Script P4: Assigning Viral Taxonomy

HANNIGAN GD, GRICE EA, ET AL.

Abstract

This protocol provides a method for identifying bacteriophage/virus taxonomy without a virome dataset using UniProt reference database. Based on the methods found in 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 P4: Assigning Viral Taxonomy. **protocols.io**
dx.doi.org/10.17504/protocols.io.efqbbmw
Published: 10 Mar 2016

Guidelines

Required Software:

- NCBI's BLAST+ v2.2.0
- UniProt Database
- Bowtie2-2.1.0
- MEGAN-5.5.3

Relevant Files

Output:

- Bray-curtis_virome_analysis/contig_otu_table_transposed_formatted.txt
- Phage_Taxonomy/order_rel_abund.tsv
- Phage_Taxonomy/genus_rel_abund.tsv
- Phage_Taxonomy/species_rel_abund.tsv

Perl Scripts:

- remove_block_fasta_format.pl
- filter_fasta_file.pl
- get_format_order.pl, get_format_family.pl, get_format_subfamily.pl, get_format_genus.pl
- contig_id_by_orfs.pl
- calculate_abundance_from_sam.pl

Python script:

- transpose_tab_delim.py

R scripts:

- R1 and R4

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

Analysis

Step 1.

Download the entire Uniprot TrEMBL reference fasta database.

cmd COMMAND

```
mkdir ./references/TrEMBL
cd ./references/TrEMBL
wget ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_trembl.fasta.gz
gunzip ./TrEMBL/uniprot_trembl.fasta.gz
mkdir ../UniProt-Virus-Phage
cd ../UniProt-Virus-Phage
```

Analysis

Step 2.

Download the virus taxonomy reference text file.

cmd COMMAND

```
wget ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/taxonomic_divisions/uniprot_trembl_viruses.dat.gz
```

Analysis

Step 3.

Unzip the files.

cmd COMMAND

```
gunzip ./uniprot_trembl_viruses.dat.gz
```

Analysis

Step 4.

To make the TrEMBL database more managable downstream, we want to pull out the viral reference genes to make a virus specific database. While still in the directory 'UniProt-Virus-Phage', get a list of the IDs associated with each virus protein sequence.

cmd COMMAND

```
egrep '^AC' uniprot_trembl_viruses.dat | sed 's/AC *//' | sed 's/\;//g' > ./virus_accession_trembl_list.txt
```

Analysis

Step 5.

Before getting the sequences that match the accession number list, we need to remove the block format from the fasta files using the perl script remove_block_fasta_format.pl

🔗 LINK:

https://figshare.com/articles/The_Human_Skin_dsDNA_Virome_Topographical_and_Temporal_Divers ity_Genetic_Enrichment_and_Dynamic_Associations_with_the_Host_Microbiome/1281248

cmd COMMAND

```
cd ../TrEMBL/
perl remove_block_fasta_format.pl ./uniprot_trembl.fasta ./uniprot_trembl_no_block.fa
cd ../UniProt-Virus-Phage
```

NOTES

Geoffrey Hannigan 14 Jan 2016

Perl scripts can be found in the supplementary information and on figshare.

Analysis

Step 6.

Use the perl script filter_fasta_file.pl to get all of the sequences with matching accession numbers.

LINK:

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

cmd COMMAND

```
perl filter_fasta_file.pl -l ./virus_accession_trembl_list.txt -
i ./references/TrEMBL/uniprot_trembl_no_block.fa -o ./uniprot_phage_virus_TrEMBL.fa -
id_regex ".*\|(.*)\|.*"
```

NOTES

Geoffrey Hannigan 14 Jan 2016

Perl scripts can be found in the supplementary information and on figshare.

Analysis

Step 7.

Generate blast database from the uniprot references (virus+phage).

cmd COMMAND

```
makeblastdb -dbtype prot -in uniprot_phage_virus_TrEMBL.fa -
out uniprot_virus_and_phage_TrEMBL_db

egrep "^(OC|ID|OS)" ./uniprot_trembl_viruses.dat | sed -e :a -e '$!N;s/\n\(OS\)/ /;ta' -
e 'P;D' | sed -e :a -e '$!N;s/\n\(OC\)/ /;ta' -e 'P;D' > ./tmp1_TrEMBL.txt
sed 's/^.*\(Viruses\)/\1/' ./tmp1_TrEMBL.txt | sed 's/ \+/ /g' | sed 's/\.//g' | sed 's/; /
\t/g' | sed 's/ /_/g' > ./tmp_taxa_tree_TrEMBL.txt
```

Analysis

Step 8.

Pull out the different taxonomic level information.

LINK:

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

cmd COMMAND

```
perl get_format_order.pl tmp_taxa_tree_TrEMBL.txt tmp_order_TrEMBL.txt
perl get_format_family.pl tmp_taxa_tree_TrEMBL.txt tmp_family_TrEMBL.txt
perl get_format_subfamily.pl tmp_taxa_tree_TrEMBL.txt tmp_subfamily_TrEMBL.txt
perl get_format_genus.pl tmp_taxa_tree_TrEMBL.txt tmp_genus_TrEMBL.txt
```

NOTES

Geoffrey Hannigan 14 Jan 2016

Perl scripts can be found in the supplementary information and on figshare.

Analysis

Step 9.

Get list of IDs.

cmd COMMAND

```
sed 's/^ID \+//' tmp1_TrEMBL.txt | sed 's/ .*$//' > tmp_id_TrEMBL.txt
```

Analysis

Step 10.

Get list of "species names"

cmd COMMAND

```
sed 's/^.*AA\. *//' tmp1_TrEMBL.txt | sed 's/Viruses.*$//' | sed 's/ \+/_/g' | sed 's/\._$/
/' > tmp_species_TrEMBL.txt
```

Analysis

Step 11.

Paste together the lists.

cmd COMMAND

```
paste tmp_id_TrEMBL.txt tmp_order_TrEMBL.txt tmp_family_TrEMBL.txt tmp_subfamily_TrEMBL.txt
 tmp_genus_TrEMBL.txt tmp_species_TrEMBL.txt > uniprot_reference_phage_and_virus_TrEMBL_tax
onomy_table.txt
```

Analysis

Step 12.

Remove the tmp files after the script is finished running.

cmd COMMAND

```
rm ./tmp*
```

WARNING: This will remove all files that start with "tmp", so be careful and make sure you don't delete files that you want to keep.

Analysis

Step 13.

Now that we have the virus reference database prepared and formatted, we can start annotating the viruses in our dataset. Instead of annotating each individual short sequence, we are going to annotate our longer contigs.

NOTES

Geoffrey Hannigan 14 Jan 2016

Note that near the end of the loop below, we need to address some name issues. There are duplicates in the TrEMBL dataset due to slight naming differences (i.e. Environmental_Halophage and Environmental_halophage) which could throw off some downstream relative abundance analyses. Therefore we standardize these names across the dataset. We also removed some 'strain' specific information to allow the phages/viruses to be a little more broadly grouped (i.e. Taking the numbers off the end of different Staphylococcus phage names to make them all 'Staphylococcus phage'.

Analysis

Step 14.

In the main working directory make a directory for the uniprot taxonomy results.

cmd COMMAND

```
mkdir ./uniprot_taxonomy_using_orfs
```

Analysis

Step 15.

Perform blastx of the predicted ORFs against the virus/phage uniprot database reference (TREMBL).

SOFTWARE PACKAGE (Unix)

BLAST Toolkit, 2.2.0

NCBI

ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

cmd COMMAND

```
blastx -query ./glimmer3/output/Contigs_no_block_with_names_glimmer_output_final.fa -
out ./uniprot_taxonomy_using_orfs/blastx_trembl_glimmer_total_orfs.txt -
```

```
db ./references/UniProt-Virus-Phage/uniprot_virus_and_phage_TrEMBL_db -outfmt 6 -num_threads 16 -max_target_seqs 1 -evalue 1e-5
```

Use ORF fasta file (from glimmer3)

Analysis

Step 16.

Move blastx output files to a specific directory.

cmd COMMAND

```
mkdir ./uniprot_taxonomy_using_orfs/blastx_raw_results
mv ./uniprot_taxonomy_using_orfs/blastx_trembl_glimmer_total_orfs.txt ./uniprot_taxonomy_using_orfs/blastx_raw_results
```

Analysis

Step 17.

Get the gene and contig IDs from the blastx results (output 6; tab delimited file).

cmd COMMAND

```
mkdir ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa
for file in $(ls ./uniprot_taxonomy_using_orfs/blastx_raw_results); do
```

NOTES

Geoffrey Hannigan 14 Jan 2016

Do this all in a loop to make life easier.

Analysis

Step 18.

Set file name variable.

cmd COMMAND

```
NAME=$(echo ${file})
```

Analysis

Step 19.

Assign contig IDs to tmp list file.

cmd COMMAND

```
cut -f 1 ./uniprot_taxonomy_using_orfs/blastx_raw_results/${file} | sed 's/_.*//' > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp1_${file}
```

Analysis

Step 20.

Assign gene hit IDs to another tmp list.

cmd COMMAND

```
cut -f 2 ./uniprot_taxonomy_using_orfs/blastx_raw_results/${file} | sed 's/.*|.*|\(.*\)/\1/' > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp2_${file}
```

Analysis

Step 21.

Paste these files together for reference.

cmd COMMAND

```
paste ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp1_${file} ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp2_${file} > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_paste_${file}
```

Analysis

Step 22.

Add in the taxonomy data from the uniprot reference database.

cmd COMMAND

```
echo Assigning taxonomy to ${file}...
```

```
    if [[ $NAME =~ .*trembl.* ]]; then
        awk 'FNR==NR { a[$1]=$2"\t"$3"\t"$4"\t"$5"\t"$6; next } $2 in a { print $1"\t"a[$2]
 }' ./references/UniProt-Virus-
Phage/uniprot_reference_phage_and_virus_TrEMBL_taxonomy_table.txt ./uniprot_taxonomy_using_
orfs/virome_phage_blastx_formatted_for_taxa/tmp_paste_${file} > ./uniprot_taxonomy_using_or
fs/virome_phage_blastx_formatted_for_taxa/tmp_awk_cat_${file}
        echo $NAME contains trembl!
    elif [[ $NAME =~ .*swissprot.* ]]; then
        echo $NAME contains swissprot!
    else
        echo $NAME does not contain trembl or swissprot!
    fi
```

Analysis

Step 23.

Add @ symbol delimiter at first tab for easier perl parsing.

cmd COMMAND

```
sed 's/\t/@/' ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_awk_
cat_${file} > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/format_f
or_perl_${file}
```

Analysis

Step 24.

Filter out the contigs that did not have at least one orf match per 10kb. Return the number of orfs that had assigned taxonomy to each contig.

cmd COMMAND

```
sed 's/@.*/_/' ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/format_
for_perl_${file} | sort | uniq -
c | sed 's/^ *//' | sed 's/ /\t/' > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_forma
tted_for_taxa/tmp_orf_count_${file}
```

Analysis

Step 25.

Assemble a list of contig numbers, only including those contigs which had at least 1 orf per 10kb.

cmd COMMAND

```
awk 'FNR==NR { a[$1]=$2; next } $2 in a { print $2"\t"$1"\t"a[$2]"\t"10000*$1/a[$2] }' ./co
ntig_stats/contig_length_without_greater_sign.txt ./uniprot_taxonomy_using_orfs/virome_phag
e_blastx_formatted_for_taxa/tmp_orf_count_${file} | awk '$4 > 1' > ./uniprot_taxonomy_using
_orfs/virome_phage_blastx_formatted_for_taxa/tmp_awk_list_less_10kb_${file}
```

Analysis

Step 26.

Get only the contig ID numbers, clean away the other information from each row.

cmd COMMAND

```
cut -
f 1 ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_awk_list_less_
10kb_${file} > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_awk
_list_less_10kb_clean_${file}
```

Analysis

Step 27.

Remove the underscores from the ends of the contig IDs so that grep can match the entire words.

cmd COMMAND

```
sed 's/\t/_\t/' ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_aw
k_cat_${file} > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_aw
k_cat_no_underscore_${file}
```

Analysis

Step 28.

Use grep to get the contig rows that match the list of contigs to keep.

COMMAND

```
grep -w --file=./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_awk_list_less_10kb_clean_${file} ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/tmp_awk_cat_no_underscore_${file} > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/coverage_filtered_${file}
```

Analysis

Step 29.

Again add @ symbol delimiter at first tab for easier perl parsing on filtered contigs.

COMMAND

```
sed 's/_\t/@/' ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/coverage_filtered_${file} > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/format_for_perl_filtered_${file}
```

Analysis

Step 30.

Remove the strain specific IDs at the ends of the phage and virus species names (for example, staph_phage_0594 is just staph_phage).

LINK:

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

COMMAND

```
sed 's/\(\t.*_phage\).*$/\1/' ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/format_for_perl_filtered_${file} | sed 's/\(\t.*_virus\).*$/\1/' | sed 's/_type.*$//' | sed 's/\(virus\)[^\t^\n]*$/\1/' | sed 's/_bacteriophage.*$/_phage/' | sed 's/\(_phage\).*_phage.*$/\1/' | sed 's/\([A-Za-z]\{2\}phage\)_.*$/\1/' | sed 's/_prophage.*$/_phage/' | sed 's/-/_/' | sed 's/\./_/' | sed 's/Streptomyce_phage/Streptomyces_phage/' | sed 's/Mycobacteriophage/Mycobacterium_phage/' | sed 's/Environmental_Halophage/Environmental_halophage/' | sed 's/Corynephage/Corynebacterium_phage/' | sed 's/Enterobacterial_phage/Enterobacteria_phage/' | sed 's/Enterobacterio_phage/Enterobacteria_phage/' > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/trimmed_format_for_perl_filtered_${file}
```

NOTES

Geoffrey Hannigan 14 Jan 2016

Perl scripts and supplementary information available at figshare.

Geoffrey Hannigan 14 Jan 2016

Here I am primarily changing the phage identifiers and am not changing much of the other taxonomic information. There were some duplicate that we noticed upon manual inspection, so I also fix those here.

Analysis

Step 31.

Use the uniprot orf contig taxonomy script for contig taxonomy assignment.

COMMAND

```
perl contig_id_by_orfs.pl ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/trimmed_format_for_perl_filtered_${file} ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/perl_taxonomy_results_${file}
```

Analysis

Step 32.

Add underscore to the end of each contig number in the file.

COMMAND

```
sed 's/\t/_\t/' ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/perl_t
```

```
axonomy_results_${file} > ./uniprot_taxonomy_using_orfs/virome_phage_blastx_formatted_for_t
axa/perl_taxonomy_results_underscore_${file}
done
```

Analysis

Step 33.

The contigs have now been annotated with taxonomic information, but this is only half of the analysis battle. Our goal is to get a relative abundance table with each column being a sample, each row being a contig (like an "OTU") which has taxonomic information, and the intersections have the relative abundance information (calculated as RPKM).

Analysis

Step 34.

Run bowtie2 of the negative cleaned samples against the contig reference database. First build bowtie reference of the contigs.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund
bowtie2-build -
f ./ray_contigs_from_total_cat_pairs/Contigs_no_block_with_names.fasta ./uniprot_contig_vir
ome_trembl_rel_abund/bowtie2_contig_build
```

Analysis

Step 35.

Map the sequences to the contigs.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/bowtie2_neg_cleaned_hits
run.bowtie2.against.contigs () {
    bowtie2 -x ./uniprot_contig_virome_trembl_rel_abund/bowtie2_contig_build -
f ./negative_clean_seqs/$1 -
S ./uniprot_contig_virome_trembl_rel_abund/bowtie2_neg_cleaned_hits/$1 -L 25 -N 1
}
export -f run.bowtie2.against.contigs
```

Analysis

Step 36.

Run as a subroutine because it can be run much quicker with multiple procs at a time.

cmd COMMAND

```
ls ./negative_clean_seqs/ | xargs -I {} --max-procs=128 bash run.bowtie2.against.contigs {}
```

Analysis

Step 37.

Rename the files to be .sam files.

cmd COMMAND

```
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/bowtie2_neg_cleaned_hits); do
    mv ./uniprot_contig_virome_trembl_rel_abund/bowtie2_neg_cleaned_hits/"${file}" ./unipro
t_contig_virome_trembl_rel_abund/bowtie2_neg_cleaned_hits/"${file/%.fa/.sam}"
done
```

Analysis

Step 38.

Calculate the hit abundances from bowtie2 using bowtie2 estimation perl script.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/abundance_from_sam
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/bowtie2_neg_cleaned_hits); do
    perl calculate_abundance_from_sam.pl ./uniprot_contig_virome_trembl_rel_abund/bowtie2_n
eg_cleaned_hits/${file} ./uniprot_contig_virome_trembl_rel_abund/abundance_from_sam/${file}
done
```

NOTES

Geoffrey Hannigan 02 Feb 2016

Perl script 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

Analysis

Step 39.

Rename the sam files to text files.

cmd COMMAND

```
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/abundance_from_sam); do
    mv ./uniprot_contig_virome_trembl_rel_abund/abundance_from_sam/"${file}" ./uniprot_cont
ig_virome_trembl_rel_abund/abundance_from_sam/"${file/%.sam/.txt}"
done
```

Analysis

Step 40.

Add in contig length information using awk and calculate RPKM.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/abundance_with_length
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/abundance_from_sam); do
    export SUM=$(awk '{ SUM += $2 } END { print SUM }' ./uniprot_contig_virome_trembl_rel_a
bund/abundance_from_sam/${file})
```

Analysis

Step 41.

Add an echo of the sum value to confirm that the sum is being calculated (read it in STDOUT).

cmd COMMAND

```
echo Sum is $SUM
    awk --
assign sum=$SUM 'FNR==NR { a[$1]=$2; next } $1 in a { print $1"\t"$2"\t"a[$1]"\t"$2*1000000
000/(a[$1]*sum) }' ./contig_stats/contig_length_without_greater_sign.txt ./uniprot_contig_v
irome_trembl_rel_abund/abundance_from_sam/${file} > ./uniprot_contig_virome_trembl_rel_abun
d/abundance_with_length/${file}
done
```

Analysis

Step 42.

Get columns of only the contig IDs and the normalized RPKM abundance counts and add sample name to top of column for when this is used in distance matrix calculations.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/abundance_RPKM_with_only_contig_id
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/abundance_with_length); do
    NAME=$(echo ${file} | sed 's/_R1\.txt//')
    echo Name is $NAME
    cut -
f 1,4 ./uniprot_contig_virome_trembl_rel_abund/abundance_with_length/${file} | sed "1 s/^/C
ontig_ID\t${NAME}\n/" > ./uniprot_contig_virome_trembl_rel_abund/abundance_RPKM_with_only_c
ontig_id/${file}
done
```

Analysis

Step 43.

Make master list of the contig numbers as a reference for merging the relative abundance matrix values.

cmd COMMAND

```
sed -
n 1~2p ./ray_contigs_from_total_cat_pairs/Contigs_no_block_with_names.fasta | sed s'/>//g'
```

```
| sed '1 s/^/Contig_ID\n/' > ./ray_contigs_from_total_cat_pairs/master_contig_list.txt
```

Analysis

Step 44.

Merge the sample hit files to the master contig list. After this runs, all of the resulting files should have the same number of lines because they were all merged with the same master list.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/abundance_RPKM_with_only_contig_i
d); do
    awk 'FNR==NR {a[$1]=$2;next}{ print $1"\t"a[$1] }' ./uniprot_contig_virome_trembl_rel_a
bund/abundance_RPKM_with_only_contig_id/${file} ./ray_contigs_from_total_cat_pairs/master_c
ontig_list.txt | sed '/[0-9]\t[0-9]/!s/$/0/' | sed '1 s/\t0//' | sed '1 s/0$//' > ./uniprot
_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list/${file}
done
```

Analysis

Step 45.

Get only the RPKM abundance values so that they can be merged to the master contig list and used for distance matrix calculations.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/abundance_RPKM_for_merge
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list)
; do
    cut -
f 2 ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list/$file > ./uni
prot_contig_virome_trembl_rel_abund/abundance_RPKM_for_merge/${file}
done
```

Analysis

Step 46.

Merge the abundances with the contig names.

cmd COMMAND

```
paste ./ray_contigs_from_total_cat_pairs/master_contig_list.txt ./uniprot_contig_virome_tre
mbl_rel_abund/abundance_RPKM_for_merge/* > ./uniprot_contig_virome_trembl_rel_abund/contig_
otu_table.txt
```

Analysis

Step 47.

Transpose the cat file.

cmd COMMAND

```
python transpose_tab_delim.py -
i ./uniprot_contig_virome_trembl_rel_abund/contig_otu_table.txt -
o ./uniprot_contig_virome_trembl_rel_abund/contig_otu_table_transposed.txt
```

Analysis

Step 48.

Remove underscores from the transposed files as this can interfere with downstream analyses.

cmd COMMAND

```
sed 's/_//g' ./uniprot_contig_virome_trembl_rel_abund/contig_otu_table_transposed.txt > ./u
niprot_contig_virome_trembl_rel_abund/contig_otu_table_transposed_formatted.txt
```

Analysis

Step 49.

This data from the section above (Bray-curtis_virome_analysis/contig_otu_table_transposed_formatted.txt) can be used in Script R3. As stated above, we can now use the relative abundance table to determine the taxonomic composition of our virome samples. This analysis is described below.

Analysis

Step 50.

The above can be used for taxonomy reference independent diversity, but we will also want to look at virus taxonomy relative abundance.

Analysis

Step 51.

First make a master contig to ID reference table that includes the ID and the corresponding contig number.

COMMAND

```
awk 'FNR==NR {a[$1]=$2"\t"$3"\t"$4"\t"$5"\t"$6"\t";next}{ print $1"\t"a[$1] }' ./uniprot_ta
xonomy_using_orfs/virome_phage_blastx_formatted_for_taxa/perl_taxonomy_results_underscore_b
lastx_trembl_glimmer_total_orfs.txt ./ray_contigs_from_total_cat_pairs/master_contig_list.t
xt | sed 's/_\t$/_\tNo_hit\tNo_hit\tNo_hit\tNo_hit\tNo_hit/' > ./uniprot_taxonomy_using_orf
s/contig_id_reference_table.tsv
```

Analysis

Step 52.

Make a directory for the taxonomy output.

COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/trembl_order_rel_abund
```

Analysis

Step 53.

Calculate the relative abundance of the viral order.

COMMAND

```
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list)
; do
    NAME=$(echo ${file} | sed 's/_R1\.txt//')
    awk 'FNR==NR { a[$1]=$2; next } $1 in a { print $2"\t"a[$1] }' ./uniprot_contig_virome_
trembl_rel_abund/RPKM_contig_count_on_master_list/${file} ./uniprot_taxonomy_using_orfs/con
tig_id_reference_table.tsv  | sed '1d' > ./uniprot_contig_virome_trembl_rel_abund/trembl_or
der_rel_abund/raw_${file}
    awk '{a[$1]+=$2}END{for(i in a) print i,a[i]}' ./uniprot_contig_virome_trembl_rel_abund
/trembl_order_rel_abund/raw_${file} | sed 's/ /\t/g' | sed "s/$/\t${NAME}/" > ./uniprot_con
tig_virome_trembl_rel_abund/trembl_order_rel_abund/rel_abund_${file}
done
```

Analysis

Step 54.

Calculate the relative abundance of the viral families.

COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/trembl_family_rel_abund
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list)
; do
    NAME=$(echo ${file} | sed 's/_R1\.txt//')
    awk 'FNR==NR { a[$1]=$2; next } $1 in a { print $3"\t"a[$1] }' ./uniprot_contig_virome_
trembl_rel_abund/RPKM_contig_count_on_master_list/${file} ./uniprot_taxonomy_using_orfs/con
tig_id_reference_table.tsv  | sed '1d' > ./uniprot_contig_virome_trembl_rel_abund/trembl_fa
mily_rel_abund/raw_${file}
    awk '{a[$1]+=$2}END{for(i in a) print i,a[i]}' ./uniprot_contig_virome_trembl_rel_abund
/trembl_family_rel_abund/raw_${file} | sed 's/ /\t/g' | sed "s/$/\t${NAME}/" > ./uniprot_co
ntig_virome_trembl_rel_abund/trembl_family_rel_abund/rel_abund_${file}
done
```

Analysis

Step 55.

Calculate the relative abundance of the viral sub-families.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/trembl_sub_family_rel_abund
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list)
; do
    NAME=$(echo ${file} | sed 's/_R1\.txt//')
    awk 'FNR==NR { a[$1]=$2; next } $1 in a { print $4"\t"a[$1] }' ./uniprot_contig_virome_
trembl_rel_abund/RPKM_contig_count_on_master_list/${file} ./uniprot_taxonomy_using_orfs/con
tig_id_reference_table.tsv  | sed '1d' > ./uniprot_contig_virome_trembl_rel_abund/trembl_su
b_family_rel_abund/raw_${file}
    awk '{a[$1]+=$2}END{for(i in a) print i,a[i]}' ./uniprot_contig_virome_trembl_rel_abund
/trembl_sub_family_rel_abund/raw_${file} | sed 's/ /\t/g' | sed "s/$/\t${NAME}/" > ./unipro
t_contig_virome_trembl_rel_abund/trembl_sub_family_rel_abund/rel_abund_${file}
done
```

Analysis

Step 56.

Calculate the relative abundance of the viral genera.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/trembl_genus_rel_abund
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list)
; do
    NAME=$(echo ${file} | sed 's/_R1\.txt//')
    awk 'FNR==NR { a[$1]=$2; next } $1 in a { print $5"\t"a[$1] }' ./uniprot_contig_virome_
trembl_rel_abund/RPKM_contig_count_on_master_list/${file} ./uniprot_taxonomy_using_orfs/con
tig_id_reference_table.tsv  | sed '1d' > ./uniprot_contig_virome_trembl_rel_abund/trembl_ge
nus_rel_abund/raw_${file}
    awk '{a[$1]+=$2}END{for(i in a) print i,a[i]}' ./uniprot_contig_virome_trembl_rel_abund
/trembl_genus_rel_abund/raw_${file} | sed 's/ /\t/g' | sed "s/$/\t${NAME}/" > ./uniprot_con
tig_virome_trembl_rel_abund/trembl_genus_rel_abund/rel_abund_${file}
done
```

Analysis

Step 57.

Calculate the relative abundance of the viral species.

cmd COMMAND

```
mkdir ./uniprot_contig_virome_trembl_rel_abund/trembl_species_rel_abund
for file in $(ls ./uniprot_contig_virome_trembl_rel_abund/RPKM_contig_count_on_master_list)
; do
    NAME=$(echo ${file} | sed 's/_R1\.txt//')
    awk 'FNR==NR { a[$1]=$2; next } $1 in a { print $6"\t"a[$1] }' ./uniprot_contig_virome_
trembl_rel_abund/RPKM_contig_count_on_master_list/${file} ./uniprot_taxonomy_using_orfs/con
tig_id_reference_table.tsv  | sed '1d' > ./uniprot_contig_virome_trembl_rel_abund/trembl_sp
ecies_rel_abund/raw_${file}
    awk '{a[$1]+=$2}END{for(i in a) print i,a[i]}' ./uniprot_contig_virome_trembl_rel_abund
/trembl_species_rel_abund/raw_${file} | sed 's/ /\t/g' | sed "s/$/\t${NAME}/" > ./uniprot_c
ontig_virome_trembl_rel_abund/trembl_species_rel_abund/rel_abund_${file}
done
```

Analysis

Step 58.

Cat together all of the sample files by taxonomic level.

cmd COMMAND

```
cat ./uniprot_contig_virome_trembl_rel_abund/trembl_order_rel_abund/rel_abund_* > ./uniprot
_contig_virome_trembl_rel_abund/order_rel_abund.tsv
cat ./uniprot_contig_virome_trembl_rel_abund/trembl_family_rel_abund/rel_abund_* > ./unipro
t_contig_virome_trembl_rel_abund/family_rel_abund.tsv
cat ./uniprot_contig_virome_trembl_rel_abund/trembl_sub_family_rel_abund/rel_abund_* > ./un
iprot_contig_virome_trembl_rel_abund/sub_family_rel_abund.tsv
cat ./uniprot_contig_virome_trembl_rel_abund/trembl_genus_rel_abund/rel_abund_* > ./uniprot
```

```
_contig_virome_trembl_rel_abund/genus_rel_abund.tsv
cat ./uniprot_contig_virome_trembl_rel_abund/trembl_species_rel_abund/rel_abund_* > ./unipr
ot_contig_virome_trembl_rel_abund/species_rel_abund.tsv
```

NOTES

Geoffrey Hannigan 02 Feb 2016

These files can be used for taxonomic analysis in R.

Analysis

Step 59.

After the environmental background removal, we also wanted to use the program MEGAN to see what percent of our un-assebmled reads were unknown compared to the NCBI non-redundant database (meaning they had no hits to the database). The following outlines our BLAST approach.

cmd COMMAND

```
echo Subsampling clean fasta files for MEGAN analysis...
mkdir ./neg_clean_seqs_blastn_for_MEGAN
mkdir ./neg_clean_subsample_blastn_for_MEGAN
```

Analysis

Step 60.

These need to be subsampled mainly for speed purposes, but also for normalization.

cmd COMMAND

```
for file in $(ls ./negative_clean_seqs); do
        seqtk sample ./negative_clean_seqs/$file 2500 > ./neg_clean_subsample_blastn_for_ME
GAN/${file}
done
```

Analysis

Step 61.

Rename the files.

cmd COMMAND

```
echo Renaming the clean files for MEGAN...
for i in $(ls ./neg_clean_subsample_blastn_for_MEGAN); do
        mv ./neg_clean_subsample_blastn_for_MEGAN/"${i}" ./neg_clean_subsample_blastn_for_M
EGAN/"${i/%.*/.fa}"
done
echo BlastNing clean subsampled files for MEGAN analysis...
run.blast.parallel.clean.for.MEGAN () {
    blastn -query ./neg_clean_subsample_blastn_for_MEGAN/${1} -
out ./neg_clean_seqs_blastn_for_MEGAN/${1} -db ./references/ncbi/nt -outfmt 5 -
num_threads 2 -evalue 1e-3
}
export -f run.blast.parallel.clean.for.MEGAN
ls ./neg_clean_subsample_blastn_for_MEGAN/ | xargs -I {} --max-procs=128 sh -
c 'run.blast.parallel.clean.for.MEGAN'
wait
echo Renaming blastn output for MEGAN...
for i in $(ls ./neg_clean_seqs_blastn_for_MEGAN); do
        mv ./neg_clean_seqs_blastn_for_MEGAN/"${i}" ./neg_clean_seqs_blastn_for_MEGAN/"${i/
%.fa/.xml}"
done
```

Analysis

Step 62.

From here we used the MEGAN GUI to determine the ration of unknown reads. It is important to note that we manually only chose all of the samples except the backgroup control samples.

Analysis

Step 63.

Data was imported using the 'Import From Blast' option. The unknown read counts were taken from the following categories: Not Assigned, No Hits, Low Complexity.

Analysis

Step 64.

This is the only time we used MEGAN for virome analysis. In order to calculate unknown reads for the Whole Metagenome samples, we performed a similar analysis, blasting the decontaminated, trimmed reads subsampled at 2500 against the nt database with an e-value of 1e-3 and importing the blast files into MEGAN.