RiboFrame tutorial

riboFrame: An Improved Method for Microbial Taxonomy Profiling from Non-Targeted Metagenomics

Matteo Ramazzotti, Luisa Berná, Claudio Donati, Duccio Cavalieri

https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2015.00329/full

DOI 10.3389/fgene.2015.00329

What is RiboFrame?

RiboFrame is a Linux-based procedure for microbial profiling based on the identification and classification of 16S rRNA sequences in non-targeted metagenomics datasets. Briefly, RiboFrame acts in two steps:

1) Reads overlapping the 16S rRNA genes are identified and positioned onto a model 16S gene by HMMER using Hidden Markov Models. A coverage plot can be created to evaluate whether we have a sufficient number of reads to continue to the following step. The “RiboTrap” script annotates the positions of the reads in a fasta file, preparing the input for the subsequent step;

2) A phylum-to-genus taxonomic assignment is given to all reads by naïve Bayesian classification using RDPclassifier on RDP or SILVA 16S databases. The “RiboMap” script generates microbial abundance profiles based on this classification, either using all the reads available or just a subset from specific locations of the 16S genes (e.g. the popular V1-V3, V3-V5 or V6-V9 regions).

In practice, RiboFrame transforms untargeted metagenomic experiments into barcoded 16S microbial survey experiment, enabling a more focused analysis of prokaryotes by leveraging well-established 16S databases. In most cases, the analysis can be performed on a PC with at least (about) 8 GB of RAM and yields microbial profiles that closely resemble those derived from full DNA samples.

Installation

RiboFrame is hosted at https://github.com/matteoramazzotti/riboFrame.
It can be downloaded as riboFrame.zip from the GitHub interface or by using thegit command

```
git clone https://github.com/matteoramazzotti/riboFrame
```

RiboFrame consists of two primary scripts: "riboTrap.pl" and "riboMap.pl".
Since both scripts are written in Perl, users must have Perl installed on their system.
The scripts can be executed from the Linux command line using the fcommand

```
perl riboTrap.pl
```

RiboFrame procedures depend on two widely used tools:

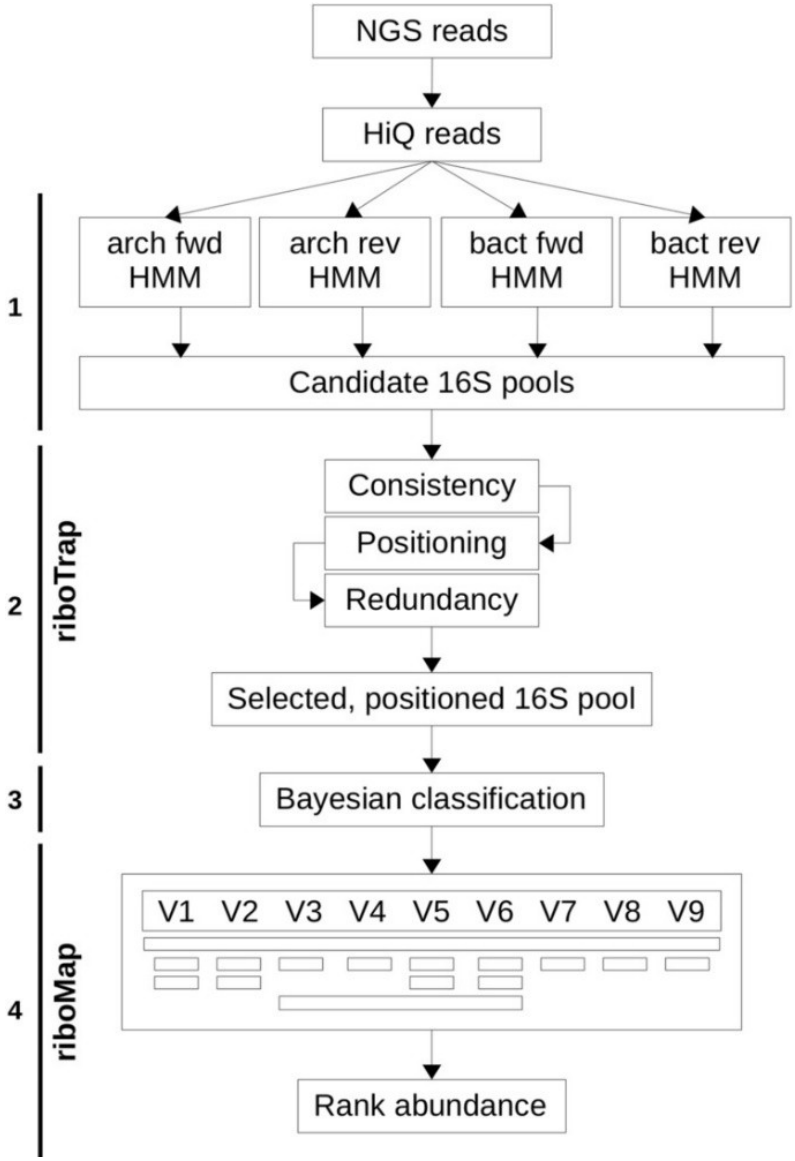
HMMER3 - easily installed with any Linux package manager (eg. apt install hmmer)

RDPclassifier - can be downloaded from http://sourceforge.net/projects/rdp-classifier/ .

The 16S hmmsfor bacteria and archaea are already contained in the downloaded RiboFrame folders. RDP classifier can be directly used to taxonomize sequencing reads using RDP taxonomy or it can be re-trained using external sources suc ash the RDP database or alternative 16S databases such as SILVA.

In case of issues related to permissions, a simple "chmod +rwx ribo* " should be sufficient to enable the analysis

- **RiboFrame pipeline**



- **How to run RiboFrame**

In this tutorial we propose to analyse 2.500 reads from a real metagenomic untargeted paired-end Illumina sequencing experiment. These FASTQ files are named “example_R1.fastq.gz” and “example_R2.fastq.gz” and are shipped with RiboFrame.

The script below assumes that the input FASTQ files, the riboFrame scripts and the RDP classifier folder are present in the same directory.

```
mkdir RiboFr_temp

mkdir RiboFr_output

cpu=4     # set the number of cpu as desired


### transforming the FASTQ in FASTA while solving eventual incompatible headers format…

zcat example_R1.fastq.gz | grep "^@" -A 1 --no-group-separator | sed "s/@/>/g" | sed "s# 1.*#/1#g" > RiboFr_temp/example_R1.fasta

zcat example_R2.fastq.gz | grep "^@" -A 1 --no-group-separator | sed "s/@/>/g" | sed "s# 2.*#/2#g" > RiboFr_temp/example_R2.fasta

# NB: do not remove these fasta from the RiboFr_temp directory until riboTrap.pl is executed


### capturing 16S regions with HMMER3…
# NB: both forward and reverse have to be analysed with the forward and reverse hmm model

# FORWARD

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.R1.fwd.bact.ribosomal.table –noali --cpu $cpu  -o /dev/null riboFrame/hmms/16S_bact_for3.hmm    RiboFr_temp/example_R1.fasta

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.R1.rev.bact.ribosomal.table –noali --cpu $cpu     -o /dev/null riboFrame/hmms/16S_bact_rev3.hmm    RiboFr_temp/example_R1.fasta

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.R1.fwd.arch.ribosomal.table –noali --cpu $cpu -o /dev/null riboFrame/hmms/16S_arch_for3.hmm    RiboFr_temp/example_R1.fasta

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.R1.rev.arch.ribosomal.table –noali --cpu $cpu     -o /dev/null riboFrame/hmms/16S_arch_rev3.hmm    RiboFr_temp/example_R1.fasta

# REVERSE

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.2.fwd.bact.ribosomal.table --noali --cpu $cpu    -o /dev/null riboFrame/hmms/16S_bact_for3.hmm   RiboFr_temp/example.2.fasta

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.2.rev.bact.ribosomal.table --noali --cpu $cpu    -o /dev/null riboFrame/hmms/16S_bact_rev3.hmm   RiboFr_temp/example.2.fasta

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.2.fwd.arch.ribosomal.table –noali --cpu $cpu    -o
```

/dev/null riboFrame/hmms/16S_arch_for3.hmm RiboFr_temp/example.2.fasta

hmmsearch -E 0.00001 --domtblout RiboFr_temp/example.2.rev.arch.ribosomal.table --noali --cpu $cpu -o /dev/null riboFrame/hmms/16S_arch_rev3.hmm RiboFr_temp/example.2.fasta

the following script will flag which reads are 16S according to HMM, using both the forward and reverse fasta along with the hmmer3 outputs generated before…

perl riboTrap.pl RiboFr_temp/example_FASTA

running RDP to classify the sequences…

java -Xmx10g -jar rdp_classifier_2.14/dist/classifier.jar -q RiboFr_temp/example_FASTA.16S.fasta -o RiboFr_output/example_FASTA.16S.rdp

If using a re-trained RDP classifier for taxonomic classification, execute the following command:
java -Xmx10g -jar rdp_classifier_2.14/dist/classifier.jar classify -c 0.8 -t retrained_RDP_folder/rRNAClassifier.properties -q RiboFr_temp/example_FASTA.16S.fasta -o RiboFr_output/example_FASTA.16S.rdp

NB: the expected taxonomic levels in the RDP output are “Domain”, “Phylum”, “Class”, “Order”, “Family” and “Genus”, in this order and with first letter capitalized.

extracting the results according to every 16S sequences (var=full), V1 and V3 (var=V1,V3) or V1,V2 and V3 (var=V1-V3)…

perl riboMap.pl file=RiboFr_output/example_FASTA.16S.rdp var=full conf=0.8 percmin=0.005 out=RiboFr_output/example_FASTA _full_count

perl riboMap.pl file=RiboFr_output/example_FASTA.16S.rdp var=V1,V3 conf=0.8 percim=0.005 out=RiboFr_output/example_FASTA_V1_V3

perl riboMap,pl file=RiboFr_output/example_FASTA.16S.rdp var=V1,V3 conf=0.8 percim=0.005 out=RiboFr_output/example_FASTA_from_V1_to_V3

Parameters:
-var: Specifies the targeted variable regions of the 16S gene
-conf: Minimum classification confidence threshold
-percmin: Minimum abundance threshold for reporting

Cleaning up temporary files and directories to free disk space…

rm RiboFr_temp -r

- **RiboFrame outputs**

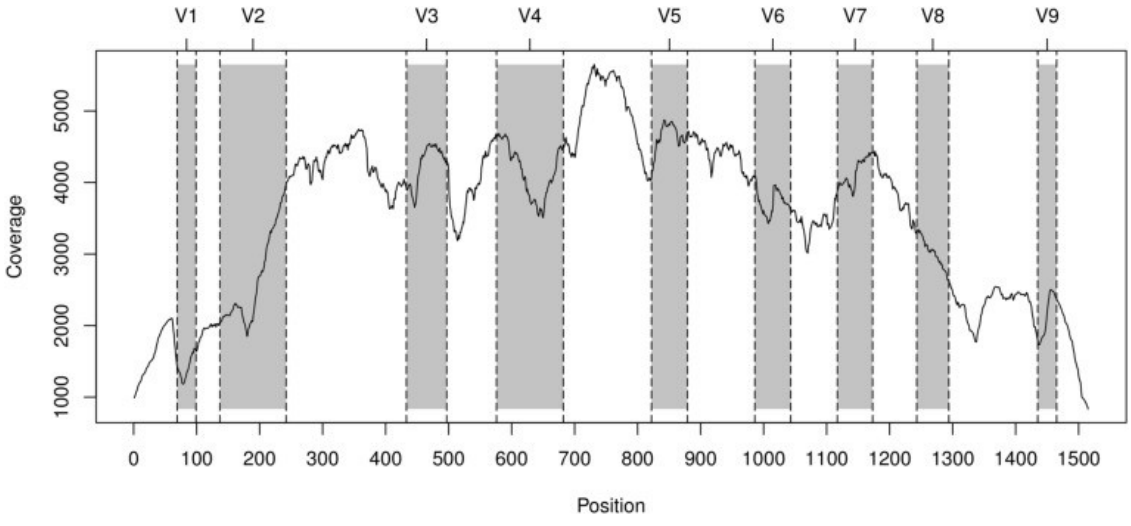
Contingency tables for each taxonomic level are obtained as tsv files (with ".cnt" extension) after running the riboMap script. In case of the example FASTQ in this tutorial, the following genus table is generated from using every variable region in riboMap.

Name	Tot	Mis	Count	Perc
Chiayiivirga	1	0	1	33.33
Culicoidibacter	1	0	1	33.33
Desertimonas	1	0	1	33.33

The column "Tot" represents the total number of reads classified as originating from a specific taxon. The column "Mis" indicates the number of reads that did not meet the user-defined thresholds. The column "Count" is calculated as the difference between "Tot" and "Mis". The column "Perc" represents the percentage abundance in the sample, derived from the "Count" values.

For paired-end reads, the abundance calculation is weighted at each taxonomic rank. If only one read in a pair is classified as ribosomal, it is considered a singleton and assigned a weight of 1, similar to a single-end read. If both reads in a pair are classified as ribosomal, their weight is adjusted to 0.5 so that their combined contribution equals 1, provided they converge on the same taxonomic assignment. Consequently, when paired reads are both classified as ribosomal but assigned to different taxa, the "Count" column may contain decimal values (e.g., 2.5). Given this weighting system, the "Count" column serves as the basis for computing the "Perc" values rather than as the primary output of RiboFrame.

In addition, the coverage of the 16S rDNA gene after region selection can be optionally checked by coverage plots produced by riboTrap, that allows to evaluate whether a sufficient number of reads are available for classification.



The image above is the coverage of the 16S gene achieved with reads form the Human Microbiome Project (HMP) sample SRS011061. extracted with HMM and topology-annotated by riboTrap. The trace represents the cumulative coverage for both paired-end reads after riboTrap processing. Shaded areas identify variable regions that are labelled form V1–V9 in the upper horizontal axis.