

Script P7: Assigning Whole Metagenome Taxonomy

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This protocol provides a method for assigning whole metagenome taxonomy using MetaPhlAn and MEGAN toolkits. Based on the methods from the following publication:

Hannigan, Geoffrey D., et al. "The Human Skin Double-Stranded DNA Virome: Topographical and Temporal Diversity, Genetic Enrichment, and Dynamic Associations with the Host Microbiome." *mBio* 6.5 (2015): e01578-15.

Citation: HANNIGAN GD, GRICE EA, ET AL. Script P7: Assigning Whole Metagenome Taxonomy. **protocols.io**
dx.doi.org/10.17504/protocols.io.egubbww

Published: 10 Mar 2016

Guidelines

Required Software:

- MetaPhlAn v1.7.7
- prinseq-lite-0.20.3
- NCBI's BLAST + v 2.2.0
- MEGAN v5.5.3

Relevant Files

Output:

- MetaPhlAn/skinmet_metaphlan_merged_output_genera.txt
- MEGAN/megan_sk.txt
- MEGAN/megan_genera_bacteria.txt
- MEGAN/megan_genera_eukaryotes.txt
- MEGAN/eukaryote_counts.txt
- MEGAN/megan_genera-viruses.txt

Perl scripts: parse_megan_output.pl

R scripts: [R8](#), [R9](#)

Before start

Perl scripts and other supplementary information available at :

https://figshare.com/articles/The_Human_Skin_dsDNA_Virome_Topographical_and_Temporal_Diversity_Genetic_Enrichment_and_Dynamic_Associations_with_the_Host_Microbiome/1281248

Protocol

MetaPhlAn

Step 1.

Generate a new working directory.


```
cmd COMMAND
mkdir MetaPhlAn_trimmed
mkdir MetaPhlAn_trimmed/input_reads
```

MetaPhlAn

Step 2.

Make function to remove sequences less than 80 nucleotides long using PRINSEQ.

 **SOFTWARE PACKAGE (Unix)**

PRINSEQ, 0.20.3 

Schmieder R and Edwards R

```
cmd COMMAND
remove.short.reads() {
    perl prinseq-lite-0.20.3/prinseq-lite.pl -min_len 80 -
fastq ./clean_phix_fastq/${1}.fastq -
out_good ./MetaPhlAn_trimmed/input_reads/${1}_trimmed -out_bad null
}
```

NOTES

Geoffrey Hannigan 28 Jan 2016

Technically, you can input your raw sequences into MetaPhlAn, but for optimal results, we are using our quality filtered, human and phix decontaminated sequences. Additionally, we want to remove any sequences less than 80 nt long. We do this using the program PRINSEQ.

Geoffrey Hannigan 28 Jan 2016

Technically, you can input your raw sequences into MetaPhlAn, but for optimal results, we are using our quality filtered, human and phix decontaminated sequences. Additionally, we want to remove any sequences less than 80 nt long. We do this using the program PRINSEQ.

MetaPhlAn

Step 3.

Export and run function.

```
cmd COMMAND
export -f remove.short.reads
ls ./clean_phix_fastq/*R1* | sed -e 's/^.*\./.*\\\\/g' | sed 's/\\.fastq//g' | xargs -I {} --
max-procs=128 sh -c 'remove.short.reads {}'
```

MetaPhlAn

Step 4.

Move to the working directory.

```
cmd COMMAND
cd MetaPhlAn_trimmed
```

MetaPhlAn

Step 5.

Make MetaPhlAn function.

 **SOFTWARE PACKAGE (Unix)**

MetaPhlAn, 1.7.8 

Curtis Huttenhower

```
cmd COMMAND
```

```

run.metaphlan() {
    # Look at all taxa assignments
    python metaphlan/metaphlan.py --
bowtie2db metaphlan/bowtie2db/mpa ./input_reads/${1}.fastq --tax_lev 'a' --no_map --
nproc 18 --output_file ./MetaPhlAn_trimmed/${1}_metaphlan_all.txt
    # Look at phyla level assignments
    python metaphlan/metaphlan.py --
bowtie2db metaphlan/bowtie2db/mpa ./input_reads/${1}.fastq --tax_lev 'p' --no_map --
nproc 18 --output_file ./MetaPhlAn_trimmed/${1}_metaphlan_phyla.txt
    # Look at genera level assignments
    python metaphlan/metaphlan.py --
bowtie2db metaphlan/bowtie2db/mpa ./input_reads/${1}.fastq --tax_lev 'g' --no_map --
nproc 18 --output_file ./MetaPhlAn_trimmed/${1}_metaphlan_genera.txt
    # Look at species level assignments
    python metaphlan/metaphlan.py --
bowtie2db metaphlan/bowtie2db/mpa ./input_reads/${1}.fastq --tax_lev 's' --no_map --
nproc 18 --output_file ./MetaPhlAn_trimmed/${1}_metaphlan_species.txt
}

```

MetaPhlAn

Step 6.

Export and run function.

```

cmd COMMAND
export -f run.metaphlan
ls ./input_reads/ | sed -e 's/^.*/.*\///g' | sed 's/\.fastq//g' | xargs -I {} --max-
procs=128 sh -c 'run.metaphlan {}'

```

MetaPhlAn

Step 7.

Merge MetaPhlAn output for individual samples into one data table.

```

cmd COMMAND
python metaphlan/utils/merge_metaphlan_tables.py *all.txt > skinmet_metaphlan_merged_output
_all.txt
python metaphlan/utils/merge_metaphlan_tables.py *phyla.txt > skinmet_metaphlan_merged_outp
ut_phyla.txt
python metaphlan/utils/merge_metaphlan_tables.py *genera.txt > skinmet_metaphlan_merged_out
put_genera.txt
python metaphlan/utils/merge_metaphlan_tables.py *species.txt > skinmet_metaphlan_merged_ou
tput_species.txt

```

NOTES

Geoffrey Hannigan 28 Jan 2016

MetaPhlAn allows you to merge the output for each individual sample into one table, where the rows are the taxa and the columns are the samples.

MetaPhlAn

Step 8.

Generate a biom file from the merged output.

```

cmd COMMAND
python metaphlan/conversion_scripts/metaphlan2biom.py skinmet_metaphlan_merged_output_all.t
xt > skinmet_metaphlan_merged_output_all.biom

```

MetaPhlAn

Step 9.

Finally, we want to clean up our directory so it is easier to work with. We make a directory for each type of output (merged, all taxa, phyla only, genera only, and species only outputs).

```

cmd COMMAND

```

```

mkdir merged_output
mv skinmet_metaphlan_merged_output_*.txt merged_output
mv *biom merged_output

mkdir all_taxa
mv *all.txt all_taxa

mkdir phyla
mv *phyla.txt phyla

mkdir genera
mv *genera.txt genera

mkdir species
mv *species.txt species

```

MetaPhlAn

Step 10.

Now that we have our taxonomy assignments, we can analyze the output in R.

MEGAN

Step 11.

We ran MEGAN on the assembled contigs. To make our job easier, we first put all of the assembled contig fasta files for the individual samples into one directory.

 [SOFTWARE PACKAGE \(Unix\)](#)

MEGAN, 5.5.3 

D. H. Huson

cmd [COMMAND](#)

```

# Move to directory containing the contigs.
cd ./Ray

```

MEGAN

Step 12.

Make a new directory for the contigs.

```

cmd COMMAND
mkdir sample_assembled_contigs

```

MEGAN

Step 13.

Copy fasta files into the new directory.

```

cmd COMMAND
for file in $(ls ./ray_contigs_from_cat); do
    cp ./ray_contigs_from_cat/$file/${file}_Contigs_with_format.fa ./sample_assembled_conti
gs
done

```

MEGAN

Step 14.

Make a new directory for the MEGAN output.

```

cmd COMMAND
mkdir MEGAN

```

MEGAN

Step 15.

Blast the assembled contigs against the NCBI nucleotide database. Use outfmt 5 since it can be directly input into MEGAN.

```

cmd COMMAND
for file in $(ls ./sample_assembled_contigs); do

```

```
VARIABLE=${file/_Contigs_with_format.fa/.xml}
blastn -query ./sample_assembled_contigs/${file} -out ./MEGAN/${VARIABLE} -
db references/ncbi_blast_db/nt -outfmt 5 -evalue 1e-10 -num_threads 12
done
```

MEGAN

Step 16.

Move to the directory containing the input fasta files.

```
cmd COMMAND
cd ./Ray/sample_assembled_contigs
```

MEGAN

Step 17.

Make a directory for the MEGAN output (.rma) files.

```
cmd COMMAND
mkdir output
```

MEGAN

Step 18.

For each sample write a command to the file that will be used to generate a MEGAN file. The MEGAN file is generated by importing the blast output and fasta file.

```
cmd COMMAND
for file in $(ls *.fa); do
printf 'import blastFile=\'Ray/MEGAN/${file/_Contigs_with_format.fa/.xml}\'' fastaFile=\'
Ray/sample_assembled_contigs/${file}\'' meganFile=\'Ray/MEGAN/output/${file/_Contigs_with_
format.fa/.rma}\'' maxMatches=100 minScore=50.0 maxExpected=0.01 topPercent=10.0 minSupport
=5 minComplexity=0.3 useMinimalCoverageHeuristic=false useSeed=false useCOG=false useKegg=f
alse paired=false useIdentityFilter=false;\n' >> Ray/MEGAN/MEGAN_file_generation_commands.t
xt
```

MEGAN

Step 19.

For each sample write a command to the file that will be used to read a MEGAN file and output taxonomic classifications. Taxonomic classifications will be made at the Super Kingdom, Phylum, Genera, and Species levels.

```
cmd COMMAND
for file in $(ls *.fa); do
printf 'open file=\'Ray/MEGAN/output/${file/_Contigs_with_format.fa/.rma}\';\ncollapse rank=S
uperKingdom;\nselect nodes=all;\nexport what=DSV format=taxonpath_count separator=tab count
s=summarized file=\'Ray/MEGAN/output/${file/_Contigs_with_format.fa/_super_kingdom.txt}\';\ns
elect nodes=none;\nuncollapse nodes=all;\nselect rank=Phylum;\nexport what=DSV format=taxon
path_count separator=tab counts=summarized file=\'Ray/MEGAN/output/${file/_Contigs_with_form
at.fa/_phyla.txt}\';\nselect nodes=none;\nselect rank=Genus;\nexport what=DSV format=taxonpa
th_count separator=tab counts=summarized file=\'Ray/MEGAN/output/${file/_Contigs_with_format
.fa/_genus.txt}\';\nselect nodes=none;\nselect rank=Species;\nexport what=DSV format=taxonpa
th_count separator=tab counts=summarized file=\'Ray/MEGAN/output/${file/_Contigs_with_format
.fa/_species.txt}\';\n'>> Ray/MEGAN/MEGAN_taxonomy_commands.txt
done
```

MEGAN

Step 20.

Add a line to the end of the command files that tells MEGAN to exit.

```
cmd COMMAND
printf 'quit;\n' >> ./Ray/MEGAN/MEGAN_file_generation_commands.txt
printf 'quit;\n' >> ./Ray/MEGAN/MEGAN_taxonomy_commands.txt
```

MEGAN

Step 21.

Move to the right working directory.

cmd **COMMAND**

```
cd Ray/MEGAN
```

MEGAN

Step 22.

Generate MEGAN files.

cmd **COMMAND**

```
xvfb-run --auto-servernum --server-num=1 /appl/MEGAN-5.5.3/MEGAN -g -E -  
c Ray/MEGAN/MEGAN_file_generation_commands.txt > Ray/MEGAN/MEGAN_file_generation_commands.o  
ut
```

MEGAN

Step 23.

Extract taxonomic classifications form MEGAN files.

cmd **COMMAND**

```
xvfb-run --auto-servernum --server-num=1 /appl/MEGAN-5.5.3/MEGAN -g -E -  
c Ray/MEGAN/MEGAN_taxonomy_commands.txt > Ray/MEGAN/MEGAN_taxonomy_commands.out
```

MEGAN

Step 24.

Now that we have the taxonomic assignments extracted from MEGAN, we need to parse the output. Move to the working directory.

cmd **COMMAND**

```
cd Ray/MEGAN/output
```

🔗 NOTES

Geoffrey Hannigan 28 Jan 2016

The problem with the way the data is extracted is that classifications do not necessarily carry accross levels. For instance, say you have 100 reads that are assigned to the phylum Actinobacteria and 95 of these reads are classified as *Propionibacterium acnes*, but 5 of the reads were not classified any further than phylum level. The only classification in the species level will be *Propionibacterium acnes* (which is somewhat misleading because we know not only is there unclassified samples, but they belong to a certain phyla). In order to deal with this, we wrote a perl script to parse our MEGAN output. This script can be found in the supplemental scripts and takes in the super kingdom, phyla, genera, and species text files generated by MEGAN_taxonomy_commands.txt. To run this perl script accross all samples, we went back to bash.

MEGAN

Step 25.

Parse MEGAN output.

cmd **COMMAND**

```
for file in *.rma; do  
  echo $file  
  perl parse_megan_output.pl ${file/.rma/_super_kingdom.txt} ${file/.rma/_phyla.txt} ${fi  
le/.rma/_genus.txt} ${file/.rma/_species.txt}  
  # Check to make sure read counts add up accross different taxonomic levels  
  sk_count=`awk '{ sum += $2 } END { print sum }' ${file/.rma/_super_kingdom.txt}`  
  phyla_count=`awk '{ sum += $2 } END { print sum }' ${file/.rma/_megan_phyla.txt}`  
  genus_count=`awk '{ sum += $2 } END { print sum }' ${file/.rma/_megan_genera.txt}`  
  species_count=`awk '{ sum += $2 } END { print sum }' ${file/.rma/_megan_species.txt}`  
  if [ $sk_count -gt 2 ]; then  
    if [ $phyla_count -ne $genus_count ]; then  
      echo "COUNTING ISSUE AT PHYLA AND OR GENUS LEVEL"  
    fi  
  fi  
done
```

```

        if [ $phyla_count -ne $species_count ]; then
            echo "COUNTING ISSUE AT PHYLA AND OR SPECIES LEVEL"
        fi

        if [ $species_count -ne $genus_count ]; then
            echo "COUNTING ISSUE AT GENUS AND OR SPECIES LEVEL"
        fi
    fi

```

done

📌 NOTES

Geoffrey Hannigan 28 Jan 2016

Perl script can be found in the supplementary information found [here](#).

MEGAN

Step 26.

Remove intermediate files.

```

cmd COMMAND
rm superkingdom.txt
rm phyla.txt
rm genus.txt
rm species.txt

```

MEGAN

Step 27.

Organize formatted output into directories.

```

cmd COMMAND
mkdir formatted_output
mv MG*_megan_*.txt ./formatted_output
cd formatted_output
mkdir genera
mkdir phyla
mkdir super_kingdom
mkdir species

mv *genera.txt ./genera
mv *species.txt ./species
mv *phyla.txt ./phyla
mv *sk.txt ./super_kingdom

```

MEGAN

Step 28.

For downstream analyses in R, we wanted data pertaining specifically to bacteria, viruses, and eukaryotes separately. We extracted this information from the formatted files using grep.

```

cmd COMMAND
# Look at viral species level
cd species

for file in *_megan_species.txt; do
    grep "sk__Viruses" $file > ${file/./txt/_viruses.txt}
done

```

MEGAN

Step 29.

Look at bacterial and fungal genera.

```

cmd COMMAND
cd ../genera

```

```

for file in *_megan_genera.txt; do
    grep "sk__Bacteria" $file > ${file/.txt/_bacteria.txt}
done

for file in *_megan_genera.txt; do
    grep "sk__Eukaryota" $file > ${file/.txt/_eukaryotes.txt}
done

```

MEGAN

Step 30.

Look at all three at the phylum level.

```

cmd COMMAND
cd ../phyla

for file in *_megan_phyla.txt; do
    grep "sk__Bacteria" $file > ${file/.txt/_bacteria.txt}
done

for file in *_megan_phyla.txt; do
    grep "sk__Viruses" $file > ${file/.txt/_viruses.txt}
done

for file in *_megan_phyla.txt; do
    grep "sk__Eukaryota" $file > ${file/.txt/_eukaryotes.txt}
done

```