README

Rhea

- normalize
  - normalize-table.R
- beta-diversity
  - beta-diversity.R
- alpha-diversity
  - alpha.diversity.R
- correlation
  - OTUs_correlations.R
- numtax
  - Taxanalzyer.R

INTRODUCTION

The importance of microbial 16S rRNA sequence amplicon profiles in understanding the influence of microbes in a variety of environments coupled with the steep reduction in sequencing costs led to a surge of microbial profiling sequencing projects. The expanded user base of such profiles is facing an intimidating range of multipurpose software platforms for analysing their data. Depending on expertize and often in personal taste different solutions are utilized. Across the available options, the R programming language and software environment for statistical computing, stands out for its power and increased flexibility to adhere to the most recent best practices and adjust to the individual project needs. Here we present Rhea pipeline, a set of R scripts that encode a series of well documented choices for the downstream analysis of Operational Taxonomic Units (OTUs) tables. This is both a straightforward starting point for beginners and a framework for advanced users to modify and expand. As the community standards evolve, Rhea will adapt to always represent the current state of the art in microbial profiles analysis in the clear and comprehensive way allowed by the R language. Rhea scripts and documentation are freely available from https://github.com/Lagkouvardos/imngs-toolbox.

REQUIREMENTS

Rhea package is written in R.

To run the R package the following step is necessary:

- Install R
  https://www.r-project.org/

  or

- Install R Studio Desktop and R (simplify usability)
  https://www.rstudio.com/products/rstudio-desktop/

Beside technical requirements the pipeline needs three input files:

1. OTU table

This table shows abundance values of each OTU over all samples. The numbers are absolute numbers where zero is indicating the absence of a particular OUT.

Each row is representing one OTU while samples are shown column wise.

The last column is always showing the assigned taxonomy information. Please note that **taxonomies** have to be delimited with exact **six semicolons**.

OTU	Sample1	Sample2	Sample3	Taxonomy
OTU_1				
OTU_8				
OTU_20				

For more information on the format please have a look at the provided *EXAMPLE-files*.

2. Tree

A phylogenetic tree in Newik format (EXAMPLE-tree.nwk) or Tree format (EXAMPLE-tree.tre). Trees can be generated with MEGA – Molecular Evolutionary Genetic Anaylsis (http://www.megasoftware.net/). There are different approaches for the computation of a phylogenetic tree.

- **Neighbor joining**

  Distance matrix method using genetic distance as a clustering metric. Assums no constant rate of evolution.

- **Maximum parsimony**

Useful approach where not every event is equally likely. The methods identifies potential phylogenetic trees with the smallest number of evolutionary events.

- **Maximum likelihood**

The method assigns probabilities to particular possible phylogenetic trees. It is well suited for the analysis of distantly related sequences.

- **UPGMA**

Unweighted Pair Group Method with Arithmetric mean. The distance from the root to every branch are equal.

- **Minimum Evolution**

The score is calculated on pair-wise distances. Note: information could get lost with this construction.

For the analysis it is recommended to use **Maximum likelihood**. For more information about the different construction methods please see http://www.megasoftware.net/docs

3. Meta file

The Meta file (Mapping file) includes grouping information. The first column corresponds to samples IDs, each row represents one sample. Groups can be assigned column-wise to the samples. The table need to be saved as tab delimited text file.

Grouping information is denoted as a character string or a combination out of numbers and characters.

Note: Groups must contain at least one letter. A group consisting only of numbers is going to be ignored.

#SampleID	Group_1	Group_2	Group_3
Sample_1	A	Ab_1	old
Sample_2	A	Ab_2	new
Sample_3	B	Bb_1	new

If you are using IMNGS these input files are generated by default. Otherwise please be sure that all files fulfil the required input format. IMNGS is an analysis platform for NGS amplicon data and publically available (www.imngs.org).

How to run a script

1. Download the github repository on your local drive
2. Unzip the folder
3. Open the folder and the subfolder of interest

Using R Studio:

4. Double-klick on the R script (R files are provided with a blue R icon)
5. R Studio will open and the script is seen in the upper left window
6. Before running the script please read through the particular README file carefully and individualize the script where it is needed.
7. If all parameters are changed mark everything in the script window (Crtl + A)
8. Click on the *green – Run – arrow* to run the script
9. The outputs are available in the current folder as well as in folders where they are needed for further analysis

Using R Desktop App

4. Open R Desktop App
5. Open the R script (Crtl + O)
6. Before running the script please read through the particular README file carefully and individualize the script where it is needed.
7. If all parameters are changed mark everything in the script window (Crtl + A)
8. In the toolbar click on the third icon (“Ausführung Zeile oder Auswahl”)
9. The outputs are available in the current folder as well as in folders where they are needed for further analysis

Where to start

The very first step is to normalize the OTU table. Therefore the R script is available in:

imngs-toolbox/Rhea/normalize/normalize-table.R

Execute the script as described above.

The other scripts can be execute in random order.

TROUBLESHOOTING

Most common error messages:

```
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'EXAMPLE-OUT-table.tab': No such file or directory
```

- R cannot find the file. What to do?
  - Check it the name of the file is set correctly (spelling mistakes, gaps, special characters).
  - Check if R is in the right folder. You will get directory information by typing the following line in the console.

```
getwd()
```