IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia do Barreiro

BIOInformática

High Throughput Sequencing at Your Fingertips

By: Marcelo Pereira, Daniel Marçal, Bernardo Augusto

24-06-2020

Professor Francisco Pina Martins, phD

Biological Sequences Analysis

2019/2020 Academic Year

Index

Introduction

This project is the summit of our team's semester during the 2nd part of the 2019/2020 Academic Year. For it to become possible we had to put together the knowledge acquired during a few previous tasks and 1st project.

The main objective was to solve a discussion that was being held by Botanists regarding a certain plant population that had unique phenotypic traits, which they called "GreenLeaf". This species, at first, had undistinguishable differences to another plant named "BlackLeaf". For this reason, some said it was a new species and others claimed to be the same in agreement with the already said small differences being irrelevant to them. To overcome this argument, they used many populations from the "BlackLeaf" species and two "GreenLeaf" individuals, using RAD-Seq.

In order to solve this same problem, our team planned an approach. Considering the learnt sequencing analysis techniques during our classes, we chose to apply High Throughput Sequencing (HTS) methods.

High-throughput-sequencing technology is often used in the study of gene expression. Twenty years ago, what was then called "high-throughput sequencing" was applied to the sequencing of partial-length DNA copies of "expressed sequence tags," in order to generate a fractional view of the repertoire of expressed genes in a sample. Today, the same is done, albeit on a much grander scale and under the new term of RNA-seq, which is essentially sequence data representing those genes that are active in a given biological sample (Progress in Molecular Biology and Translational Science, 2012).

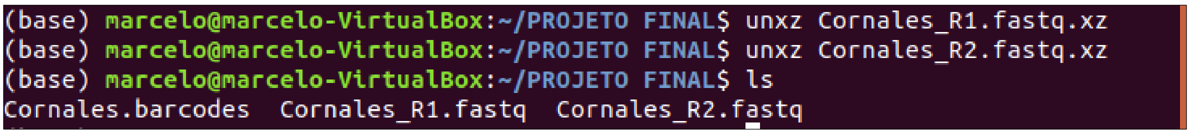
To further continue our investigation, the dataset we used was the one provided by these Botanists which included, as mentioned, both the "GreenLeaf" and "BlackLeaf" species.

Materials and Methods

In this part of the project we will procced to a detailed procedure of what we did. In the beginning, we were given two fastq files and the respective barcodes. On these files we could find the datatype which was ddRAD and because of the split files we knew they were pair-ended. These required an adjustment on the code as we will later confirm. Each fastq file contained 1 million reads.

Starting with the download of the files, we used GIT to clone them and then extract them (Figure 1). For this step we recommend checking if the needed utilities for extraction are installed on the machine and then use the command:

```
unxz filename.xz
```



```
(base) marcelo@marcelo-VirtualBox:~/PROJETO FINAL$ unxz Cornales_R1.fastq.xz
(base) marcelo@marcelo-VirtualBox:~/PROJETO FINAL$ unxz Cornales_R2.fastq.xz
(base) marcelo@marcelo-VirtualBox:~/PROJETO FINAL$ ls
Cornales.barcodes  Cornales_R1.fastq  Cornales_R2.fastq
```

Figure 1: Unzipping both fastq files and all files needed

The next step is to setup an ipyrad environment activated by conda with the command:

```
conda activate ipyrad
```

Then a parameters file is needed so we continued to use shell in order to create one, using:

```
ipyrad -n parametersfilename
```

Now that we had the parameters by default, we needed to do adjustments accordingly with our files and file paths. This will be shown with our filenames but it’s just an example (Figure 2). This will remain valid for all the figures in this project. Also, the datatype was not the default one, so this needed to change too. Both files were read because we used “*” on the file path. This allows us to tell the machine that all files with the given extension must be read as input files.

```
------- ipyrad params file (v.0.9.53)-------------------------------------------

projeto                        ## [0] [assembly_name]: Assembly name. Used to name output
directories for assembly steps

/home/marcelo/projetofinal     ## [1] [project_dir]: Project dir (made in curdir if not
present)

/home/marcelo/projetofinal/*.fastq     ## [2] [raw_fastq_path]: Location of raw non-
demultiplexed fastq files

/home/marcelo/projetofinal/Cornales2.barcodes     ## [3] [barcodes_path]: Location of
barcodes file

              ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files

denovo        ## [5] [assembly_method]: Assembly method (denovo, reference)

              ## [6] [reference_sequence]: Location of reference sequence file

ddrad         ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.

TGCAG,        ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)

5             ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read

33            ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very
standard)

6            ## [11] [mindepth_statistical]: Min depth for statistical base calling

6            ## [12] [mindepth_majrule]: Min depth for majority-rule base calling

10000        ## [13] [maxdepth]: Max cluster depth within samples

0.85         ## [14] [clust_threshold]: Clustering threshold for de novo assembly

0            ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in
barcodes

2            ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)

35           ## [17] [filter_min_trim_len]: Min length of reads after adapter trim

2            ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences

0.05         ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus

0.05         ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus

4            ## [21] [min_samples_locus]: Min # samples per locus for output

0.2          ## [22] [max_SNPs_locus]: Max # SNPs per locus

8            ## [23] [max_Indels_locus]: Max # of indels per locus

0.5          ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus

0, -1, 14, -1   ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)

0, 0, 0, 0      ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)

*      ## [27] [output_formats]: Output formats (see docs)

       ## [28] [pop_assign_file]: Path to population assignment file

       ## [29] [reference_as_filter]: Reads mapped to this reference are removed in step 3
```

File 1: Complete parameters file

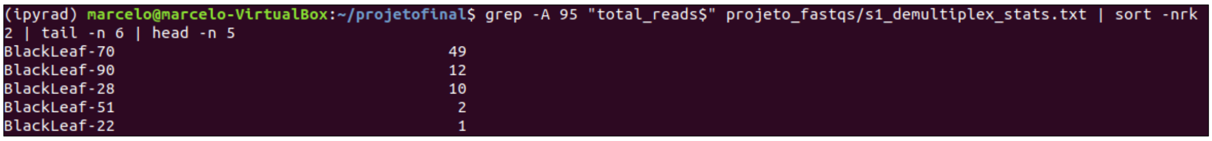
This procedure set us up to continue with the 7 steps used for the analysis, but after step 1, which matches the code:

```
ipyrad -p parametersfilename -s 1 -c 6
```

We realized one of the taxa, BlackLeaf-82, had no match on the barcodes file. This taxon was removed to avoid conflict in following steps, and it wouldn't matter anyway because even if it was one of the individuals of "GreenLeaf" since we had no biological information associated with it, there was no way we could analyse it. On the steps code the -s stands for each step and it's just an indicator for the machine regarding the one we want to run. Finally, the -c stands for cores and since we had 6 available to allocate that's where it comes from. Still in this part of the analysis, we decided to check the total reads as it was advised by the guide provided by the teacher (https://stuntspt.gitlab.io/asb2020/assignments/Assignment01.5.html). To do so, we used the command:

```
cat parametersfilename_fastqs/s1_demultiplex_stats.txt
```

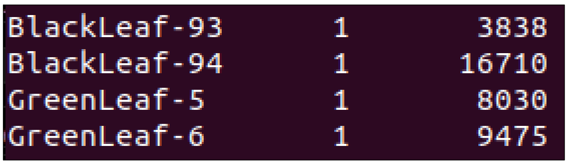
We found out that same taxa had low number of total reads which meant that it didn't contain enough reads to be relevant. Since we only wanted relevant analysis, we had to delete them. The easier way to reduce the list was creating a file with the samples we wanted removed and then creating a new barcodes file. The command to perform such task is shown in Figure 3 and so are the taxa we removed.



```
(ipyrad) marcelo@marcelo-VirtualBox:~/projetofinal$ grep -A 95 "total_reads$" projeto_fastqs/s1_demultiplex_stats.txt | sort -nrk
2 | tail -n 6 | head -n 5
BlackLeaf-70                                49
BlackLeaf-90                                12
BlackLeaf-28                                10
BlackLeaf-51                                 2
BlackLeaf-22                                 1
```

Figure 2: Taxa removed, and the command used to separate the taxa with lower reads from the rest. It was then sorted from the highest to lowest

For comparison, we decided to show some of the individuals with higher number of reads, as shown in Figure 4.



```
BlackLeaf-93        1        3838
BlackLeaf-94        1       16710
GreenLeaf-5         1        8030
GreenLeaf-6         1        9475
```

Figure 3: Taxa with higher reads

Now we were ready to run step 2, which filters the reads. But again, we had to change the default values for the "trim_reads". To get the right values we used a tool from aliview that allows us to see the part of the sequencing that is relevant and its corresponding index. These numbers are then changed in the parameters file for each fastq file.

From step 2 until the before last one, step 6, there is no need to do changes as these are analytical steps. During them, the files that we had to prepare are analysed and on step 7 we decide which format or formats we want the output. There are a couple of them by default. We decided to get them in all available formats as we hadn't decided which format was going to be needed for further visual analysis.

Finally, to better understand the main objective which was if the individuals were from separate species or not, we wanted to build trees to visualize and compare the differences that we will later discuss. For this, *Figtree* was our program of choice as it was the program we had already worked with before. Although *Figtree* was the program we needed to visualize, the output of the analysis didn't have a tree, so we still needed to build it. The first tree we built was with RAxML and the following command:

```
raxmlHPC-PTHREADS-AVX -f a -m GTRCAT -p 112358 -x 112358 -N
100 -s /path/to/your/aligned/file.fasta -n runname
```

To make possible the comparison we needed a second tree which was built with MrBayes method on an online platform, CIPRES (http://www.phylo.org/), that allows us to use a virtual machine with thousands of cores so the processing time is significantly lower. The output we are looking for from this platform is the Bayesian tree.

Results/Discussion

Starting with the tree we obtained with RAxML, we can easily distinguish the GreenLeaf branch from the other branches, which are representing the BlackLeaf. The tree is supported with bootstrap values as shown is Figure 6.

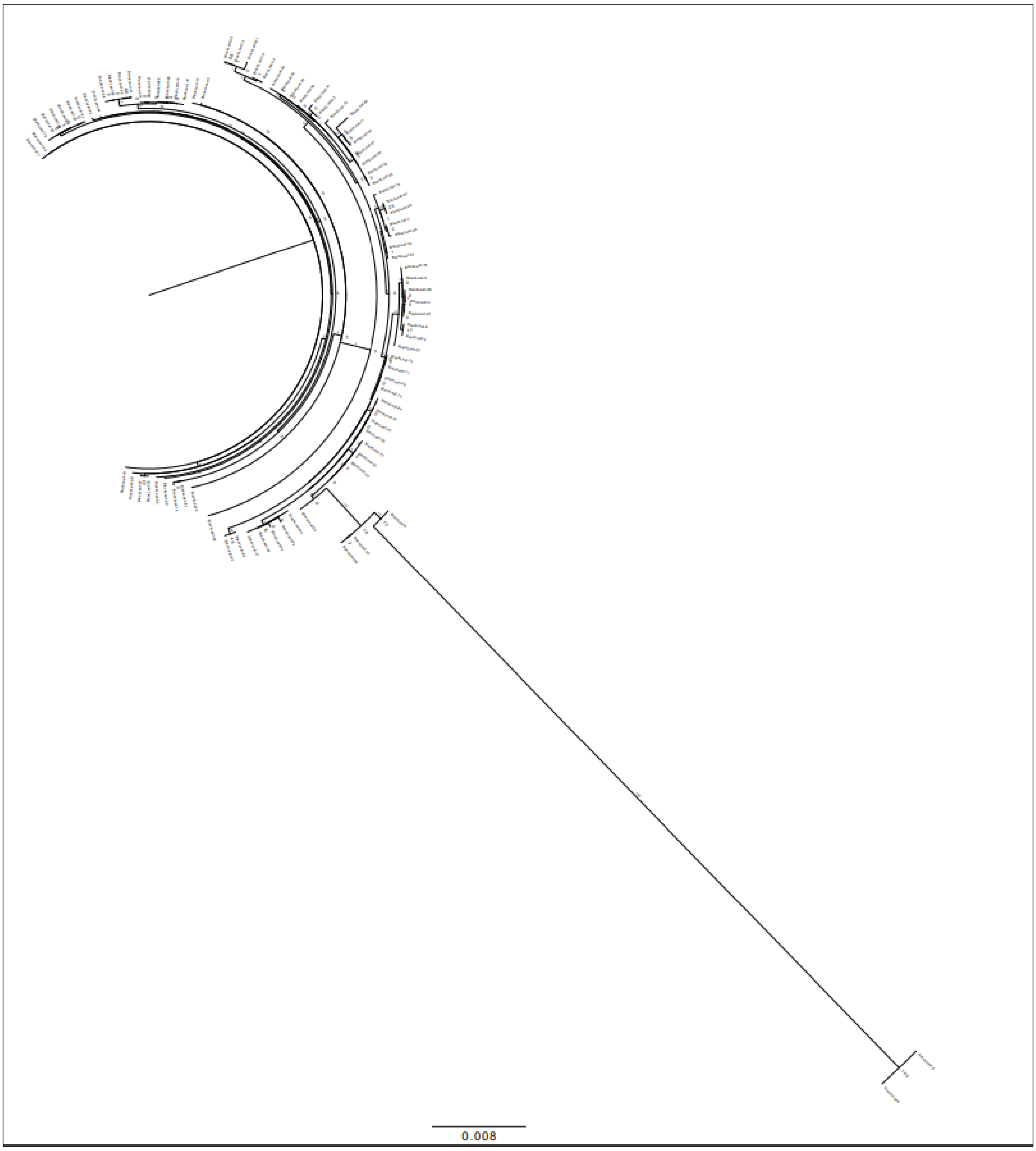


Figure 4: Tree obtained with RAxML where we can clearly see that the small groups (GreenLeaf) are far from the BlackLeaf groups

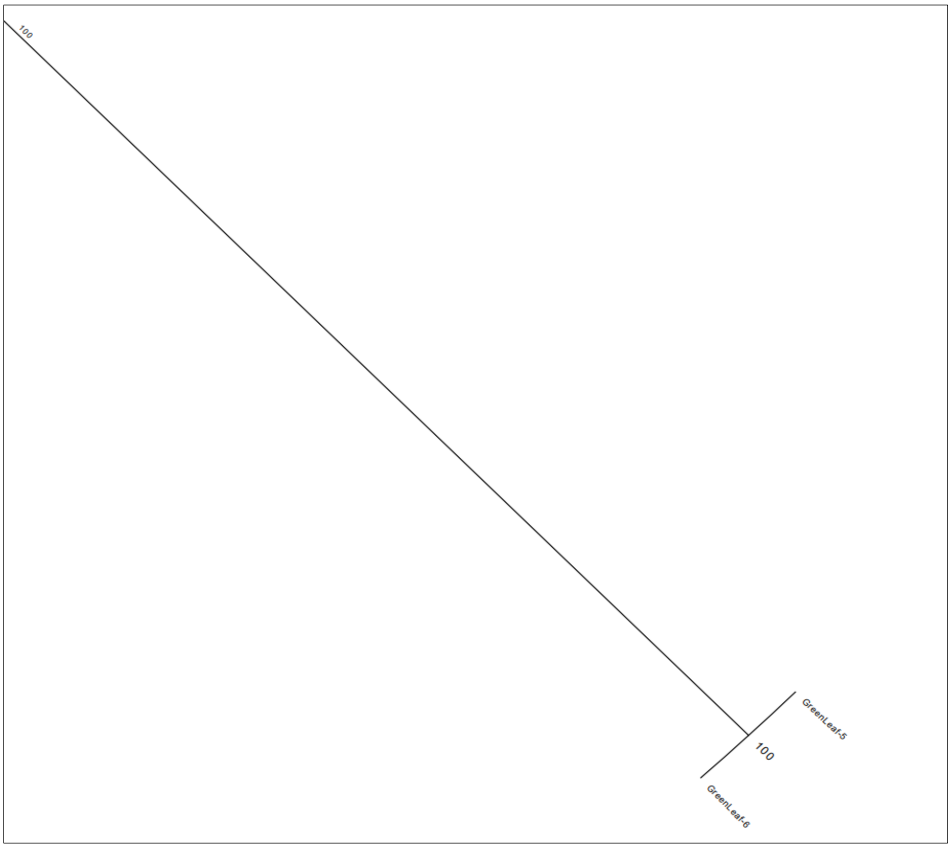


Figure 5: Bootstrap values for the RAxML tree, zoomed on the GreenLeaf branch

As one tree wasn't enough to make conclusions and comparisons, here is, the already mentioned, Bayesian tree built with MrBayes methodology.

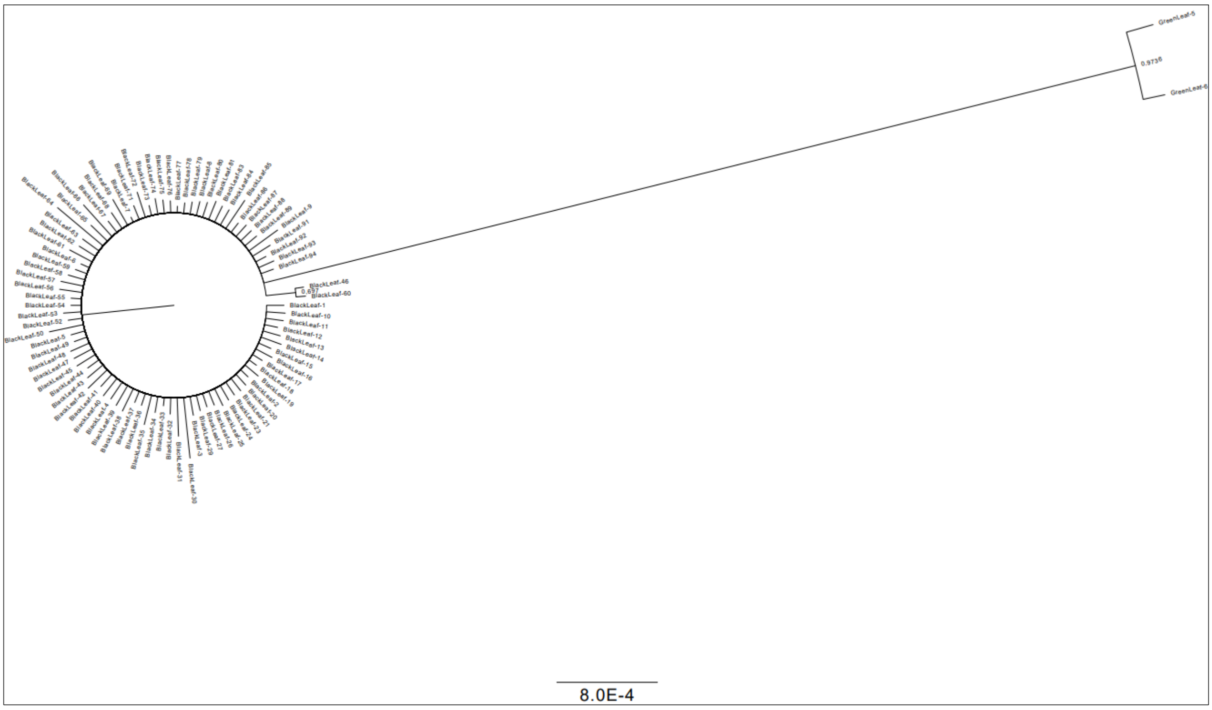


Figure 6: Bayesian tree built with MrBayes. We can, once again, clearly distinguish the GreenLeaf branches from the BlackLeafs

Concerning, the Bayesian tree, it's a consensus tree that doesn't have a chronological relation. It means that it is a consensus from all the trees into one. We obtained a posterior probability value of 0.9736 and it means that in 97% of cases both GreenLeaf individuals appear together.

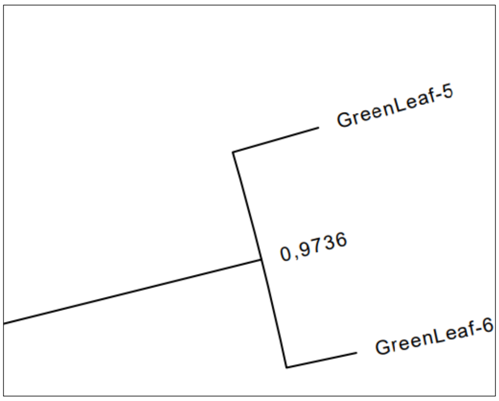


Figure 7: GreenLeaf branch of the Bayesian tree. It is the posterior probability value for this branch, and it means that in 97% of cases these 2 individuals appear together

As final thoughts, our results suggest a clear difference between the GreenLeaf individuals and the BlackLeaf population since with only the genetical information we cannot make clear assumptions about them being different species or not. We would still need to evaluate more information such as phenotype. This kind of analysis gives a good idea to investigators to make further researches on the field and have better and more supported statements, even without the chronological relation.

References

(2012). In l. D.Parnell, *Progress in Molecular Biology and Translational Science* (pp. 17-50).

CIPRES. (n.d.). Retrieved from http://www.phylo.org/

Martins, F. P. (2020, May). *Gitlab*. Retrieved from https://stuntspt.gitlab.io/asb2020/assignments/Assignment01.5.html