Lamborgenei

Manual

Table of contents

Lamborgenei manual
By tom Luijts

Installation & dependencies

Installation

1. Place the downloaded folder in a location where you have read and write permission.
2. Add the 'lamborgenei' file in the Program folder to your PATH

Dependencies

This program is to be used in the Bash UNIX Shell.

- The fastq-dump command from the SRA Toolkit should be in your PATH.

Make sure you have R installed with following packages:

- Dada2
- Tidyverse

Make sure python is installed on your machine. These Python modules are used:

- Pandas

Usage

Preparation

This program is intended to be used along the ncbi *'Sra Run Selector'* tool. Click here for a comprehensible video on how to use this tool.

The required input is an SRA Run Table generated by the SRA Run Selector tool from PAIRED END 16S rRNA sequences. In order to achieve these requirements, a few filters are important:

Requirements:

- **LibraryLayout:** PAIRED
- **Assay Type**: amplicon
- **LibrarySelection**: PCR

Recommendations to make sure you have 16S rRNA

- If there is a column **datatype**: 16S

Since column names between bioprojects are inconsistent in the SRA database, it's your own responsibility to make sure you only use compatible data.

Once you are happy with your SRA run selection, you ought to download the *RunInfo Table* (NOT Accession List). This Will download a .txt file called 'SraRunTable.txt'. This file will be used as the input for the program. Make sure you remember the directory where it's located since this path is the only required command line argument for the program.

Always execute the program from a working directory where you have read and write permissions.

Command line options

The program can be called with 'lamborgenei' (when the Program folder is in your path).

positional arguments:

input Directory where SraRunTable.txt input file is located. "*/Path_to_file/filename"*

optional arguments:

-h, --help show help message and exit

-c [COLUMN COLUMN ...], --columns [COLUMN COLUMN ...]

Specify column names that should be in the samples_SRA.csv file. It's recommended you use a column that can identify the sample's place of isolation. replace spaces in column names by underscores. Multiple column names can be used as input, separated by spaces. If the option remains unspecified the program will automatically search for the column names from a default list (see samples_SRA.csv).

--output *output directory*, -o *output directory*

Output directory, default is current working directory.

--threads *number*, -t *number*

Number of threads used **for downloading runs** from ncbi, default is 1. Note that whatever you choose for this command line option, dada2 (used for processing and classifying the .fastq sequences) uses all threads. Multithreaded performance does not scale perfectly with the number of threads used, but for large queries the difference in runtime is significant.

--keep *directory*, -k *directory*

Directory you want to store the downloaded fastq files. They will be deleted by default.

--database directory/filename, -d directory/filename

Specify the directory and filename of a costum database. By default it will used the database that comes with this program (./DB/BigData.fa.gz). More info on the costum database can be found in the 'database' section.

Output

The output is written in 3 different files. These will be stored in your current working directory or the output directory you have specified in the *--output* command line argument. You can also keep the downloaded fastq files with the *--keep* command line argument.

These 3 output files are meant to be used in the tidyamplicons R package (created by Stijn Wittouck). However, as they are provided as .csv files, feel free to process them in any way you see fit.

abundance_SRA.csv

This table provides for every sample_id (= SRA Run id) whether a sequence (taxon_id) is present and it's abundance.

abundance_SRA.csv can be linked with the samples_SRA.csv via the sample_id column.

abundance2_SRA.csv

The same as abundance_SRA.csv but in a spreadsheet layout. For better integration with tidyamplicons.

Samples_SRA.csv

The aim is to provide every Run (sample_id) with a relevant place of isolation. Since column names in the SRA database are not consistent between BioProjects, the program tries to find a few different column names that should contain relevant information about where the sample was taken.

["Organism","host","host_phenotype","env_biome","env_material","env_feature","specific_host","BioProject"]

If you use the command line argument *--columns*, the program discards previous list and searches for the column name you specified. If it doesn't find any of the column names you provided, it'll use the list again. It's recommended you look in the SraRunSelector for relevant columns and add them to the samples_SRA.csv file using the *--columns* command line argument (replace spaces in column names by underscores).

Note that you can choose columns that have nothing to do with place of isolation to be included in the samples_SRA.csv file. You can choose them as you see fit.

taxa_SRA.csv

Gives the sequence (taxon_id) and the full classification output from the dada2 script.

Taxa_SRA.csv can be linked with the abundance_SRA.csv table via the taxon_id column.

Database

The default database that's used is a combination of two databases.

The first one is a filtered version of the silva taxonomic training data formatted for dada2. It's filtered in a way that only one bacterial 16S rRNA sequence is included for every bacterial genus.

The second database is based on a wgs *Lactobacillus* database provided by Stijn Wittouck where the 16S rRNA from every entry was extracted using barrnap (by Torsten Seemann). From here we filtered out the sequences that are under 1400 bp long. Next some sequences were removed as to only leave one 16S rRNA sequence for every *Lactobacillus* species. The species name was then replaced by a subgenus name based on a paper by *Duar et al*. [1].

You can use your own database by using the command line option '--database' or '-d'. You'll have to specify the directory/filename. Make sure your costum database is in accordance with the DADA2 requirements. Fasta headers must contain the following taxonomical levels, seperated by a semi-colon: Kingdom;Phylum;Class;Order;Family;genus;Species. The database file must be a zipped (.gz) fasta file (.fa).

The default database directory location is: /path_to_program_directory/DB/

Pitfalls

If the program, for some reason fails, don't worry. I've found that the second time you download the SRA fastq data, it always goes a lot faster than the first time. This is probably due to the ncbi server setup. Keep this in mind.

Paired data

The data must be paired sequences, meaning there must be a forward and reverse read available in the SRA database. These reads will be separated in two files (*run_1.fastq* for forward reads and *run_2.*fastq for reverse reads) and used in the dada2 R program. However, sometimes the label in the **LibraryLayout** column indicates that a certain run is paired even though there is only one file downloaded (forward or reverse). If this happens, dada2 will stop and give an error message like this:

$$Error\ in\ filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen\ =\ a, maxN\ =\ b,\quad :$$

$$Paired\ forward\ and\ reverse\ input\ files\ must\ correspond.$$

$$Execution\ halted$$

If this happens, run the program with the '-k [directory]' command line option. This will store the retrieved fastq files in the specified directory. From there you can manually look up which run has a missing *_1.fastq* (forward) or *_2.fastq* (reverse) suffix. Afterwards you should remove this run from the SraRunSelector and download the SraRunTable again. Rerun the program with the exclusively paired data.

16S rRNA

The reference database only contains 16S rRNA of a large variety of bacteria. The unique characteristic of this database is it's ability to classify bacteria from the genus *Lactobacillus* up to a subgenus level.

If the provided fastq files are not 16S rRNA sequences, the program (dada2) won't classify correctly. You'll probably end up with a bunch of 'NA' in your classification table at Domain or Phylum level.

It's your own responsibility to provide only runs containing paired 16S rRNA data. You can check the description of the corresponding BioProject to figure out what DNA piece is amplified. Other methods of checking for 16S rRNA are explained in the Usage section of this manual.

Unsuccessful download

Sometimes, the fastq-dump fails to download the .fastq file. When this happens the program will retry to download it up to 3 times. If the download keeps failing the program will move on without that Run.

Minimum input

The program will not work if only one SRA Run is supplied. The dada2 script needs at least 2 Runs to produce an output. Fastq files of insufficient quality will be dropped so make sure you at least provide 2 high quality Runs. This quality filtering is based on the dada2 pipeline.

Credits and Citations

Huge Thanks to Stijn Wittouck for his guidance and supervision of this project. He was essential to conceptualising and debugging the code. He also provided the basis for the lactobacillus database.

Alexander Van Uffelen provided the adapted dada2 script.

Jaro Verbeeck helped with the R coding and making sure the output can be easily visualised in R.

Tom Luijts compiled the database and made a working pipeline.

The silva 16S rRNA Database provided us with the rest of the reference 16S rRNA sequences to compile our own database.

Torsten Seemann for his barrnap script

[1]. Duar, R.M., Lin, X.B., Zheng, J., Martino, M.E., Grenier, T., Perez-Munoz, M.E., Leulier, F., Ganzle, M. & Walter, J. (2017). Lifestyles in transition: evolution and natural history of the genus Lactobacillus. *FEMS Microbiol Rev*, 41(Supp_1), S27-S48.