Pipeline DADA2 tutorial.

All the information presented here is based on the DADA2 tutorial[1], the steps to execute the script and the fine details are presented in this document. We use SILVA_SSU_r138_2019.RData for IDTaxa[2]. It is assumed that the necessary data was previously downloaded.

Step 1 (Installing libraries)

First, several libraries are needed: Biostrings, stringr, ggplot2, dada2, DECIPHER, and BiocManager. You can install them using the Packages tab in RStudio. In the Script, there is a section called "Install_Libraries Section" to install some packages using console. Lines 4 to 10.

Step 2 (loading libraries)

Once the necessary packages were installed, we proceed to load them using the function: library(desired_package), in the script are the lines 13 to 17.

Step 3 (choosing paths)

We need to choose paths to get and set data, executing lines 20 to 23, dialog boxes appear to select folders and/or files. Lines 25 to 35 are to create result storage folders, these folders are going to be stored in the same folder where the dataset lies.

Lines 20 to 23:

- **Dataset_dir** to select the dataset location.
- **IDTaxa_Directory** to select the IDTaxa file to be used.
- **GetData_Path** select the fastq files location.
- **SetData_Path** select the fastq filtered storage.

Step 4 (loading fastq files)

Lines 38 and 39 serve to load fastq files, the common format for fastq files is SAMPLENAME_1. Fastq and SAMPLENAME_2.fastq, for forward and reverse respectively, if you have different formats, you have to change *pattern="_1.fastq"* and *pattern="_2.fastq"* with the appropriate ending**. There may be cases where there is only one file, in this case, you just use file *"pipelineDADA2_SingleFile.R".* The workflow for this file is the same described here, the difference is you are working only with single files.** We need to save the reading files order to identify samples for control group and case group, to do this just execute line 43.

Step 5 (plotting quality profile)

Lines 46 and 50 plot the data loaded, this information can be used to determine cutoff values for filtering, at this moment, this is based on the observation of the graphs. At the same time, you can storage these graphs using the lines 47 and 51.

Step 6 (filtering)

Lines 55 and 56 place filtered fastq in the path selected at the beginning, the output name for each filtered file is SAMPLENAME_F_filt_fastq.gz for forward and SAMPLENAME_R_filt_fastq.gz for reverse. Filtering is performed using the function called filterAndTrim() using the fastq files, the path for filtering files, the trunc and/or trim parameters corresponding to the dataset, and others parameters, lines 61 to 63. Line 65 save the filtered results in a .csv file. You can plot and save filtered results using lines 68 to 71 to observe the filtered sequence and validate the parameters chosen for the filterAndTrim() function, if the results are not what you want, execute lines 61 to 63 modifying parameters. **It can take time!!!**

Step 7 (error rate)

Lines 74 and 75 compute the error rate learning from the data filtered, lines 76 and 77 save the resulting error rate in a .csv file. You can print the error rate, if you considered necessary, uncommenting lines 79 and 80. It is always worthwhile, as a sanity check and nothing else, to visualize the estimated error rates and save the plots (lines 83 to 86). **It can take time!!!**

[1] https://benjjneb.github.io/dada2/tutorial.html

[2] http://www2.decipher.codes/Classification/TrainingSets/

Step 8 (DADA inference)

Lines 89 and 90 serve to apply the core sample inference algorithm. This function returns an object with sequence variants inferred from input unique sequences and multiple diagnostics about the quality of each denoised sequence variant. For this purpose, you have to execute lines 93 and 94 to inspect the DADA objects. **It can take time!!!**

Step 9 (merge)

Line 97 merges the forward and reverse reads to obtain the full denoised sequences**. In the single file case, this step is not necessary and is not present in the code for single files.** Merging is performed by aligning the denoised forward reads with the reverse-complement of the corresponding denoised reverse reads. By default, merged sequences are only output if the forward and reverse reads overlap by at least 12 bases, and are identical to each other in the overlap region. Execute line 99 to inspect the merger data. **It can take time!!!**

Step 10 (sequence/abundance matrix)

Lines 101 to 104 construct the sequence/abundance matrix and stored in a file called "seqtab_pre_removal.csv", in this file lies all the information. Execute lines 106, 108 and 110 to split the information to be used in machine learning algorithm, these lines generates two files "data_0_chim.csv" and "features_0.csv". In line 113 we can inspect the distribution of sequence lengths. It is important to say, at this point, these files have chimeras.

Step 11 (remove chimeras)

Lines 116 to 122 perform this step and stored in a file called "seqtab_post_removal.csv",. The core dada method corrects substitution and indel errors, but chimeras remain. Fortunately, the accuracy of the sequence variants after denoising makes identifying chimeras simpler than it is when dealing with fuzzy OTUs. Chimeric sequences are identified if they can be exactly reconstructed by combining a left-segment and a right-segment from two more abundant "parent" sequences. As in step 10, lines 124 to 128 split the information to be used in machine learning algorithm, these lines generates two files "data_0_Nochim.csv" and "features_0.csv". **It can take time!!!**

Step 12 (sanity check)

As a final check of our progress, we'll look at the number of reads that made it through each step in the pipeline. To do this execute lines 131 to 137.

Step 13 (assign taxonomy)

Remember, we have two sequence/abundance matrix: with chimeras and without chimeras, considering the last, you have two sections for this step: taxonomy withchim (lines 140 to152) and taxonomy nochim (lines 155 to 167). The process is the same for both, you have to create a DNAStringSet, load the IDTaxa file (selected at the beginning in IDTaxa_Directory) and perform the IdTaxa function (line 142 and 157), define ranks of interest (lines 143 and 158), convert the output object of class "Taxa" to a matrix analogous to the output from assignTaxonomy (lines 145 to 151 and 160 to 166) and finally, save the taxaid (lines 152 and 167). **It is important to mention, both TaxID_chimera.csv and TaxID.csv are very important.** The difference between section is the data used: seqtb for withchim section and seqtab.nochim for nonchim section. **It can take time, sometimes days!!!**

Step 14 (print assignments)

Finally, you can print in console the taxa id withchim and nochim and save them in a .csv file (lines 170 to 177).