Bacterial WGS training : Exercise 4

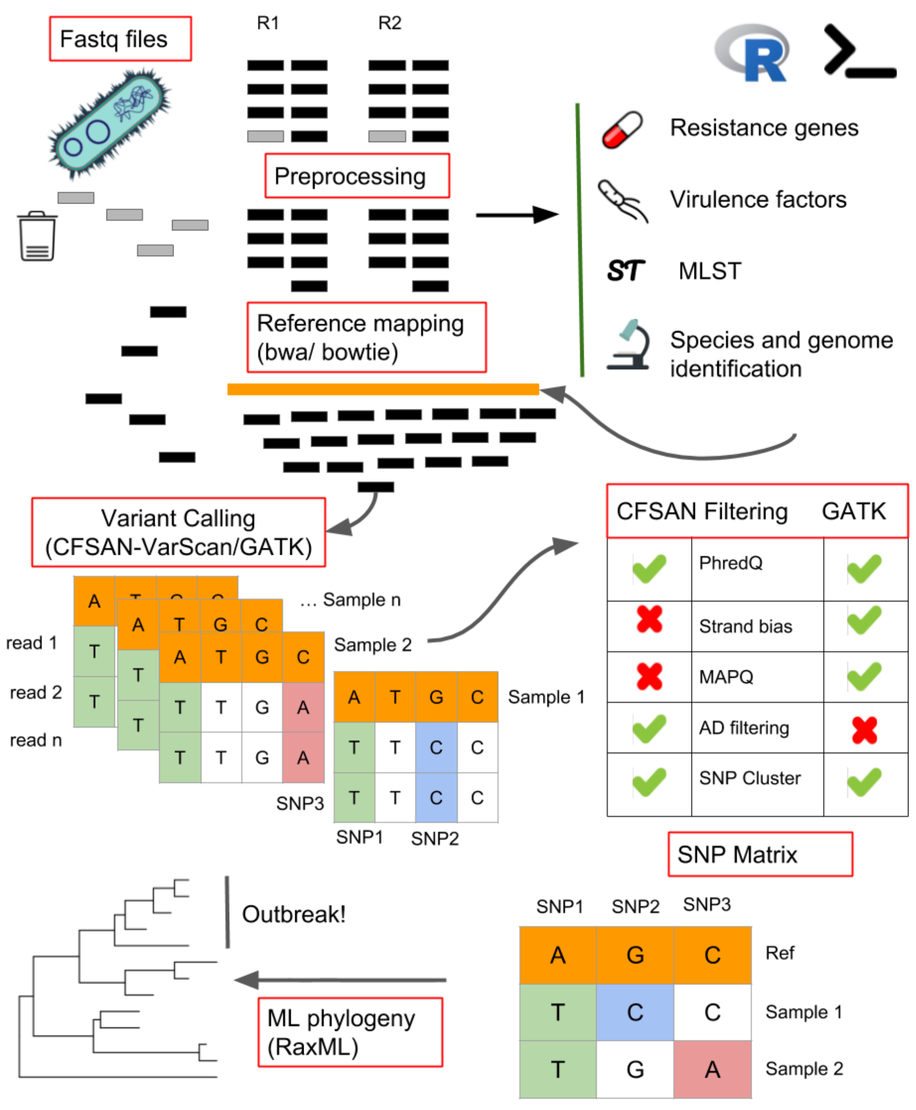
Title	**SNP-based bacterial outbreak investigation.**
Training dataset:	
Questions:	• Do I have the needed depth of coverage? • Have I chosen the correct reference? • How do I create a SNP matrix? How many SNPs do I have? • How can I visualize my phylogenetic tree? Which problems can I encounter? • Which strains belong to the outbreak?
Objectives:	• Trimming and quality control of raw reads. • Mapping against genome reference and duplicate filter. • Variant Calling. • SNP matrix creation. • Maximum Likelihood phylogeny. • Visualization of results.
Time estimation:	1 h
Key points:	• Importance of reference selecion in SNP-based tipification. • Variant calling and SNP reconstruction is a key step in the process. • Interpretation of results is case, species and epidemiology dependant.

Introduction

Although scientific community efforts have been focused on assembly-based methods and the optimization of reconstructing complete genomes, variant calling is a essential procedure that allows per base comparison between different genomes (Olson et al 2015).

SNP-based strain typing using WGS can be performed via reference-based mapping of either reads or assembled contigs. There are many available microbial SNP pipelines, such as Snippy, NASP, SNVphyl, CFSAN SNP Pipeline, or Lyve-SET.

Variant calling is a process with a bunch of potential error sources that may lead to incorrect variant calls. Identifying and resolving this incorrect calls is critical for bacterial genomics to advance. In this exercise we will use WGS-Outbreaker a SNP-based tool developed by BU-ISCIII that uses bwa mapper, GATK variant caller and several SNP-filtering steps for SNP matrix contruction following maximun likelihood phylogeny using RAxML. Next image resumes the steps we are going to execute:



Preprocessing

We have already done our data prerpocesing in the previous exercise. If you remember, we executed a nextflow order which trimmed our raw reads and reutned a quality report for both pre- and post-trimming files. We used FastQC for checking the data quality, Trimmomatic for the trimming and MultiQC for building the statitstics report.

For mapping our reads, we will need to preprocess our data in the same way as we did for the assembly. As the results will be exactly the same we reviewed in that exercise, we will not spend more time and will move to exiting new topics.

Mapping

In the previous lecture we covered how to assamble the reads in the fastq file to recreate the original genome, or at least contigs of it. This technique requieres high sequence coverage, high read lenght reads and good read quality, plus being highly computationally demanding. This means that it is an expensive and slow method, plus having one big dissavantage when trying to compare assambled genomes: different algorithms (and even different versions of the same software) may produce different assambles from the same input.

For this reason, when the objective is to compare genomes of different samples, we use another method for rebuilding the genome called mapping. This technique consists in using a previously assembled genome as reference against which sequenced reads will be independently aligned against. Every read will be placed in the most likely position, ignoring any synergies between reads. This produces genomes with the same structure and coordinates that can be easily compared.

There are multiple mapping algorithms and softwares, but for this exercise we will use only bwa (H. Li and R. Durbin, 2010). bwa is implements a backward search with Burrows-Wheeler Transform to efficiently align short sequencing reads against a large reference sequence such as the human genome, allowing mismatches and gaps. For longer reads, it combines its algorthm with a modified Smith-Waterman's alignment, achieving the same results as the starndard algorithm but thousands of times faster. While still slower than BLAST for long query sequences, it is able to find all matches without heuristics, which makes it able to detect chimeras potentially caused by structural variations or reference misassemblies.

To map our samples with bwa, we only have to execute this command:

```
cd
cd wgs/bacterial_wgs_training_dataset/ANALYSIS
nextflow run ../../bacterial_wgs_training/main.nf \
  -profile conda \
  --reads '../RAW/DOWNSAMPLED/*_R{1,2}.fastq.gz' \
  --fasta '../REFERENCES/listeria_NC_021827.1_NoPhagues.fna' \
  --outdir 03-mapping \
  --step mapping \
  -resume
```

This command will internally execute the following programs with our samples:

1. **Preprocessing** Quality control and read trimming with FastQC and Trimmomatic, as used in the previous exercise.

    1. **Building bwa index** Bwa needs to build an index from the reference genome in order to now how to map the reads. This type of algorithms allows the software to do very fast searches on the genome. `bwa index fasta_file`
    2. **Mapping** Map each read against the reference genome using bwa mem software. `bwa mem fasta reads | samtools view -bT fasta - > bam`
    3. **Post-processing and statistics** A handful of steps have to be executed before using the bam files resulting from the mapping. First, bam files have to be sorted and indexed: `samtools sort bam -o sorted.bam samtools index sorted.bam` This step allows first to reduce file size due to the compression algorithm used to generate bam files it works better if the file is sorted, and second with the file sorted and the index generated searches in the file are way faster. Imagine to look for something in a book with the pages unordered and without an index, difficult right?

Next mapping stats generated:

```
samtools stats sorted.bam > stats.txt
```

And finally we can remove some sequencing and mapping artifacts, as the duplicated reads:

```
picard MarkDuplicates \
    INPUT=sorted.bam \
    OUTPUT=dedup.bam \
    ASSUME_SORTED=true \
    REMOVE_DUPLICATES=true \
    METRICS_FILE=picardDupMetrics.txt \
    VALIDATION_STRINGENCY=LENIENT \
```

Parameters explanation:

- ASSUME_SORTED: TRUE or FALSE. Is the bam file sorted or not?
- REMOVE_DUPLICATES: TRUE or FALSE. Do we want to remove duplicates from our bam file or do we want to just mark them?
- METRICS_FILE: where do we want to save the metrics file?
- VALIDATION_STRINGENCY = SILENT, LENIENT or STRICT. If an error comes out how the software is going to behave. SILENT the software just ignores the error and does not output anything, LENIENT it continous to run but outputs an error and STRICT the software stops when it encounters any error (this last option can be pretty annoying).

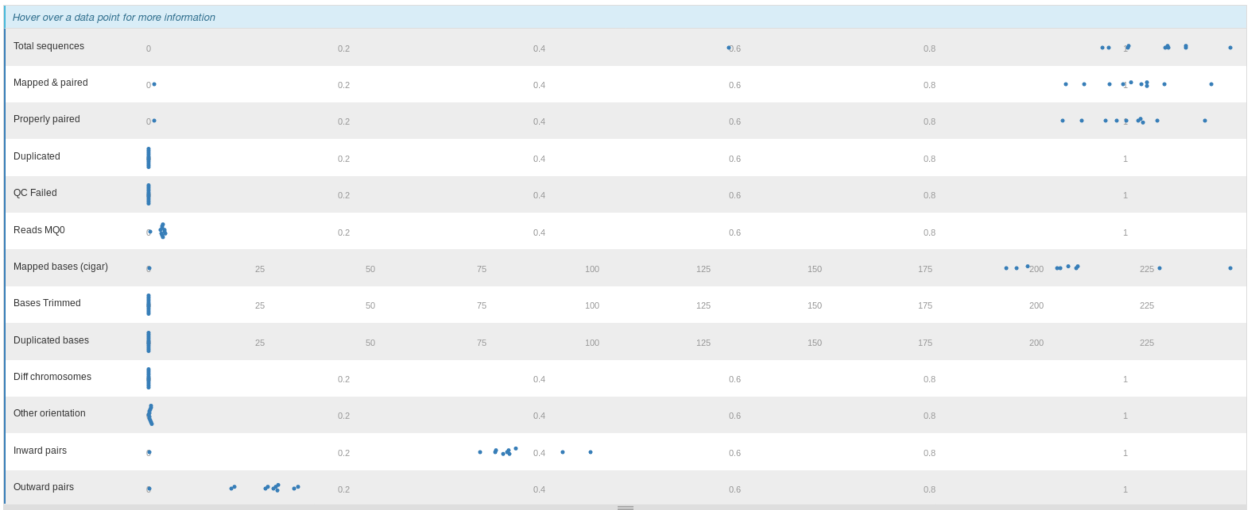
1. **MultiQC report** MultiQC will automatically search for the stats files and will compare them in user-friendly graphs `multiqc RESULTS_DIRECTORY` Finally, we have our mapped genomes. Now we can open them with IGV to see how they have mapped against the reference genome, and which variants are.

Mapping stats

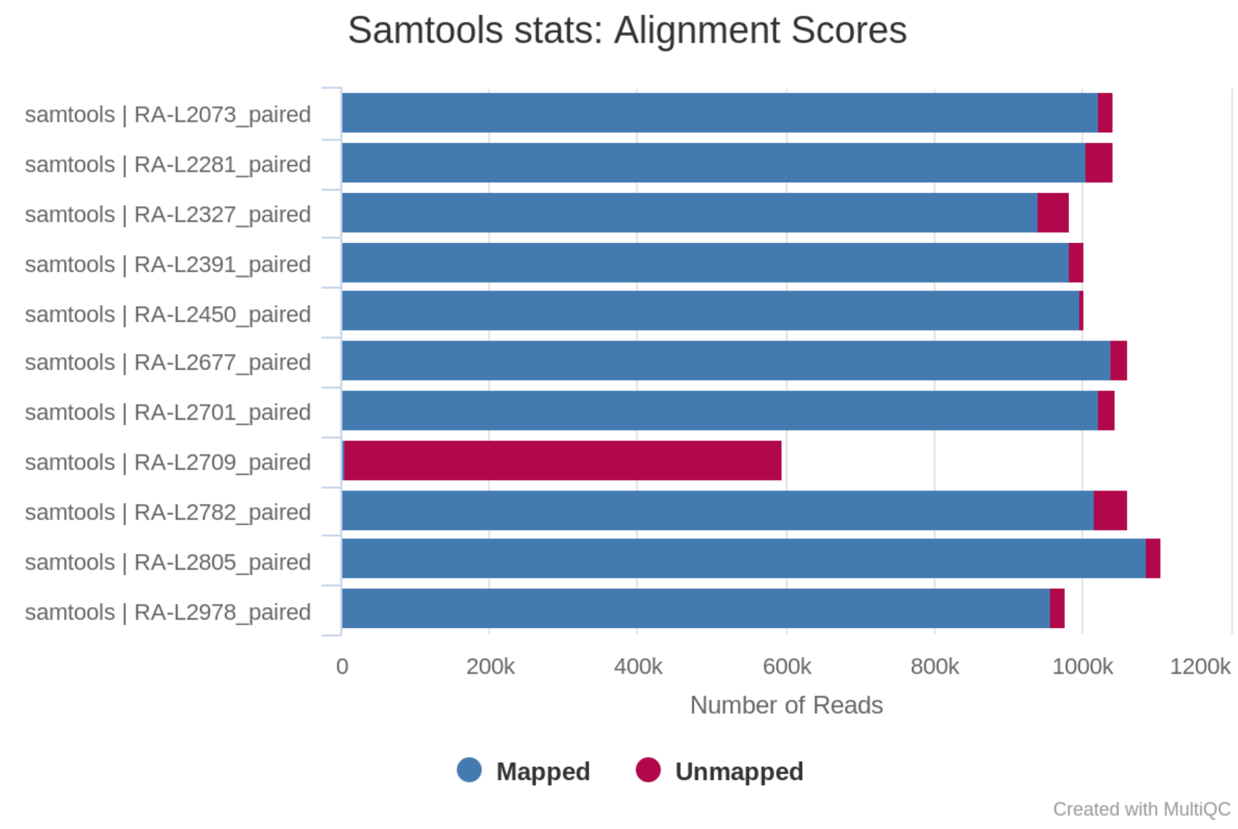
The mapping stats are saved in `results/bwa/stats/`, where you can find a stats file per mapped sample.

We are going to visualize the summary statistics of the mapping step with MultiQC:

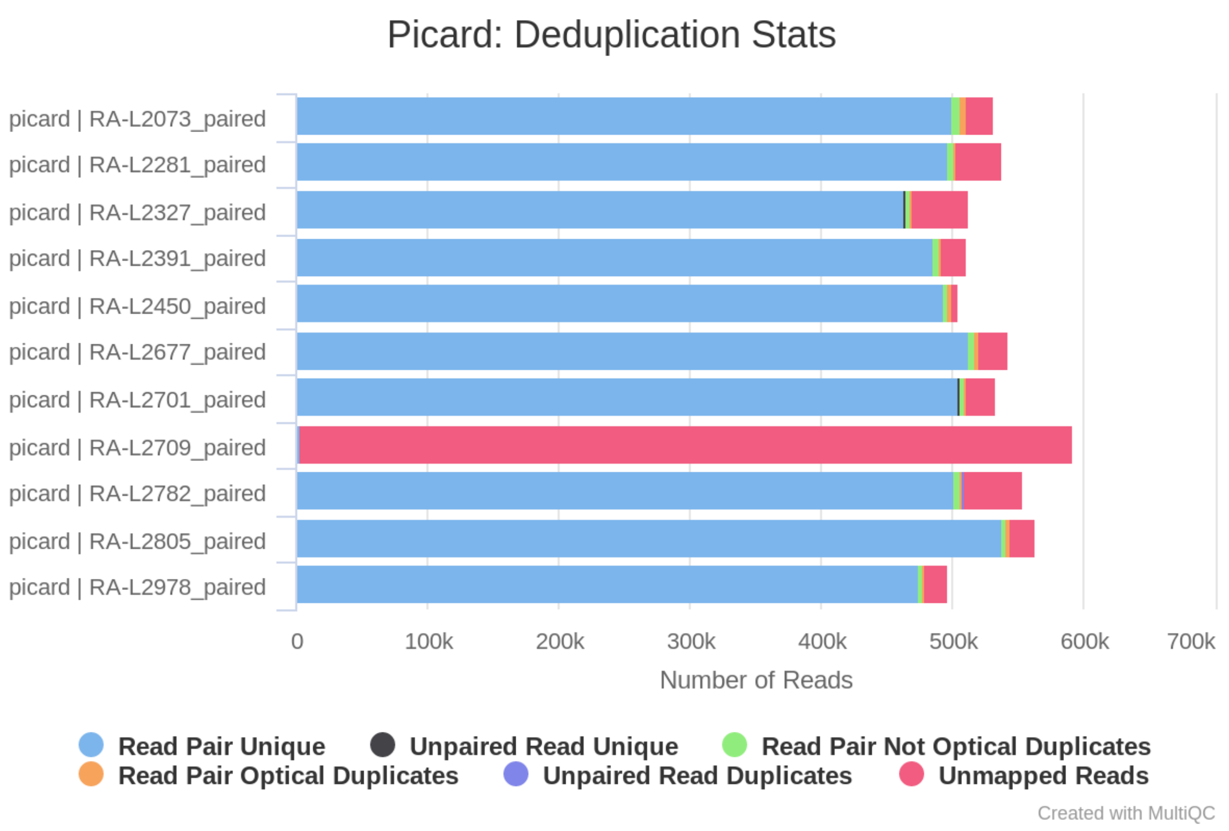
- The stats are represented in "Alignment metrics":

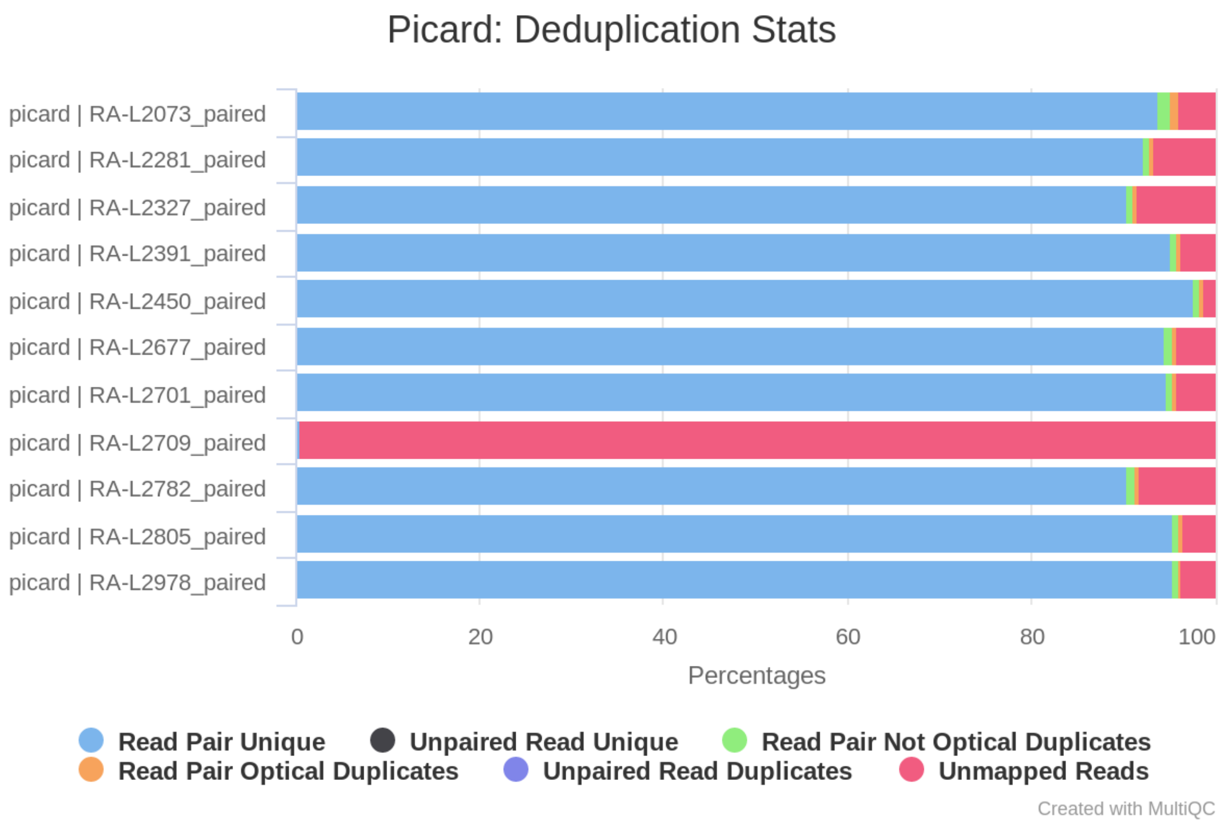


- The percentage of mapped reads per sample are plotted in "Percent Mapped":



- The number of duplicated reads per sample are plotted in "Mark Duplicates":



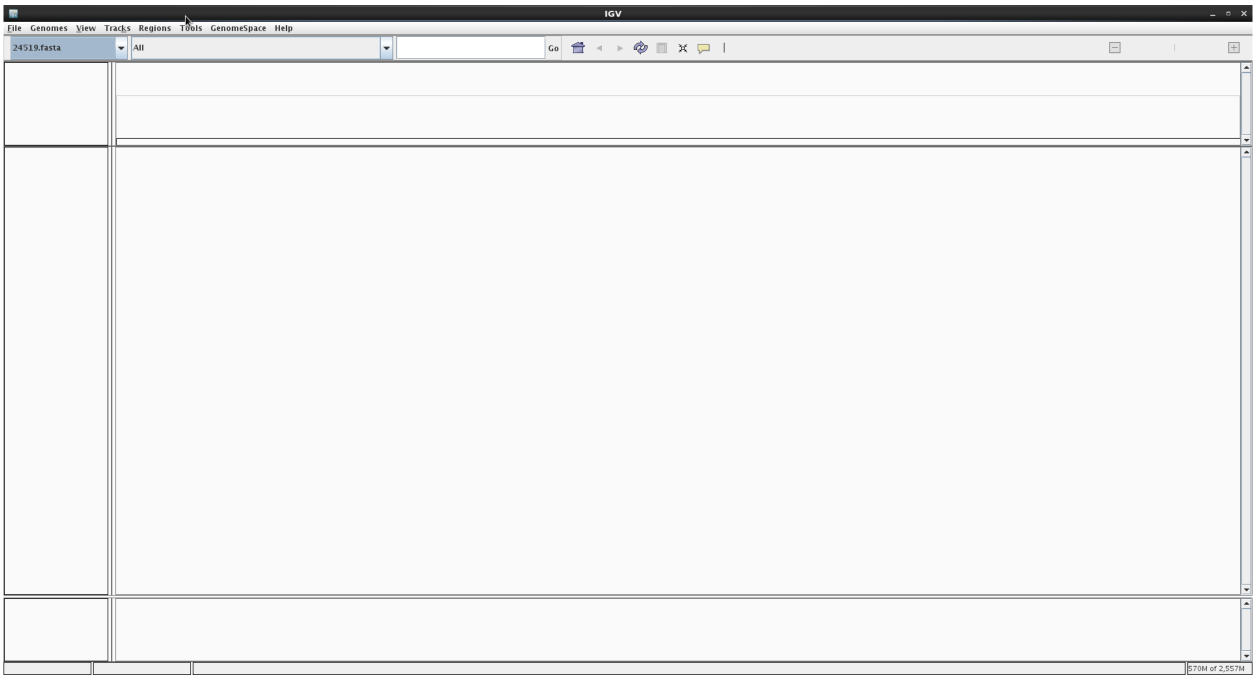


Visualizing your mapping

In order to visualize our mapping we will use IGV (Integrative Genomics Viewer). This is an open source and freely available lightweight visualization tool that enables intuitive real-time exploration of diverse, large-scale genomic data sets on standard desktop computers. It supports flexible integration of a wide range of genomic data types including aligned sequence reads, mutations, copy number, RNA interference screens, gene expression, methylation and genomic annotations.

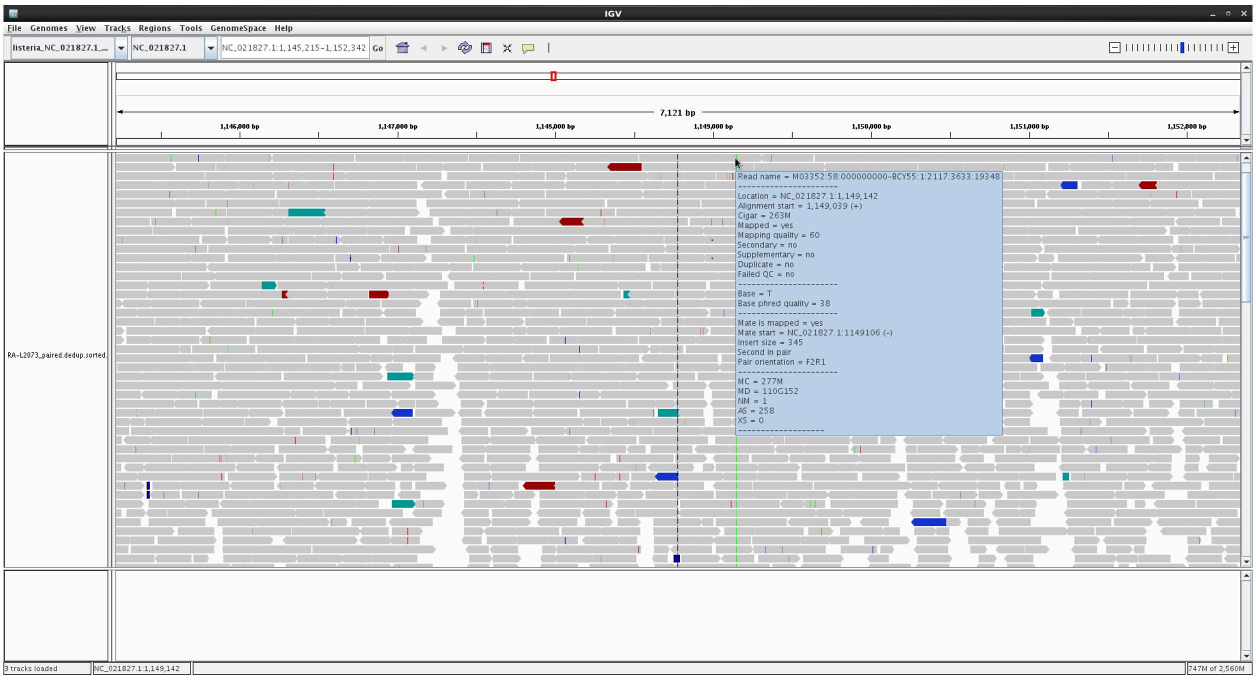
Navigation through a data set is similar to that of Google Maps, allowing the user to zoom and pan seamlessly across the genome at any level of detail from whole genome to base pair. Data sets can be loaded from local or remote sources, including cloud-based resources, enabling investigators to view their own genomic data sets alongside publicly available data.

Let's launch IGV! Navigate to your desktop and find the icon we have left for you. Double click on it and wait until it finishes loading.



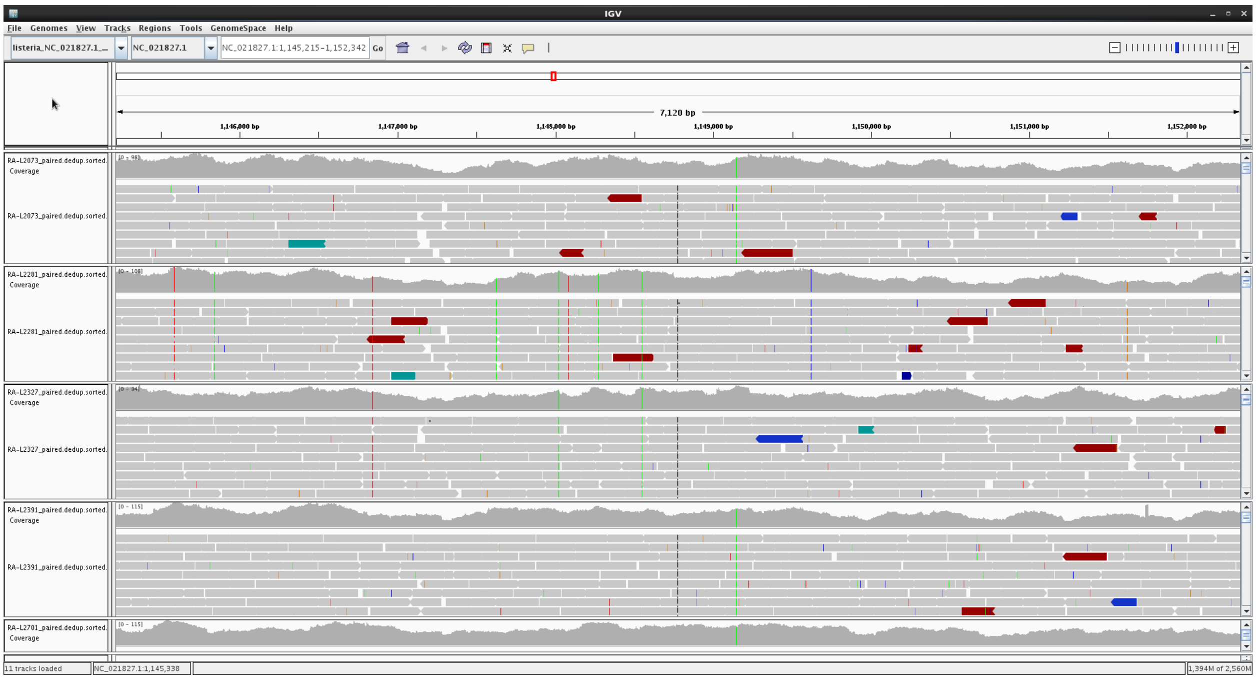
First we have to load the reference genome. Click on "Genomes" and "Load Genome from File...", navigate to our training_dataset folder and select the reference genome "listeria_NC_021827.1_NoPhagues.fna".

Now, load our mapped genomes by clicking on "File" and "load from File...", navigate to our mapped genomes (HINT: they are in the shared folder inside the "results/picard" folder), and load one of them:



```
Read name = M03352:58:000000000-BCY55:1:2117:3633:19348
----------------------
Location = NC_021827.1:1,149,142
Alignment start = 1,149,039 (+)
Cigar = 263M
Mapped = yes
Mapping quality = 60
Secondary = no
Supplementary = no
Duplicate = no
Failed QC = no
----------------------
Base = T
Base phred quality = 38
----------------------
Mate is mapped = yes
Mate start = NC_021827.1:1149106 (-)
Insert size = 345
Second in pair
Pair orientation = F2R1
----------------------
MC = 277M
MD = 110G152
NM = 1
AS = 258
XS = 0
-------------------
```

Finally, we can load as many as we want (or as many as the virtual machine survives) to compare them:


IGV
File Genomes View Tracks Regions Tools GenomeSpace Help
listeria_NC_021827.1...
NC_021827.1
NC_021827.1:1,145,215-1,152,342
Go
7,120 bp
RA-L2073_paired.dedup.sorted Coverage
RA-L2073_paired.dedup.sorted
RA-L2281_paired.dedup.sorted Coverage
RA-L2281_paired.dedup.sorted
RA-L2327_paired.dedup.sorted Coverage
RA-L2327_paired.dedup.sorted
RA-L2391_paired.dedup.sorted Coverage
RA-L2391_paired.dedup.sorted
RA-L2701_paired.dedup.sorted Coverage
11 tracks loaded
NC_021827.1:1,145,338
1,394M of 2,560M

Variant Calling

We are using snippy as the main software for variant calling, SNP-matrix creation and phylogeny performance. Following the development of the former exercises we are using nextflow, in this case using `outbreakSNP` step. This step includes the following processes:

- Preprocessing:
  - Trimming with fastp software.
  - Quality control with fastQC.
- Snippy software:
  - Mapping with bwa.
  - Variant calling with freebayes.
  - SNP-Matrix creation.
  - SNP-core-filtering
- Phylogeny with IQ-tree

Everything clear..? So let's run it.

Run the exercise

First of all we need to be clear in which folder we are. We need to be in our working directory `/home/alumno/Documents/wgs` and our training dataset downloaded the first day must be there (If you had any problem the previous sessions please refere to the setup tutorial).

You can run this command to check where you are:

```
pwd
```

Output:

```
/home/alumno/Documents/wgs/bacterial_training_dataset/ANALYSIS
```

And this one to list all the files in your working directory. Check there is the training_dataset folder and the results folder from previous sessions.

```
ls
```

Output:

```
01-handsonLinux 02-assembly work
```

Once our localization is correct we will launch nextflow with the next parameters:

- Raw reads
- step outbreakSNP
- saveTrimmed -> this parameters saves the fastq files trimmed in our results dir.
- outbreaker_config <- config file with all the parameters required by WGS-Outbreaker

```
nextflow run ../../bacterial_wgs_training/main.nf \
  -profile conda \
  --reads '../RAW/FULL_DATA/*_R{1,2}.fastq.gz' \
  --fasta ../REFERENCES/listeria_NC_021827.1_NoPhagues.fna \
  --step outbreakSNP \
  --outdir 04-outbreakSNP \
  -resume
```

> This will take a while so we need to move forward and understand what we are doing and learn how to see and interpret our results.

Results analysis

Let's proceed to analyze the results. We can find them in:

```
/home/alumno/wgs/bacterial_wgs_training_dataset/RESULTS/04-outbreakSNP
```

Since alignment and quality control results has been previously addresed in this course (see 02_QualityAndAssembly.md and Mapping Section), we will proceed to analyze variant calling results.

Variant calling results

Variants are stored in plain text files in vcf format (variant calling format). Vcf files can be found in:

```
/home/alumno/wgs/bacterial_wgs_training_dataset/RESULTS/04-outbreakSNP/Snippy/{sample_name}/snps.vcf
```

Vcf file for one sample, looks like:

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##fileDate=20210623
##source=freeBayes v1.3.2-dirty
##reference=reference/ref.fa
##contig=<ID=NC_021827.1,length=2953716>
##contig=<ID=NC_022047.1,length=55804>
##phasing=none
##commandline="freebayes -p 2 -P 0 -C 10 --min-repeat-entropy 1.5 --strict-vcf -q 13 -m 60 --min-coverage 10 -F 0.05 -f refere
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total read depth at the locus">
##INFO=<ID=RO,Number=1,Type=Integer,Description="Count of full observations of the reference haplotype.">
##INFO=<ID=AO,Number=A,Type=Integer,Description="Count of full observations of this alternate haplotype.">
##INFO=<ID=QR,Number=1,Type=Integer,Description="Reference allele quality sum in phred">
##INFO=<ID=QA,Number=A,Type=Integer,Description="Alternate allele quality sum in phred">
##INFO=<ID=AB,Number=A,Type=Float,Description="Allele balance at heterozygous sites: a number between 0 and 1 representing the
##INFO=<ID=TYPE,Number=A,Type=String,Description="The type of allele, either snp, mnp, ins, del, or complex.">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GL,Number=G,Type=Float,Description="Genotype Likelihood, log10-scaled likelihoods of the data given the called ge
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=RO,Number=1,Type=Integer,Description="Reference allele observation count">
##FORMAT=<ID=QR,Number=1,Type=Integer,Description="Sum of quality of the reference observations">
##FORMAT=<ID=AO,Number=A,Type=Integer,Description="Alternate allele observation count">
##FORMAT=<ID=QA,Number=A,Type=Integer,Description="Sum of quality of the alternate observations">
##bcftools_viewVersion=1.9+htslib-1.9
##bcftools_viewCommand=view --include 'FMT/GT="1/1" && QUAL>=100 && FMT/DP>=10 && (FMT/AO)/(FMT/DP)>=0' snps.raw.vcf; Date=Wed
##bcftools_annotateVersion=1.9+htslib-1.9
##bcftools_annotateCommand=annotate --remove ^INFO/TYPE,^INFO/DP,^INFO/RO,^INFO/AO,^INFO/AB,^FORMAT/GT,^FORMAT/DP,^FORMAT/RO,^
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  RA-L2073
NC_021827.1 5507    .   T   G   657.752 .   AB=0;AO=21;DP=21;QA=754;QR=0;RO=0;TYPE=snp  GT:DP:RO:QR:AO:QA:GL    1/1:21:0:0:21:
NC_021827.1 5659    .   A   C   848.065 .   AB=0;AO=27;DP=27;QA=993;QR=0;RO=0;TYPE=snp  GT:DP:RO:QR:AO:QA:GL    1/1:27:0:0:27:
NC_021827.1 18836   .   T   C   1946.82 .   AB=0;AO=61;DP=61;QA=2194;QR=0;RO=0;TYPE=snp GT:DP:RO:QR:AO:QA:GL    1/1:61:0:0:61:
NC_021827.1 33004   .   A   T   1940.53 .   AB=0;AO=59;DP=59;QA=2188;QR=0;RO=0;TYPE=snp GT:DP:RO:QR:AO:QA:GL    1/1:59:0:0:59:
NC_021827.1 36868   .   T   C   1415.17 .   AB=0;AO=44;DP=44;QA=1601;QR=0;RO=0;TYPE=snp GT:DP:RO:QR:AO:QA:GL    1/1:44:0:0:44:
NC_021827.1 37145   .   T   C   1951.32 .   AB=0;AO=60;DP=60;QA=2200;QR=0;RO=0;TYPE=snp GT:DP:RO:QR:AO:QA:GL    1/1:60:0:0:60:
NC_021827.1 49245   .   G   A   1613.73 .   AB=0;AO=52;DP=52;QA=1824;QR=0;RO=0;TYPE=snp GT:DP:RO:QR:AO:QA:GL    1/1:52:0:0:52:
```

Phylogeny results

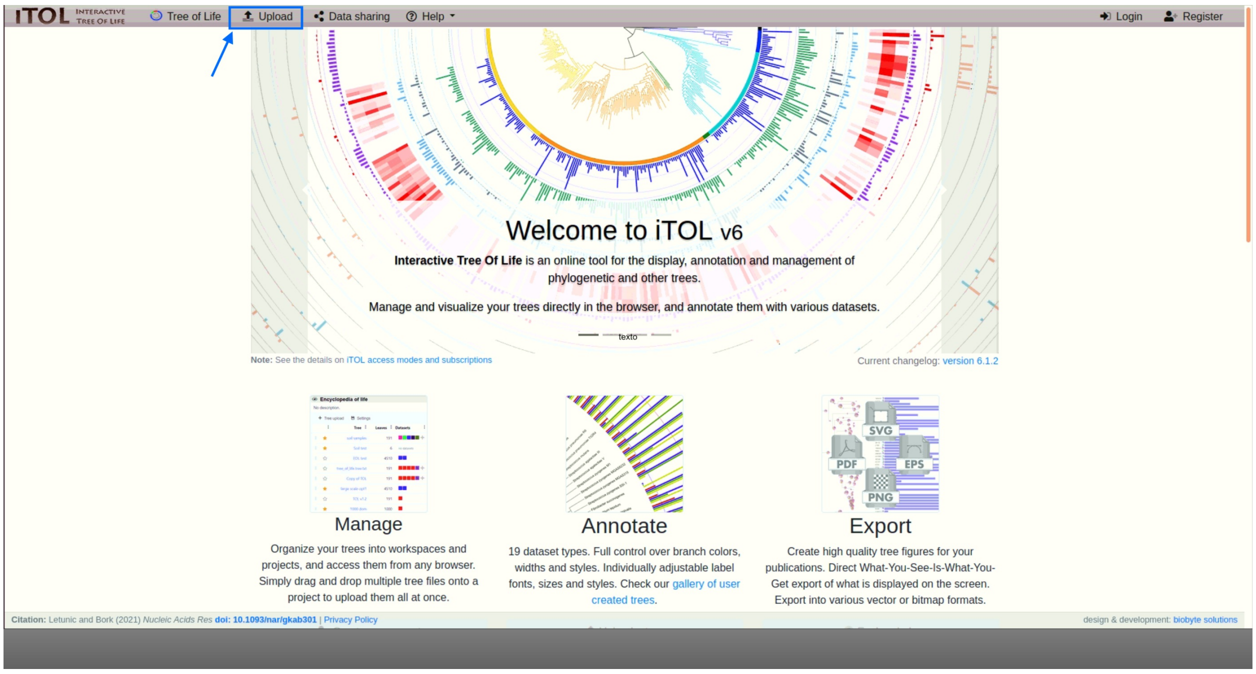
Phylogenetic tree reconstruction is performed using RAxML with 100 inferences and 100 bootstrap repetitions. RAxML results can be checked in RAxML folder:

```
/home/Alumno/Documents/wgs/bacterial_wgs_training_dataset/04-outbreakSNP/iqtree/
```

Don't worry we only need the final tree file, which is in newick format for visualization. The file is called:

```
clean.core.aln.treefile
```

Now we are going to open firefox browser and go to iTOL website. This web allows us to visualize and annotate phylogenetics trees with a very user-friendly interface. Also, it has good exporting options for publication.



Welcome to iTOL v6

Interactive Tree Of Life is an online tool for the display, annotation and management of phylogenetic and other trees.

Manage and visualize your trees directly in the browser, and annotate them with various datasets.

Note: See the details on iTOL access modes and subscriptions

Current changelog: version 6.1.2

Manage

Organize your trees into workspaces and projects, and access them from any browser. Simply drag and drop multiple tree files onto a project to upload them all at once.

Annotate

19 dataset types. Full control over branch colors, widths and styles. Individually adjustable label fonts, sizes and styles. Check our gallery of user created trees.

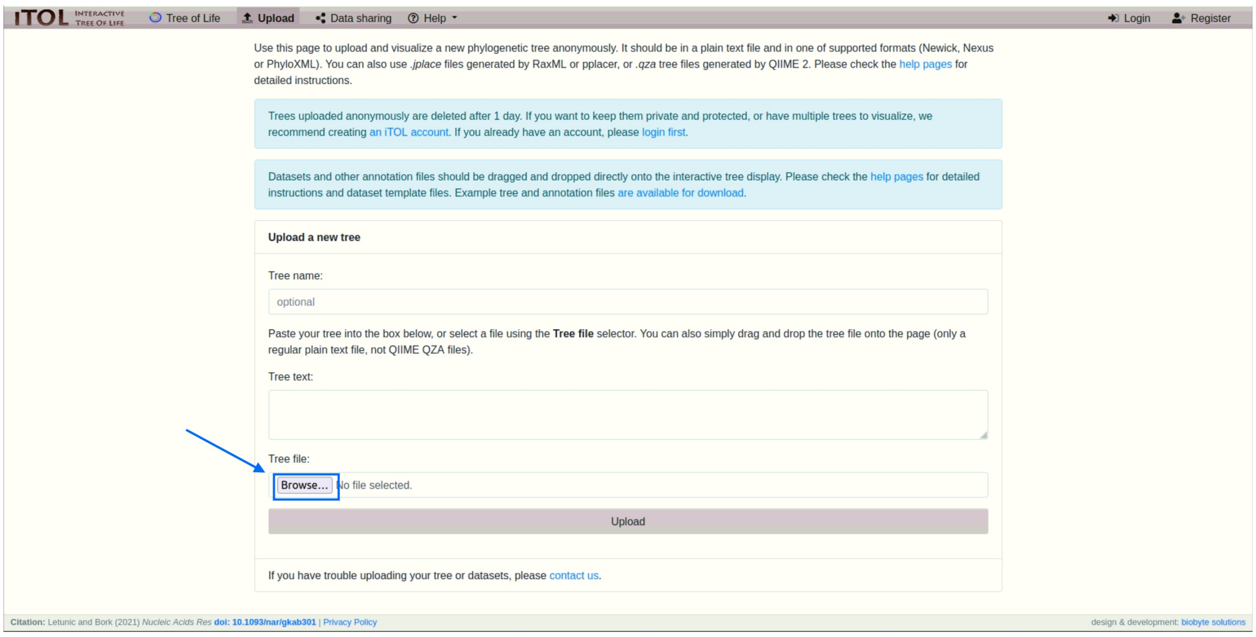
Export

Create high quality tree figures for your publications. Direct What-You-See-Is-What-You-Get export of what is displayed on the screen. Export into various vector or bitmap formats.

Citation: Letunic and Bork (2021) *Nucleic Acids Res* **doi: 10.1093/nar/gkab301** | Privacy Policy

design & development: biobyte solutions

Once in iTOL website we click in Upload in the top menu. Next we upload our tree as shown in the image:



Use this page to upload and visualize a new phylogenetic tree anonymously. It should be in a plain text file and in one of supported formats (Newick, Nexus or PhyloXML). You can also use *.jplace* files generated by RaxML or pplacer, or *.qza* tree files generated by QIIME 2. Please check the help pages for detailed instructions.

Trees uploaded anonymously are deleted after 1 day. If you want to keep them private and protected, or have multiple trees to visualize, we recommend creating an iTOL account. If you already have an account, please login first.

Datasets and other annotation files should be dragged and dropped directly onto the interactive tree display. Please check the help pages for detailed instructions and dataset template files. Example tree and annotation files are available for download.

Upload a new tree

Tree name:

optional

Paste your tree into the box below, or select a file using the **Tree file** selector. You can also simply drag and drop the tree file onto the page (only a regular plain text file, not QIIME QZA files).

Tree text:

Tree file:

Browse... No file selected.

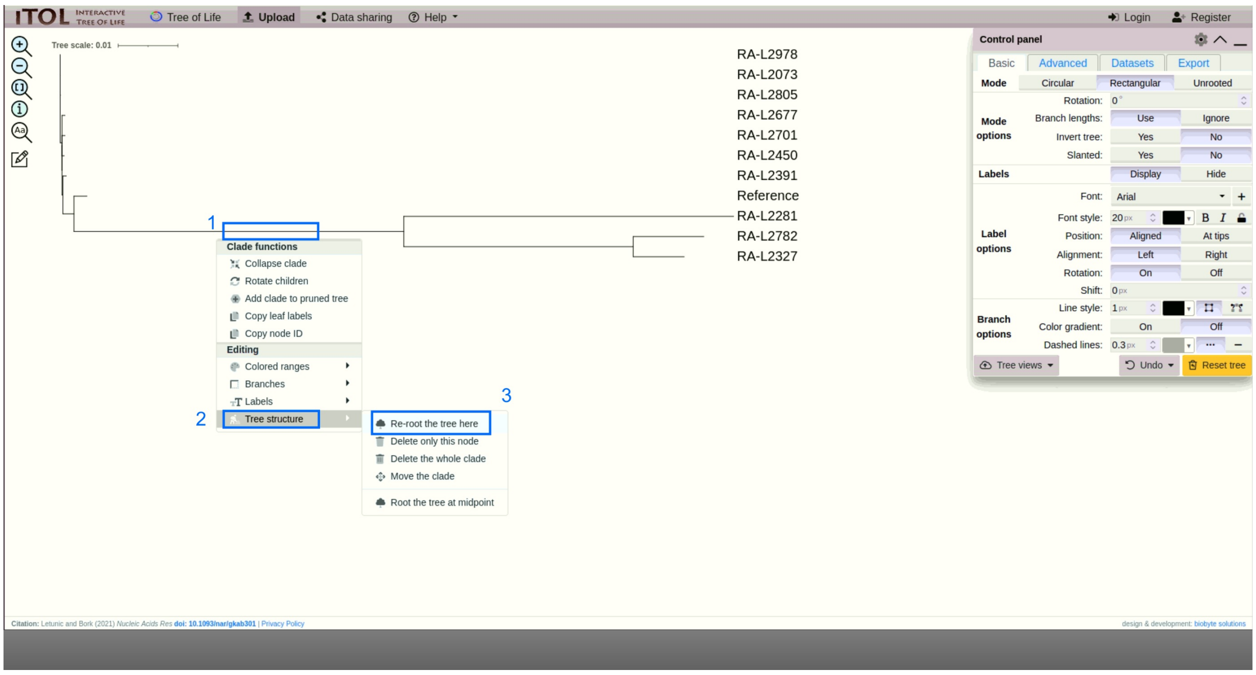
Upload

If you have trouble uploading your tree or datasets, please contact us.

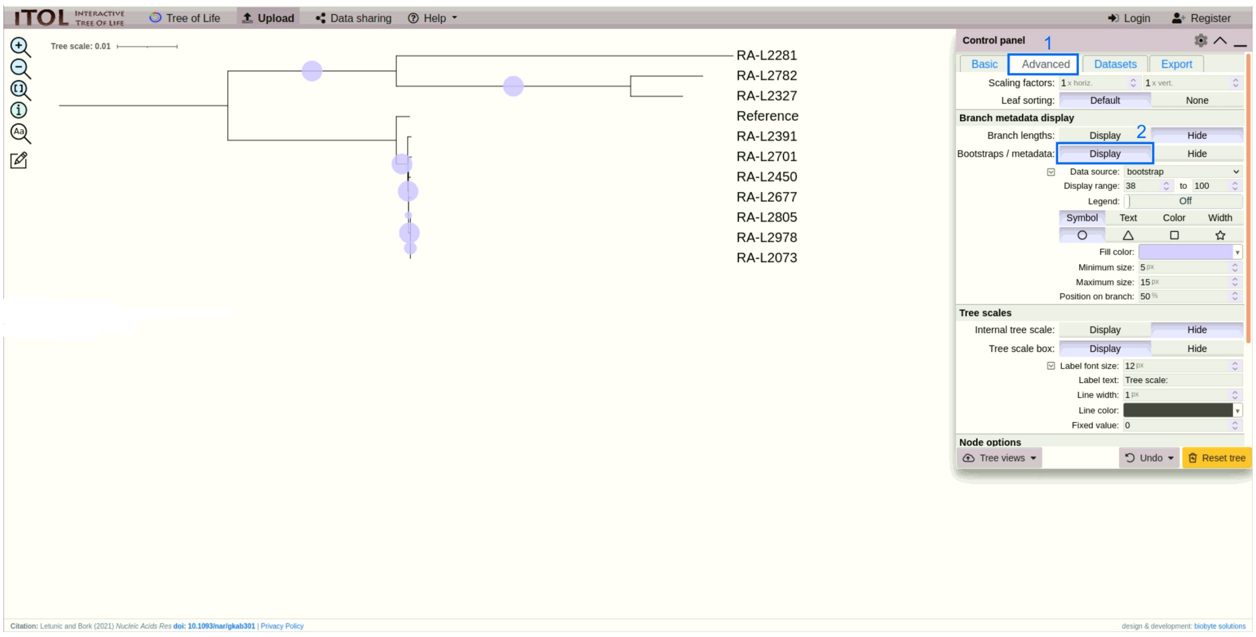
Citation: Letunic and Bork (2021) *Nucleic Acids Res* **doi: 10.1093/nar/gkab301** | Privacy Policy

design & development: biobyte solutions

Now we are visualizing our tree and we can manipulate it. First of all, as we are facing an unrooted tree with long and small branches, we are going to perform a midpoint rooting method for improve the visualization. For this we choose the longest branch an we click on it. We have to get a menu like this image:

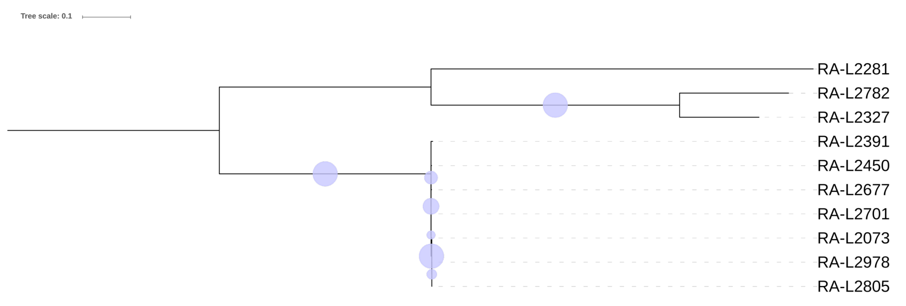


iTOL offers multiple annotation and maniputation options. We can selecto for example the display of bootstrap in the advance tab.



You can play with all the options iTOL offers reading its documentation, and export the tree in the export tab.

Now we are going to focus on our results. Our final tree should look something like this:



Which strains do you think belong to the outbreak?

Following the instructions seen in the theory class, here we can focus on topology and bootstrap.

Do we find any monophyletic group? Does it have more than 80 boostrap value?

SNPs distance

As we have studied in the theory class, maximum likelihood methods for phylogeny only offers as branch lenght the average nucleotide substitution rate, this means the branch lenght is only a estimation of the number of nucleotide changes a strain has suffer respect to another.

In order to know exactly the SNPs differences between the strains WGS-Outbreaker outputs a distance matrix showing the paired diffences among the samples, we can use MEGA software, we are not covering this on this course but it should be straight-forward.

samples	RA-L2073	RA-L2281	RA-L2327	RA-L2391	RA-L2450	RA-L2677	RA-L2701	RA-L2782	RA-L2805	RA-L2978	Reference
RA-L2073											
RA-L2281	9128,00										
RA-L2327	8718,00	8286,00									
RA-L2391	80,00	9140,00	8730,00								
RA-L2450	46,00	9122,00	8712,00	74,00							
RA-L2677	46,00	9122,00	8712,00	74,00	38,00						
RA-L2701	49,00	9127,00	8717,00	79,00	45,00	45,00					
RA-L2782	8725,00	8669,00	4159,00	8737,00	8719,00	8719,00	8724,00				
RA-L2805	4,00	9128,00	8718,00	80,00	46,00	46,00	49,00	8725,00			
RA-L2978	2,00	9126,00	8716,00	78,00	44,00	44,00	47,00	8723,00	2,00		
Reference	227,00	9109,00	8699,00	239,00	221,00	221,00	226,00	8706,00	227,00	225,00	

As we see the SNP difference cutoff is important here, and it will depend on the strain and the case. If we stablish 3-5 snps as our cutoff we can detect that the strains belonging to the outbreak are: 2978, 2805 and 2073