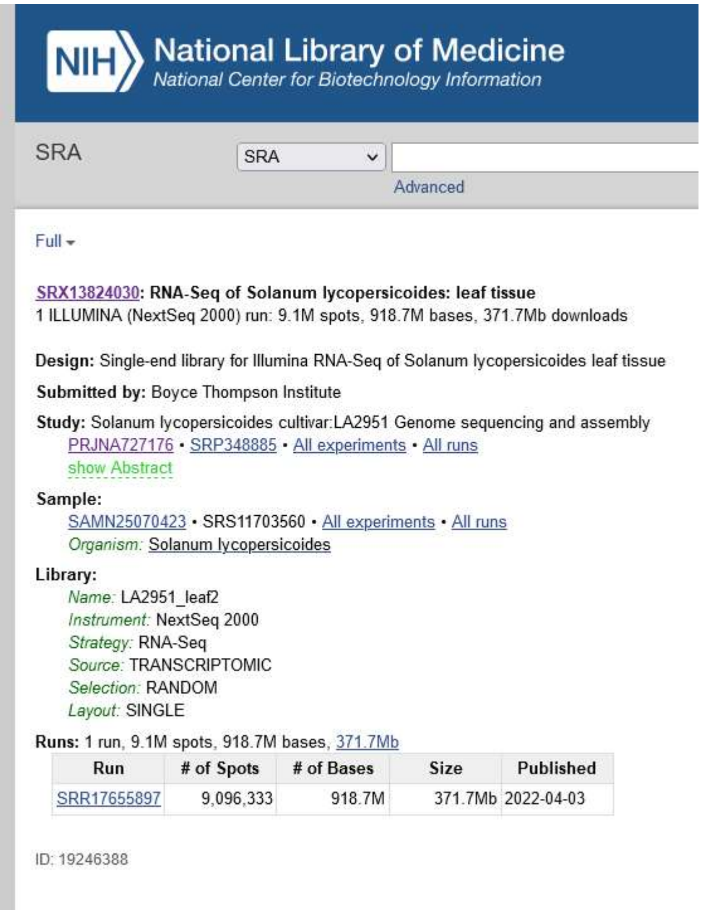
Figure 3: tomato short read archive (SRA) in the NCBI genome section


NIH National Library of Medicine
National Center for Biotechnology Information
SRA
SRA
Advanced
Full

SRX13824030: **RNA-Seq of Solanum lycopersicoides: leaf tissue**
1 ILLUMINA (NextSeq 2000) run: 9.1M spots, 918.7M bases, 371.7Mb downloads

**Design:** Single-end library for Illumina RNA-Seq of Solanum lycopersicoides leaf tissue

**Submitted by:** Boyce Thompson Institute

**Study:** Solanum lycopersicoides cultivar:LA2951 Genome sequencing and assembly
PRJNA727176 • SRP348885 • All experiments • All runs
show Abstract

**Sample:**
SAMN25070423 • SRS11703560 • All experiments • All runs
*Organism:* Solanum lycopersicoides

**Library:**
*Name:* LA2951_leaf2
*Instrument:* NextSeq 2000
*Strategy:* RNA-Seq
*Source:* TRANSCRIPTOMIC
*Selection:* RANDOM
*Layout:* SINGLE

**Runs:** 1 run, 9.1M spots, 918.7M bases, 371.7Mb

| Run | # of Spots | # of Bases | Size | Published |
|---|---|---|---|---|
| SRR17655897 | 9,096,333 | 918.7M | 371.7Mb | 2022-04-03 |

ID: 19246388

Figure 4: specific read set for *Solanum lycopersicoides*

## Student Exercises Sequence Database:

1. Search for the genomes of *Solanum lycopersicum, Solanum pennellii* and *Solanum lycopersicoides*. Compare the differences.
2. Explore the different search options and filters available in the SRA or ENA database. Use these options to find all the sequencing data available for a specific gene or genomic region.
3. Find all the RNA-seq datasets available for *Solanum lycopersicoides* in the SRA or ENA database. Filter the results based on a specific tissue (e.g. leaf)
4. Locate the data for SRX13824030
5. Gather all metadata for the read sets that you downloaded

## Day 3/4 Quality Control

# FastQC

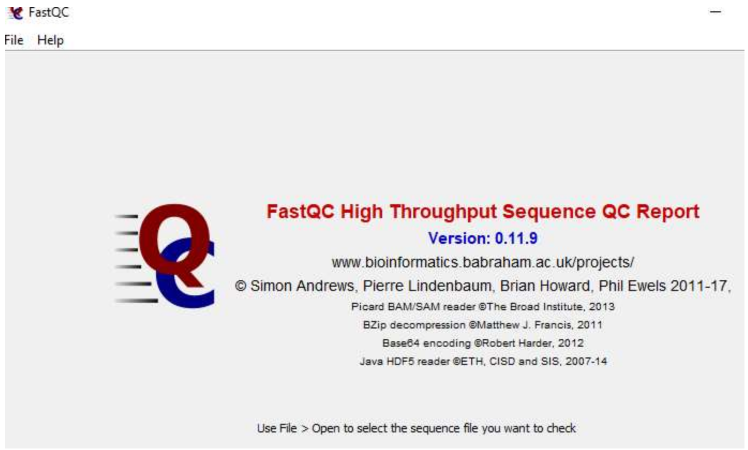
# Getting java

or https://adoptium.net/de/

If you use Mac it offers you x64 and aarc64;  x64 are **older** Macs aarch64 are **new** Mac with M1 M2 etc processors. On windows choose x64

## https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

FastQC is a widely used tool for assessing the quality of Next-Generation Sequencing (NGS) data, including RNA-seq data. It provides a comprehensive set of quality metrics that can help identify issues that may affect downstream analysis, such as low sequencing quality, adapter contamination, and overrepresented sequences. FastQC is a free, open-source software package that can be easily installed and run on most operating systems.



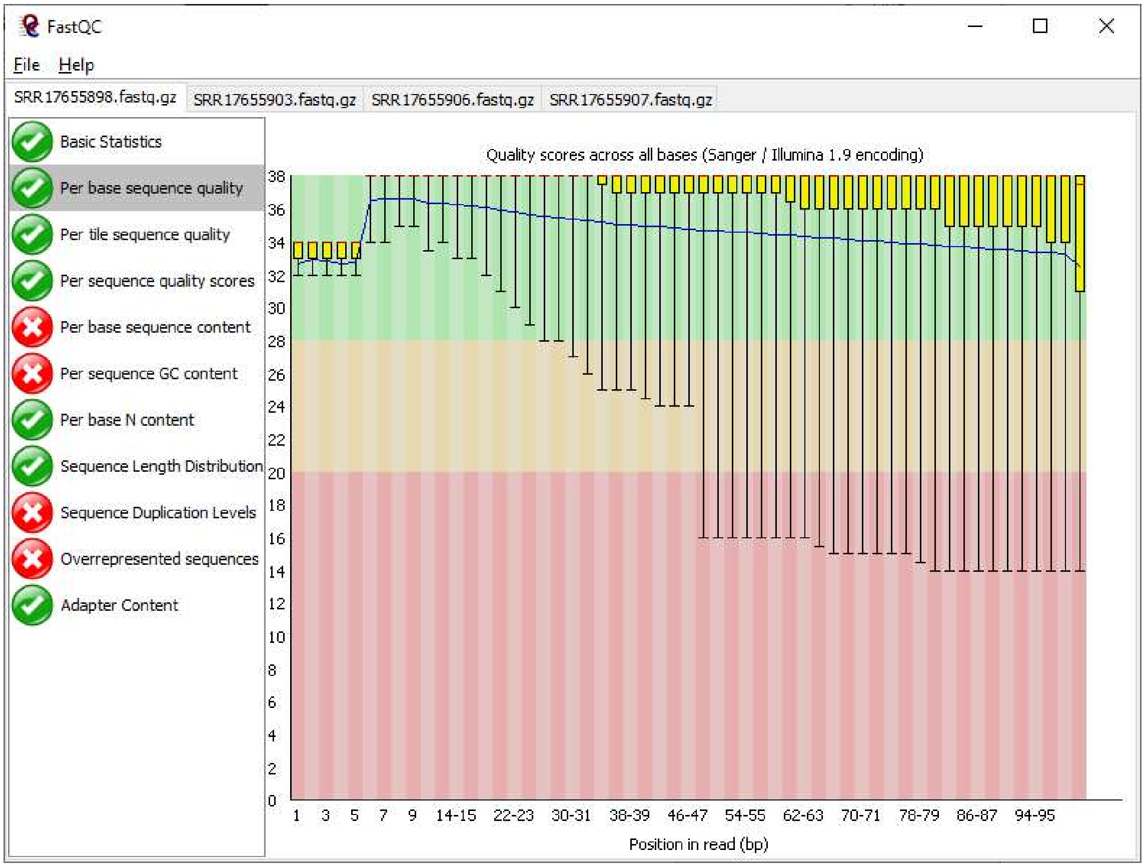
FastQC High Throughput Sequence QC Report
Version: 0.11.9
www.bioinformatics.babraham.ac.uk/projects/
© Simon Andrews, Pierre Lindenbaum, Brian Howard, Phil Ewels 2011-17,
Picard BAM/SAM reader ©The Broad Institute, 2013
BZip decompression ©Matthew J. Francis, 2011
Base64 encoding ©Robert Harder, 2012
Java HDF5 reader ©ETH, CISD and SIS, 2007-14

Use File > Open to select the sequence file you want to check

To use FastQC, the first step is to launch the program and load your dataset. Like most other tools working on sequencing files, FastQC allows for compressed (gz) files. After the data has been loaded, FastQC will perform a series of quality control checks and generate a detailed report with visualizations that allow you to quickly assess the quality of your data.

If FastQC doesn't see your WSL2 linux folder. Go there once in the File explorer so it appears in your recent items. Then select the recent one in FASTQC.

**Or in the file explorer copy the full file path** (it starts with \\wsl or similar and add it as a file and hit return in FastQC



- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile 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

**CHECK: Can you run FastQC and does it run on all files on all files?**

# Preprocessing reads: adapter trimming and quality filtering

## Trimmomatic

Trimmomatic is a widely used tool for trimming adapter and PCR primer sequences from DNA and RNA-seq reads. It provides several different functions, including adapter clipping, quality clipping, and discarding reads below a certain quality or length threshold. Trimmomatic is a modular tool, allowing users to perform only the specific functions they require, which can save time and computational resources.

To use Trimmomatic, the first step is to determine which library preparation kit was used and whether the library was run in **paired-end** or **single-end mode**. This information is necessary to configure the trimming parameters appropriately.

Trimmomatic includes a variety of options and parameters that can be customized to suit your specific needs. Some commonly used options include:

Trimmomatic is typically run as a command-line tool, with options and parameters specified in the command line.

```
cd
cd course
wget http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/Trimmomatic-0.39.zip
#Mac
curl http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/Trimmomatic-0.39.zip --output Trimmomatic-0.39.zip
#both
unzip Trimmomatic-0.39.zip
ls
```

you might have to install stuff ! Here we need java in our ubuntu machine if you are working on Mac or Linux you should by now have java installed already. **In ubuntu on windows type**

```
sudo apt install default-jre
```

if this doesn't work do sudo apt update first your ubuntu might be out of sync

For example, to trim adapter sequences and remove low-quality bases from single-end RNA-seq reads (**SE**), keeping at least reads of length 70 (**MINLEN:70**) and removing the first 5 bases (**HEADCROP 5)** from the beginning and apply a quality cutter **(SLIDINGWINDOW:4:15**) you might use the following command:

Everything needs to be in one single line

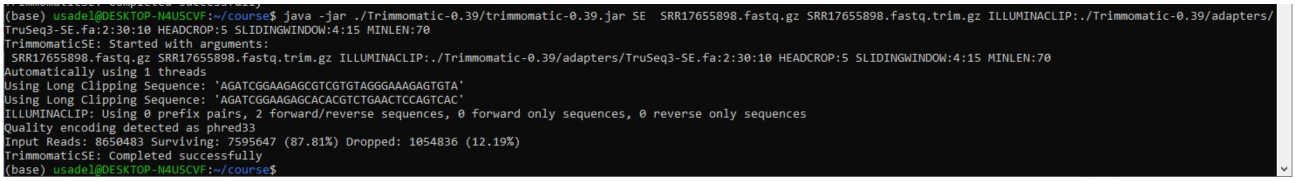
```
java -jar ./Trimmomatic-0.39/trimmomatic-0.39.jar SE  SRR17655898.fastq.gz SRR17655898.trim.fastq.gz
ILLUMINACLIP:./Trimmomatic-0.39/adapters/TruSeq3-SE.fa:2:30:10 HEADCROP:5
SLIDINGWINDOW:4:15 MINLEN:70
```

Repeat this for the other 3 files accordingly

```
java -jar ./Trimmomatic-0.39/trimmomatic-0.39.jar SE  SRR17655903.fastq.gz SRR17655903.trim.fastq.gz
ILLUMINACLIP:./Trimmomatic-0.39/adapters/TruSeq3-SE.fa:2:30:10 HEADCROP:5
SLIDINGWINDOW:4:15 MINLEN:70

java -jar ./Trimmomatic-0.39/trimmomatic-0.39.jar SE  SRR17655906.fastq.gz SRR17655906.trim.fastq.gz
ILLUMINACLIP:./Trimmomatic-0.39/adapters/TruSeq3-SE.fa:2:30:10 HEADCROP:5
SLIDINGWINDOW:4:15 MINLEN:70

java -jar ./Trimmomatic-0.39/trimmomatic-0.39.jar SE  SRR17655907.fastq.gz SRR17655907.trim.fastq.gz
ILLUMINACLIP:./Trimmomatic-0.39/adapters/TruSeq3-SE.fa:2:30:10 HEADCROP:5
SLIDINGWINDOW:4:15 MINLEN:70
```

*In theory you can name your files as you like. But the next step will rely on the ending fastq.gz. So much for Unix flexiblility*

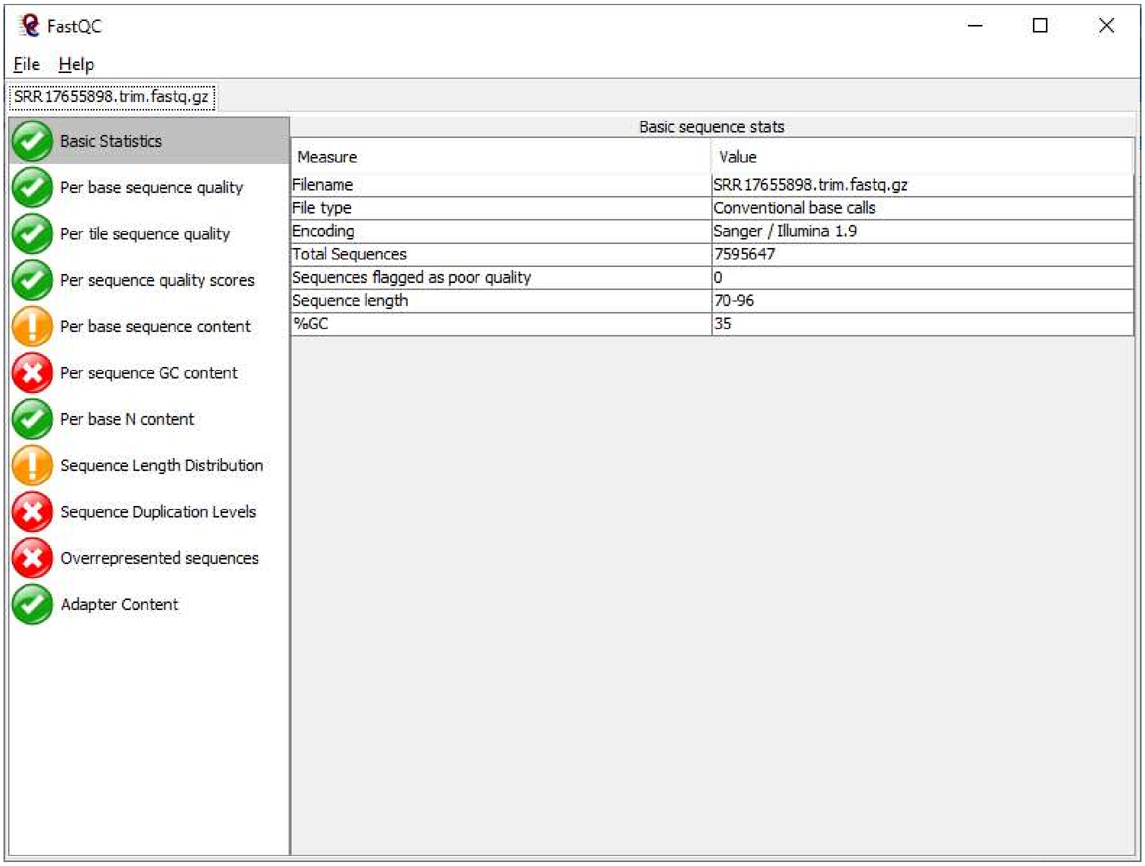
Always mind the output. It is good to keep this in a log file. Trimmomatic gives this in STDERR so we would need 2>trim.xxx.log for piping. Or note down surviving and/or dropped



```
(base) usadel@DESKTOP-N4USCVF:~/course$ java -jar ./Trimmomatic-0.39/trimmomatic-0.39.jar SE  SRR17655898.fastq.gz SRR17655898.fastq.trim.gz ILLUMINACLIP:./Trimmomatic-0.39/adapters/
TruSeq3-SE.fa:2:30:10 HEADCROP:5 SLIDINGWINDOW:4:15 MINLEN:70
TrimmomaticSE: Started with arguments:
 SRR17655898.fastq.gz SRR17655898.fastq.trim.gz ILLUMINACLIP:./Trimmomatic-0.39/adapters/TruSeq3-SE.fa:2:30:10 HEADCROP:5 SLIDINGWINDOW:4:15 MINLEN:70
Automatically using 1 threads
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Quality encoding detected as phred33
Input Reads: 8650483 Surviving: 7595647 (87.81%) Dropped: 1054836 (12.19%)
TrimmomaticSE: Completed successfully
(base) usadel@DESKTOP-N4USCVF:~/course$
```

# CHECK : could you trim all files

We can recheck our files in FASTQC


FastQC
File
Help
SRR17655898.trim.fastq.gz
✓ Basic Statistics
✓ Per base sequence quality
✓ Per tile 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 sequence stats

| Measure | Value |
|---|---|
| Filename | SRR17655898.trim.fastq.gz |
| File type | Conventional base calls |
| Encoding | Sanger / Illumina 1.9 |
| Total Sequences | 7595647 |
| Sequences flagged as poor quality | 0 |
| Sequence length | 70-96 |
| %GC | 35 |

In this special case -public and easily accessible data- we can delete the original files to save some space.

# Analyzing data

We now map the reads against the genome for this we use hisat2 as you guessed it blat is too slow.

# HISAT2

Hence, we must build an index for searching.

In the simplest case hisat2-build genomefileinfasta INDEXNAME

It makes sense to give the Index a meaningful name and document this very well

After that we can go ahead and search

- `-q`: Input file(s) are in FASTQ format
- `-x`: Path to the HISAT2 index files for the reference genome
- `-U`: Input file(s) contain unpaired reads

- `-1` and `-2`: Input file(s) contain paired-end reads, with `-1` specifying the file with the first mate and `-2` specifying the file with the second mate
- `-S`: Path to the output SAM file
- `--rna-strandness`: Specifies the strand-specificity of the RNA-seq data. Options are `FR`, `RF`, and `FF`.
- `--dta`: for gene definition
- `--threads` or `-p`: Number of threads to use for alignment (default is 1)

```
cd course
LINUX (UBUNTU on WINOWS)
wget https://cloud.biohpc.swmed.edu/index.php/s/oTtGWbWjaxsQ2Ho/download
unzip download
wget https://solgenomics.net/ftp/genomes/Solanum_lycopersicoides/SlydLA2951_v2.0/SlydLA2951_v2.0_chromosomes.fasta

ONLY FOR MAC
curl https://cloud.biohpc.swmed.edu/index.php/s/zMgEtnF6LjnjFrr/download --output hisatosx.zip
unzip  hisatosx.zip
curl https://solgenomics.net/ftp/genomes/Solanum_lycopersicoides/SlydLA2951_v2.0/SlydLA2951_v2.0_chromosomes.fasta --output  SlydLA2951_v2.0_chromosomes.fasta

./hisat2-2.2.1/hisat2-build SlydLA2951_v2.0_chromosomes.fasta PERSI
OR
./hisat2-2.2.1/hisat2-build-l SlydLA2951_v2.0_chromosomes.fasta PERSI
```

# CHECK do you now have several new files starting with the name PERSI? And when you do ls -l you see that they have a file size >0

Now we run Hisat2 in unpaired (single ended) mode as we only have single ended reads

```
./hisat2-2.2.1/hisat2 -x PERSI -U SRR17655898.trim.fastq.gz -p 4 -S SRR17655898.sam
./hisat2-2.2.1/hisat2 -x PERSI -U SRR17655903.trim.fastq.gz -p 4 -S SRR17655903.sam
./hisat2-2.2.1/hisat2 -x PERSI -U SRR17655906.trim.fastq.gz -p 4 -S SRR17655906.sam
./hisat2-2.2.1/hisat2 -x PERSI -U SRR17655907.trim.fastq.gz -p 4 -S SRR17655907.sam
```

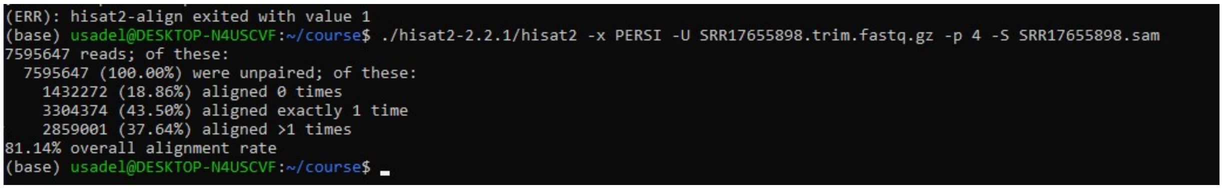
-x gives the reference we had just built

-U is for unpaired reads

-p is for multiple threads (if you have that)

-S is for sam output

Once again we get very valuable information that we should keep by using 2> or by noting it down



```
(ERR): hisat2-align exited with value 1
(base) usadel@DESKTOP-N4USCVF:~/course$ ./hisat2-2.2.1/hisat2 -x PERSI -U SRR17655898.trim.fastq.gz -p 4 -S SRR17655898.sam
7595647 reads; of these:
  7595647 (100.00%) were unpaired; of these:
    1432272 (18.86%) aligned 0 times
    3304374 (43.50%) aligned exactly 1 time
    2859001 (37.64%) aligned >1 times
81.14% overall alignment rate
(base) usadel@DESKTOP-N4USCVF:~/course$ 
```

# CHECK do you have 4 large SAM files now?

# Samtools: Getting the pocket knife

*Ubuntu: Simple solution but it will install an old version if it works*

```
sudo apt install samtools
#check if if works
samtools
```

*if it works in the following just write samtools instead of ./samtools-1.9/samtools*

## MacOS and if the above doesn’t work on ubuntu

**Ubuntu only if the above solution doesn’t work MAC Skip to next red section**

```
sudo apt-get update
sudo apt-get install gcc
sudo apt-get install make
sudo apt-get install libbz2-dev
sudo apt-get install zlib1g-dev
sudo apt-get install libncurses5-dev
sudo apt-get install libncursesw5-dev
sudo apt-get install liblzma-dev
sudo apt-get install libncurses5-dev libncursesw5-dev
```

```
cd
cd course
 wget https://github.com/samtools/htslib/releases/download/1.17/htslib-1.17.tar.bz2
tar -vxjf htslib-1.17.tar.bz2
cd htslib-1.17
configure –disable-lzma
make
```

```
cd ..
wget https://github.com/samtools/samtools/releases/download/1.17/samtools-1.17.tar.bz2
tar -vxjf samtools-1.17.tar.bz2
cd samtools-1.17
configure –disable-lzma
make
```

**Mac**

Install Xcode
You can of course also download the bz2 files (bz2 is a compression) in your browser and add it to the course folder

```
cd
cd course
 curl https://github.com/samtools/htslib/releases/download/1.17/htslib-1.17.tar.bz2 --output htslib-1.17.tar.bz2
tar -vxjf htslib-1.17.tar.bz2
cd htslib-1.17
configure –disable-lzma
make
```

```
cd ..
wget https://github.com/samtools/samtools/releases/download/1.17/samtools-1.17.tar.bz2 --output sammtools-1.17.tar.bz2
tar -vxjf samtools-1.17.tar.bz2
cd samtools-1.17
configure –disable-lzma
make
```

Make will report warnings but can’t report an error

# CHECK that samtools is working try

# ./samtools-1.17/samtools

We just sort convert to bam and then index the files

> *If you are short in hard disk space convert a file then delete the sam file. Tpyically one would redirect hisat’s output directly to bam file using “|”*

If you used apt install samtools above don’t write

./samtools-1.17/samtools

But just

samtools

```
 ./samtools-1.17/samtools sort SRR17655898.sam -o SRR17655898.bam
./samtools-1.17/samtools index SRR17655898.bam

./samtools-1.17/samtools sort SRR17655906.sam -o SRR17655906.bam
./samtools-1.17/samtools index SRR17655906.bam

./samtools-1.17/samtools sort SRR17655907.sam -o SRR17655907.bam
./samtools-1.17/samtools index SRR17655907.bam

./samtools-1.17/samtools sort SRR17655903.sam -o SRR17655903.bam
./samtools-1.17/samtools index SRR17655903.bam

#let’s have a look how file sizes change of all that ends in “am”
ls -alh *am
#not bad so we can delete all SAM files now
#!!!!!! Careful only do this is if you are sure you have all bam files
rm *sam
```

## CHECK that you have 4 bam file now and 4 bai files (these are indeces on the bam file for faster navigation)

## IGV Displaying and analyzing data

Get IGV for Mac or Windows or Linux https://software.broadinstitute.org/software/igv/download

1. Load data: To load the genome data into IGV, click "Genomes" in the top menu and select "Load Genome from File" or "Load from URL". This will open a dialog box

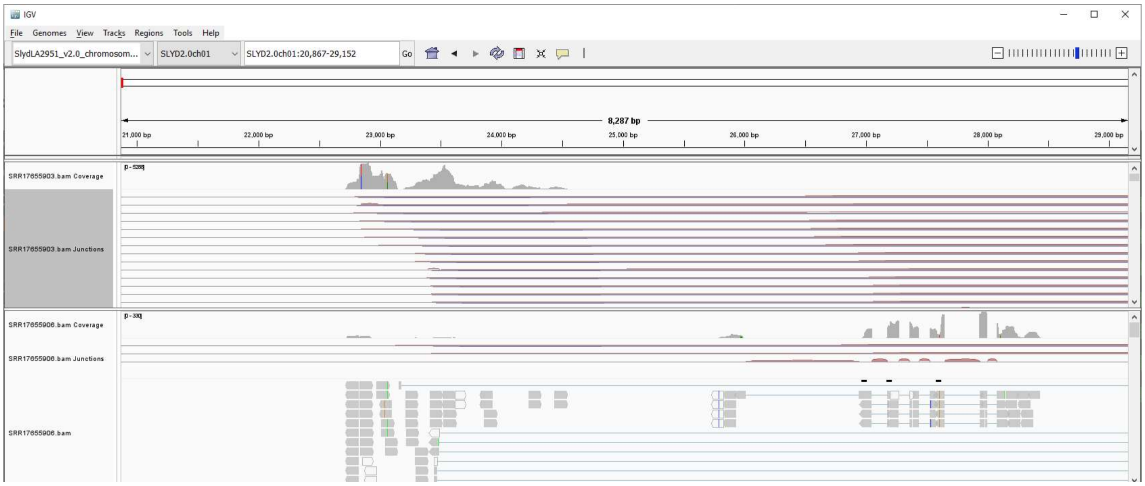
where you can select your data file. You might have to rename the genome file to XXX.fasta (where XXX is any name you like : no spaces no special characters)

2. Next load your indexed bam files Click File=>load from file Load at least …03.bam and ..06.bam
3. Navigate the genome: Once your data is loaded, you can navigate the genome using the graphical interface. You can zoom in and out by scrolling with your mouse or trackpad, or by using the zoom slider in the top left corner of the window. You can also use the "Go To" box to navigate to a specific genomic location.
4. View data: To view your data, click on the "Tracks" menu and select the tracks you want to view. Aligned reads, annotations, and other data types can be added to the view. You can adjust the display settings of the tracks, such as color and height, by right-clicking on the track and selecting "Configure".

Move around in the data check location

SLYD2.0ch01:20,867-29,152

(you can paste this into the box next to GO)



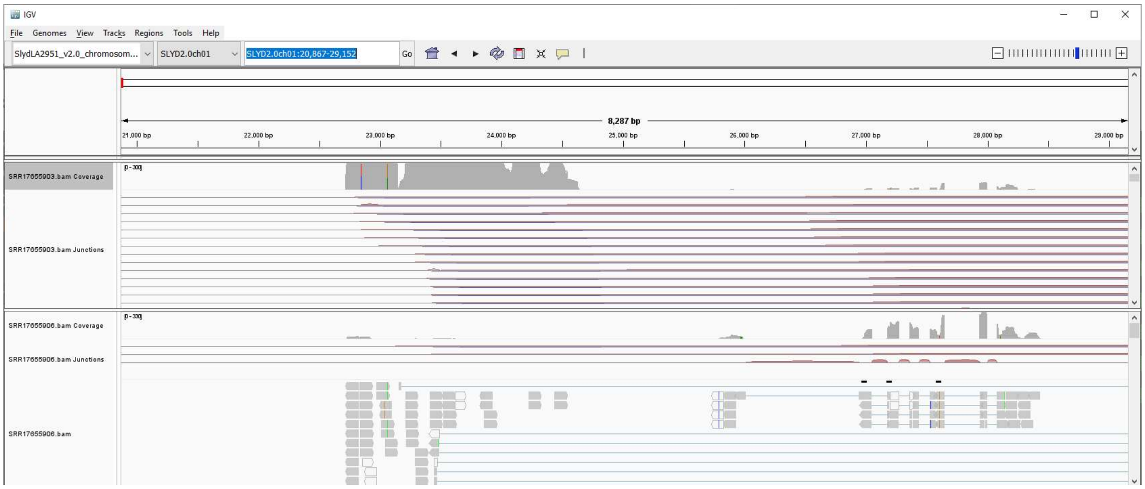
SlydLA2951_v2.0_chromosom...

SLYD2.0ch01

SLYD2.0ch01:20,867-29,152

What is happening here?

Look very carefully!



But alas we are missing something. Where are the genes?

We can load them using File Load from file using the GFF file

## Genome annotation

You can also skip directly to using kallisto now as an alternative way to pseudomap reads to transcripts directly

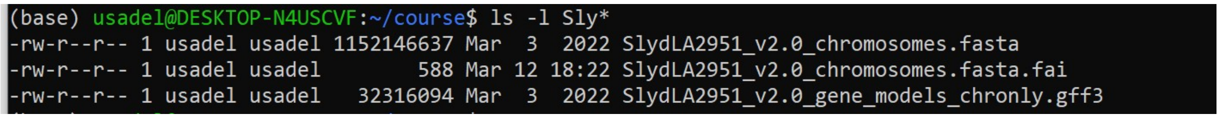
Now we know (or have an idea) where all the reads were coming from in the genome. This is stored in the sam (bam/cram) file an we can go ahead in using this. However, in the end we wanted gene expression data.

We can get everything from here

https://solgenomics.net/ftp/genomes/Solanum_lycopersicoides/SlydLA2951_v2.0/

```
wget https://solgenomics.net/ftp/genomes/Solanum_lycopersicoides/SlydLA2951_v2.0/SlydLA2951_v2.0_gene_models_all.gff3
wget https://solgenomics.net/ftp/genomes/Solanum_lycopersicoides/SlydLA2951_v2.0/SlydLA2951_v2.0_chromosomes.fasta


#mac
curl -l https://solgenomics.net/ftp/genomes/Solanum_lycopersicoides/SlydLA2951_v2.0/SlydLA2951_v2.0_gene_models_all.gff3 --output SlydLA2951_v2.0_gene_models_all.gff3
curl https://solgenomics.net/ftp/genomes/Solanum_lycopersicoides/SlydLA2951_v2.0/SlydLA2951_v2.0_chromosomes.fasta --output SlydLA2951_v2.0_chromosomes.fasta
```

**CHECK do you have the gene model and the chrosome.fasta files now and the contain data (ls -l head etc)**



```
(base) usadel@DESKTOP-N4USCVF:~/course$ ls -l Sly*
-rw-r--r-- 1 usadel usadel 1152146637 Mar  3  2022 SlydLA2951_v2.0_chromosomes.fasta
-rw-r--r-- 1 usadel usadel        588 Mar 12 18:22 SlydLA2951_v2.0_chromosomes.fasta.fai
-rw-r--r-- 1 usadel usadel   32316094 Mar  3  2022 SlydLA2951_v2.0_gene_models_chronly.gff3
```

# Annotated genomes and data to expression

## featurecounts

Now we want to annotate data using featurecounts

Featurecounts is written in C and very fast

```
featureCounts -a SlydLA2951_v2.0_gene_models_chronly.gff3 -t 'gene' -g 'ID' -o counts *.bam
```

Inputs are -a the gtf (gff) file describing where the genes are -t what we want to count in the gff file and -g tells us how this is named in this file -o gives the output and finally we add the bam/sam files.

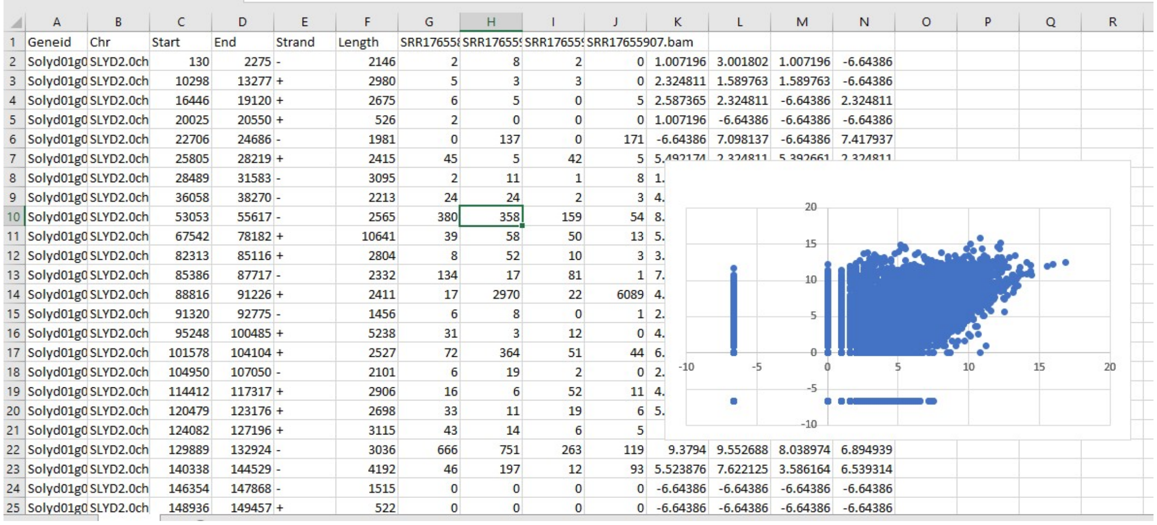
*Featurecounts MIGHT only work on older MACs.*

```
wget https://sourceforge.net/projects/subread/files/subread-2.0.3/subread-2.0.3-Linux-x86_64.tar.gz
tar -xvzf subread-2.0.3-Linux-x86_64.tar.gz

#macOS
curl https://sourceforge.net/projects/subread/files/subread-2.0.3/subread-2.0.3-macOS-x86_64.tar.gz --output subread-2.0.3-macOS-x86_64.tar.gz

tar -xvzf  subread-2.0.3-macOS-x86_64.tar.gz

#now we have subread for feastuecounts let's use it
#UBUNTU
./subread-2.0.3-Linux-x86_64/bin/featureCounts -a SlydLA2951_v2.0_gene_models_chronly.gff3 -t 'gene' -g 'ID' -o counts *.bam
#MAC
./ subread-2.0.3-macOS-x86_64/bin/featureCounts -a SlydLA2951_v2.0_gene_models_chronly.gff3 -t 'gene' -g 'ID' -o counts *.bam
```

# CHECK you have a counts file now

## Let's inspect our data

We have our first result which we can inspect now



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Geneid | Chr | Start | End | Strand | Length | SRR17655 | SRR17655 | SRR17655 | SRR17655907.bam | | | | |
| 2 | Solyd01g0 | SLYD2.0ch | 130 | 2275 | - | 2146 | 2 | 8 | 2 | 0 | 1.007196 | 3.001802 | 1.007196 | -6.64386 |
| 3 | Solyd01g0 | SLYD2.0ch | 10298 | 13277 | + | 2980 | 5 | 3 | 3 | 0 | 2.324811 | 1.589763 | 1.589763 | -6.64386 |
| 4 | Solyd01g0 | SLYD2.0ch | 16446 | 19120 | + | 2675 | 6 | 5 | 0 | 5 | 2.587365 | 2.324811 | -6.64386 | 2.324811 |
| 5 | Solyd01g0 | SLYD2.0ch | 20025 | 20550 | + | 526 | 2 | 0 | 0 | 0 | 1.007196 | -6.64386 | -6.64386 | -6.64386 |
| 6 | Solyd01g0 | SLYD2.0ch | 22706 | 24686 | - | 1981 | 0 | 137 | 0 | 171 | -6.64386 | 7.098137 | -6.64386 | 7.417937 |
| 7 | Solyd01g0 | SLYD2.0ch | 25805 | 28219 | + | 2415 | 45 | 5 | 42 | 5 | 5.492174 | 2.324811 | 5.392661 | 2.324811 |
| 8 | Solyd01g0 | SLYD2.0ch | 28489 | 31583 | - | 3095 | 2 | 11 | 1 | 8 | 1. | | | |
| 9 | Solyd01g0 | SLYD2.0ch | 36058 | 38270 | - | 2213 | 24 | 24 | 2 | 3 | 4. | | | |
| 10 | Solyd01g0 | SLYD2.0ch | 53053 | 55617 | - | 2565 | 380 | 358 | 159 | 54 | 8. | | | |
| 11 | Solyd01g0 | SLYD2.0ch | 67542 | 78182 | + | 10641 | 39 | 58 | 50 | 13 | 5. | | | |
| 12 | Solyd01g0 | SLYD2.0ch | 82313 | 85116 | + | 2804 | 8 | 52 | 10 | 3 | 3. | | | |
| 13 | Solyd01g0 | SLYD2.0ch | 85386 | 87717 | - | 2332 | 134 | 17 | 81 | 1 | 7. | | | |
| 14 | Solyd01g0 | SLYD2.0ch | 88816 | 91226 | + | 2411 | 17 | 2970 | 22 | 6089 | 4. | | | |
| 15 | Solyd01g0 | SLYD2.0ch | 91320 | 92775 | - | 1456 | 6 | 8 | 0 | 1 | 2. | | | |
| 16 | Solyd01g0 | SLYD2.0ch | 95248 | 100485 | + | 5238 | 31 | 3 | 12 | 0 | 4. | | | |
| 17 | Solyd01g0 | SLYD2.0ch | 101578 | 104104 | + | 2527 | 72 | 364 | 51 | 44 | 6. | | | |
| 18 | Solyd01g0 | SLYD2.0ch | 104950 | 107050 | - | 2101 | 6 | 19 | 2 | 0 | 2. | | | |
| 19 | Solyd01g0 | SLYD2.0ch | 114412 | 117317 | + | 2906 | 16 | 6 | 52 | 11 | 4. | | | |
| 20 | Solyd01g0 | SLYD2.0ch | 120479 | 123176 | + | 2698 | 33 | 11 | 19 | 6 | 5. | | | |
| 21 | Solyd01g0 | SLYD2.0ch | 124082 | 127196 | + | 3115 | 43 | 14 | 6 | 5 | | | | |
| 22 | Solyd01g0 | SLYD2.0ch | 129889 | 132924 | - | 3036 | 666 | 751 | 263 | 119 | 9.3794 | 9.552688 | 8.038974 | 6.894939 |
| 23 | Solyd01g0 | SLYD2.0ch | 140338 | 144529 | - | 4192 | 46 | 197 | 12 | 93 | 5.523876 | 7.622125 | 3.586164 | 6.539314 |
| 24 | Solyd01g0 | SLYD2.0ch | 146354 | 147868 | - | 1515 | 0 | 0 | 0 | 0 | -6.64386 | -6.64386 | -6.64386 | -6.64386 |
| 25 | Solyd01g0 | SLYD2.0ch | 148936 | 149457 | + | 522 | 0 | 0 | 0 | 0 | -6.64386 | -6.64386 | -6.64386 | -6.64386 |

You can open it in a spreadsheet or phyton and try to plot the 4 files against each other….

As the spread is to large apply the log first (to avoid NA numbers you  can add a small offset like 0.01 or 0.1 counts to each read here for visualiuation)

What do you observe?

*Stop and think!*

*Is  this enough to analyze our data ? Is the right way*

## Using  kallisto

We will be using kallisto which hopefully should  also work on latest Mac versions

The principle is very similar to salmon (almost identical indeed)

For Mac you might need to change your privacy to allow installation

```
cd
cd course
wget https://github.com/pachterlab/kallisto/releases/download/v0.48.0/kallisto_linux-v0.48.0.tar.gz
tar -xzvf kallisto_linux-v0.48.0.tar.gz

#mac
curl https://github.com/pachterlab/kallisto/releases/download/v0.48.0/kallisto_mac-v0.48.0.tar.gz --output kallisto_mac-v0.48.0.tar.gz
tar -xzvf kallisto_mac-v0.48.0.tar.gz
```

We need to build an index on our genome first

ILIAS and github course website contain all transcripts lyco.fa in compressed or uncompressed format download and uncompress them

```
./kallisto/kallisto index -i LYCOINDEX lyco.fa
```

Now that we have built an index (against the transcriptome) we can pseudomap read files against it. However unlike salmon we need to tell kallisto the insert size of the sequenced reads first with -l as well as the standard deviation -s . For single ended reads we can only infer this if the fragments were actually size analyzed and we got the data. (Salmon needs this as well but sets it to a similar value like below). In both cases for paired end reads this can be automatically be inferred.

```
./kallisto/kallisto quant -i LYCOINDEX -o k_SRR17655898  --single -l 180 -s 20 SRR17655898.trim.fastq.gz
./kallisto/kallisto quant -i LYCOINDEX -o k_SRR17655903 --single -l 180 -s 20 SRR17655903.trim.fastq.gz
./kallisto/kallisto quant -i LYCOINDEX -o k_SRR17655906 --single -l 180 -s 20 SRR17655906.trim.fastq.gz
./kallisto/kallisto quant -i LYCOINDEX -o k_SRR17655907 --single -l 180 -s 20 SRR17655907.trim.fastq.gz
```