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.bwt
Oryza_sativa.IRGSP-1.0.dna.toplevel.chr10.fa.fai

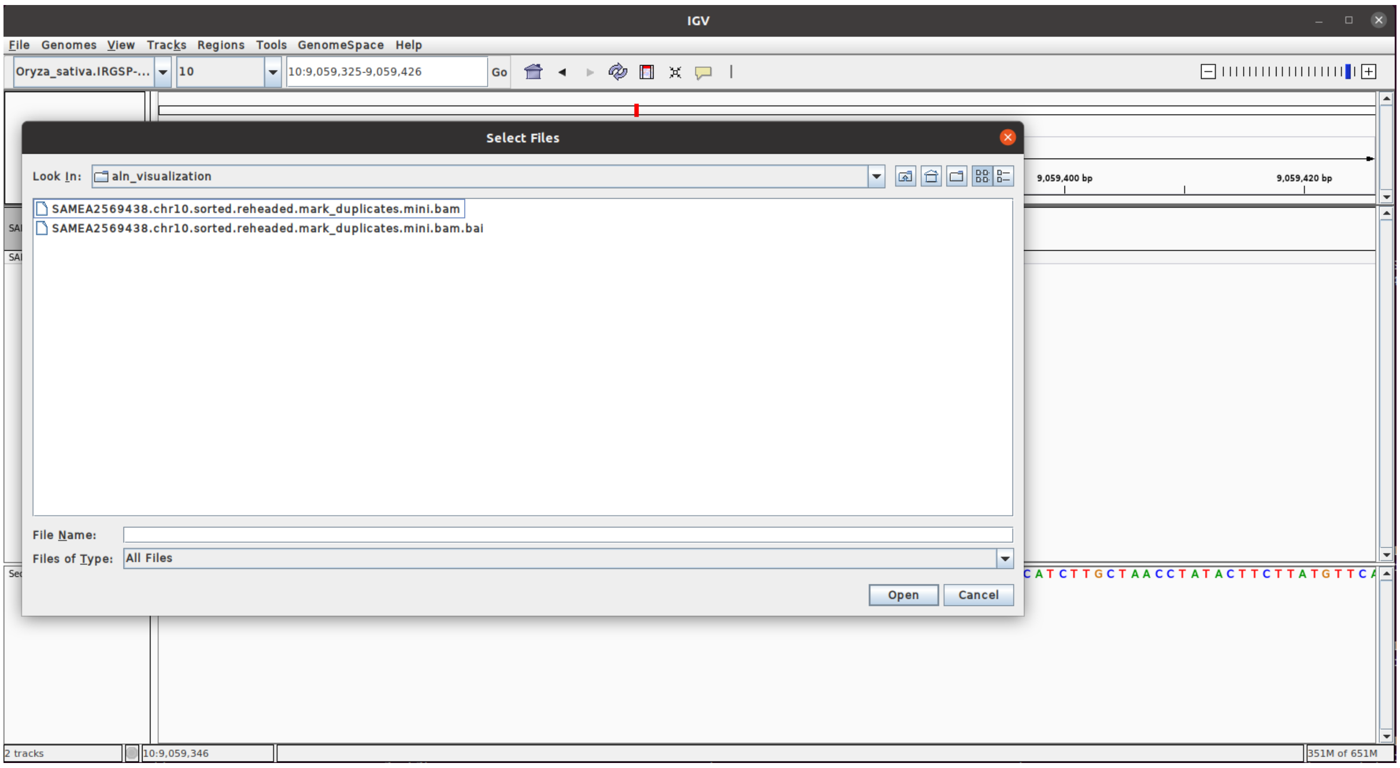
File Name: Oryza_sativa.IRGSP-1.0.48.chr10.gtf.gz
Files of Type: All Files

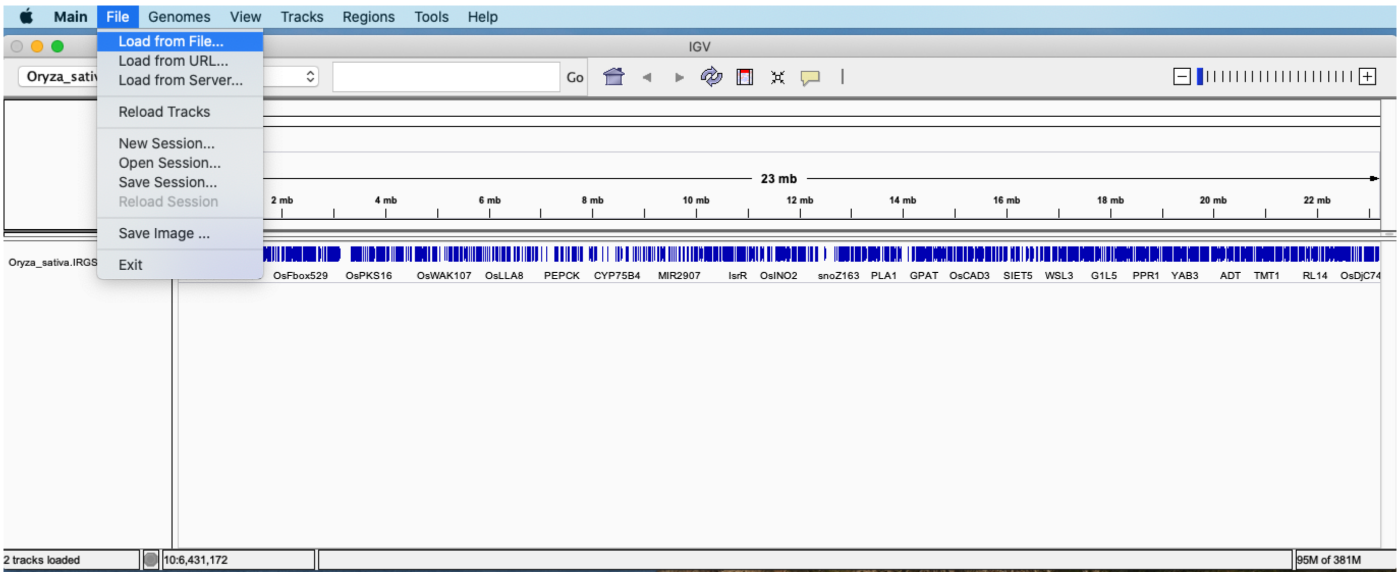
You will see a new track with all the genes annotated in chromosome 10:


IGV
Oryza_sativa.IRGSP...
10
Go
23 mb
Oryza_sativa.IRGSP-1.0.48.chr10
00200-01 U6 OsFbox529 OsPKS16 OsWAK107 OsLLA8 PEPCK CYP75B4 MIR2907 IsfR OsINO2 snoZ163 PLA1 GPAT OsCAD3 SIET5 WSL3 G1L5 PPR1 YAB3 ADT TMT1 RL14 OsDjC74
2 tracks
10:1,376,704
90M of 381M

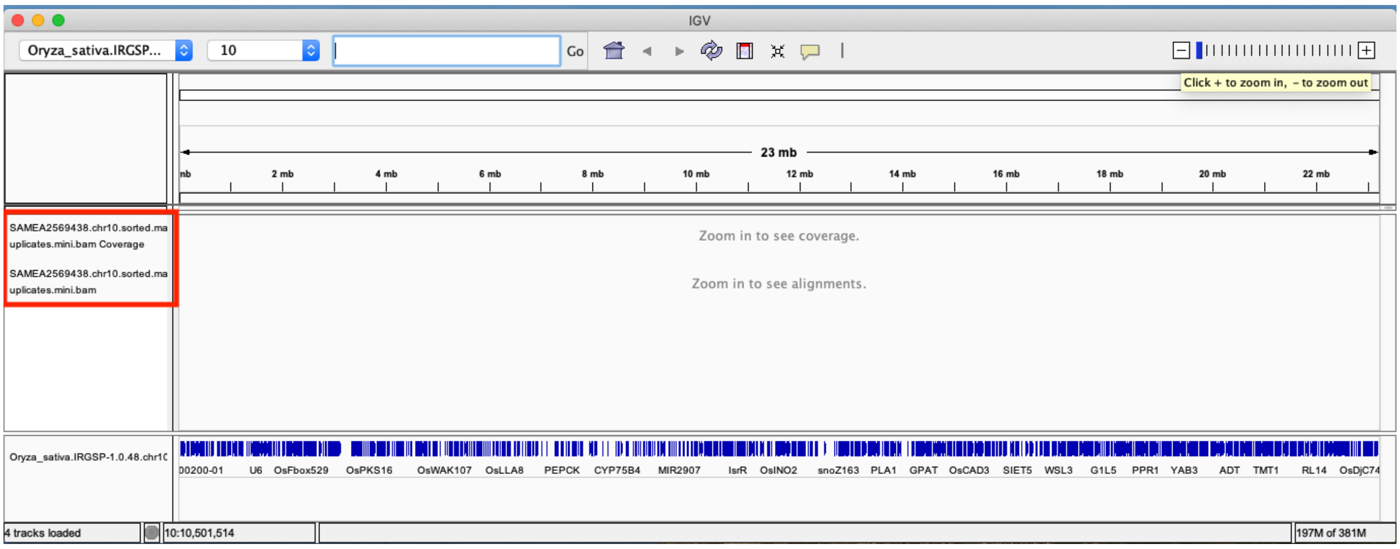
Now, load the BAM file with your alignments by going to the folder named (`m_jan2020`) and clicking on the different folders until you find the BAM file named `SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam`:

```
course->alignment->aln_visualization->SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam
```

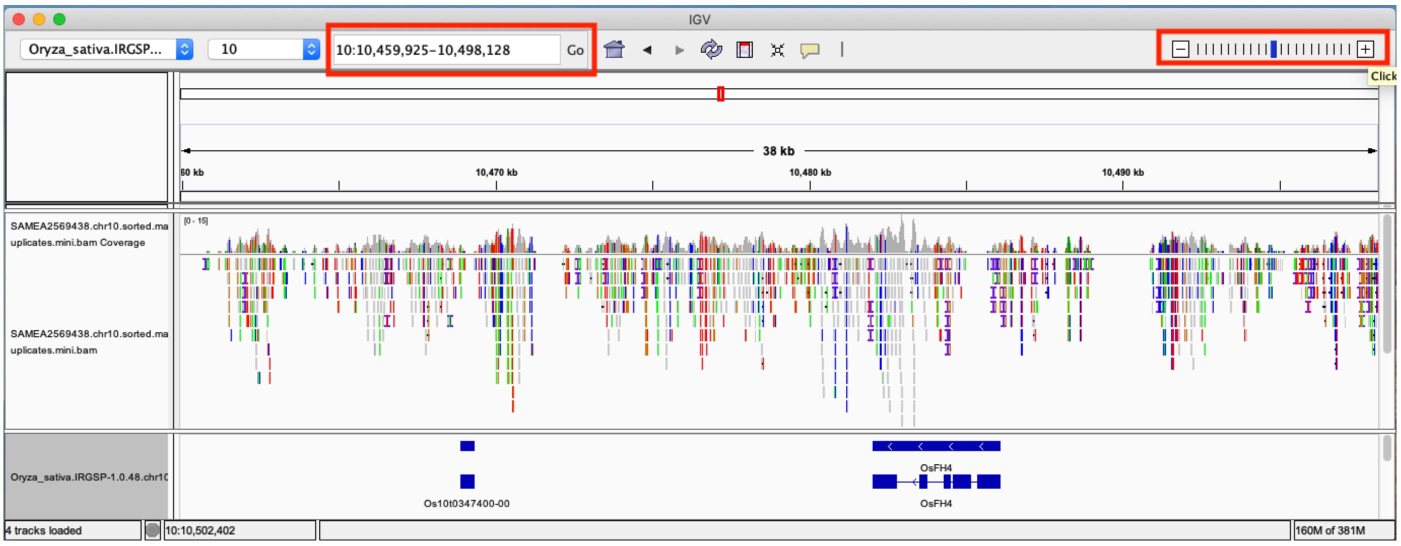

Select Files
Look In: aln_visualization
SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam
SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam.bai
File Name:
Files of Type: All Files
Open
Cancel



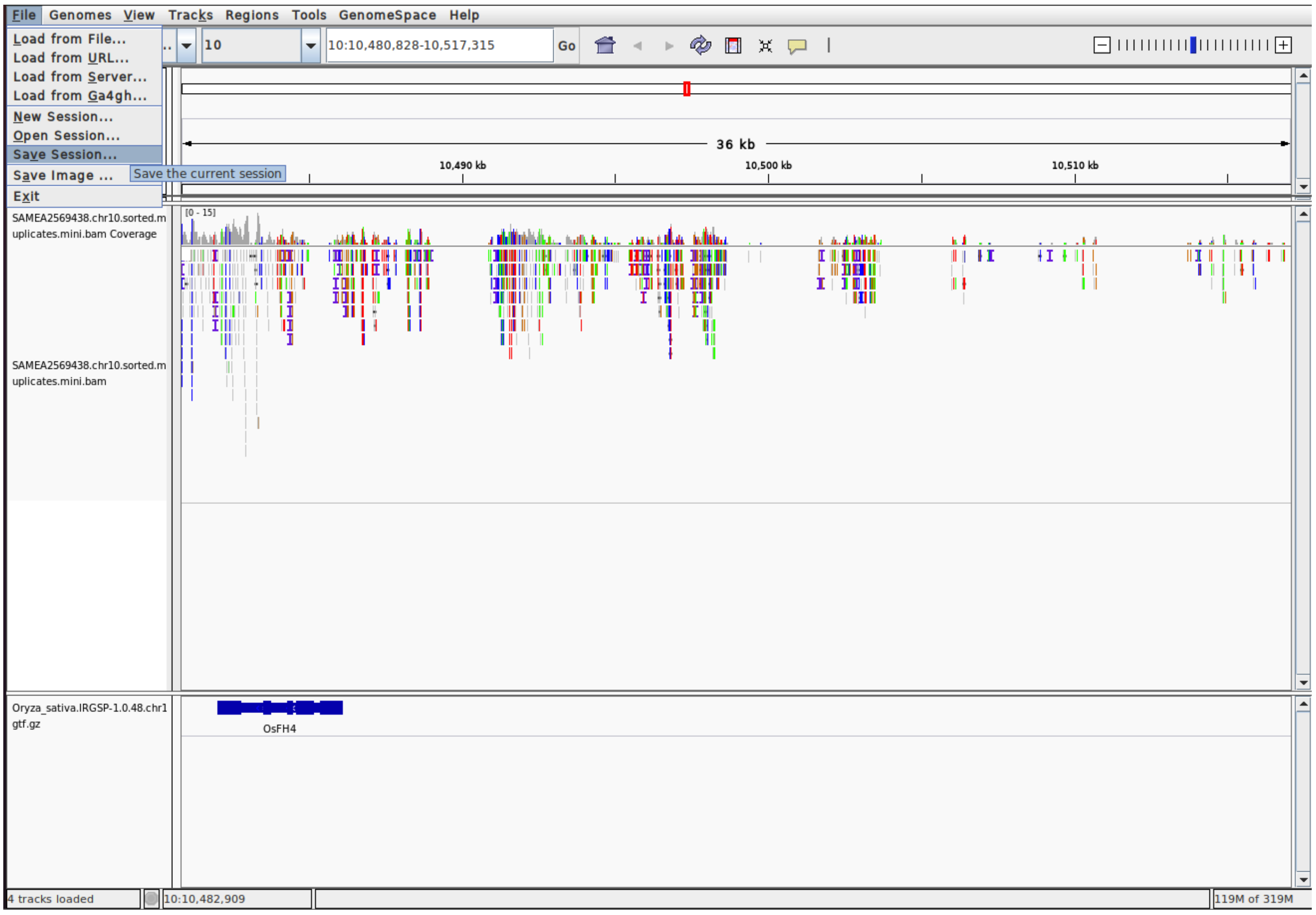
And you will see two new tracks, one with the alignments and the other with the depth of coverage. The alignments do not appear yet, as the region is too large to render the data.

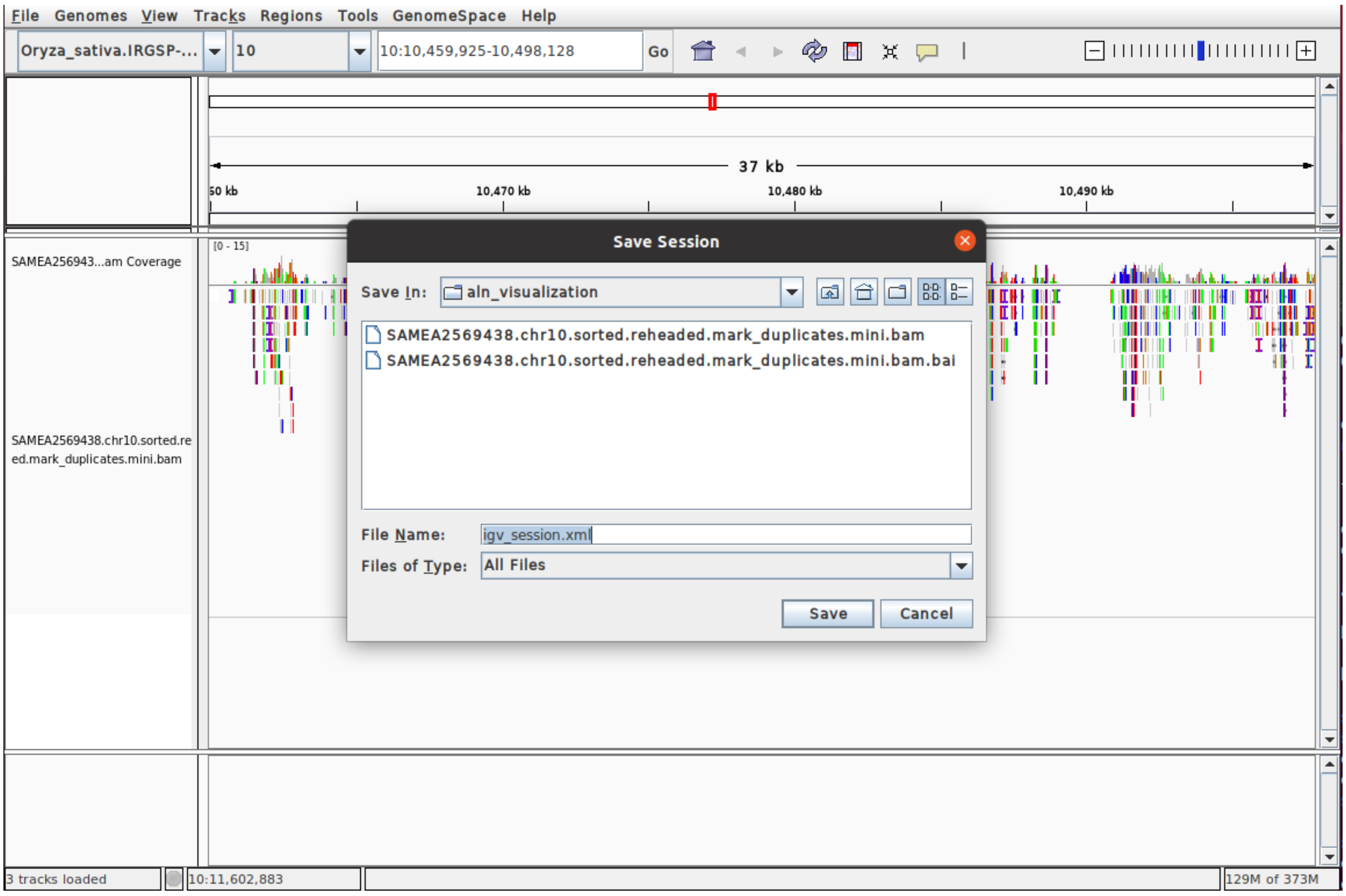

SAMEA2569438.chr10.sorted.mark_duplicates.mini.bam Coverage
SAMEA2569438.chr10.sorted.mark_duplicates.mini.bam
Zoom in to see coverage.
Zoom in to see alignments.
Click + to zoom in, – to zoom out

Enter the genomic interval you want to display in the blank search box or press the '-' or '+' keys to zoom in or out:



When you are done working, you can save the session by doing:





Save In: aln_visualization

SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam
SAMEA2569438.chr10.sorted.reheaded.mark_duplicates.mini.bam.bai

File Name: igv_session.xml
Files of Type: All Files

Save