Searching further on viral matching HMM profiles

Studying further what matched the fOTUs for purposes of trying to validate the results

LJM

2019-12-31

Contents

1 Introduction

It would be nice to know further what viruses contributed to HMMs that matched to the fOTUs. Fortunately EggNOG database v 5.0 (Huerta-Cepas et al. 2019) provides that information. Let's download the relevant file from EggNOG database and taxonomic data from NCBI and create one big tsv file with all that information. Thereafter, let's check which HMMs matched significantly (with $-log_{10}$(e-value) of 2 or higher) to which fOTUs.

1.1 Load libraries

```
library(tidyverse)
```

2 Download from eggNOG database

2.1 Download members information on all viral HMMs

```
OUTPUT="../analyses/HMMsearch/validation_and_further_analyses/10239_members.tsv.gz"
wget --verbose --continue --output-document="$OUTPUT" \
"http://eggnog5.embl.de/download/eggnog_5.0/per_tax_level/10239/10239_members.tsv.gz"
gunzip "$OUTPUT"
```

2.2 Download annotation information on all viral HMMs

```
OUTPUT="../analyses/HMMsearch/validation_and_further_analyses/10239_annotations.tsv.gz"
wget --verbose --continue --output-document="$OUTPUT" \
"http://eggnog5.embl.de/download/eggnog_5.0/per_tax_level/10239/10239_annotations.tsv.gz"
gunzip "$OUTPUT"
```

3 Merge annotations and members files

```
INPUT1="../analyses/HMMsearch/validation_and_further_analyses/10239_annotations.tsv"
INPUT2="../analyses/HMMsearch/validation_and_further_analyses/10239_members.tsv"
SCRIPT="../scripts/merge_files.awk"
OUT="../analyses/HMMsearch/validation_and_further_analyses/10239_members_annotations.tsv"

awk -f $SCRIPT $INPUT1 $INPUT2 > $OUT
```

Now let's take a short look at the output

```
IN="../analyses/HMMsearch/validation_and_further_analyses/10239_members_annotations.tsv"
head -n 3 $IN
```

```
## taxid    HMM eggNOG_cat  mem_taxid.pfam_id   mem_taxid   description
## 10239    4QAIH   S   1229753.K7QJT8_9CAUD,948870.I7HXC4_9CAUD    1229753,948870  NA
## 10239    4QAII   S   1337877.R9VYA7_9CAUD,1589751.A0A0C5AES7_9CAUD   1337877,1589751 NA
```

Looks pretty ok.

Now we need to fetch taxonomic information from NCBI based on mem_taxid column information and append it to this previous data.

4 Fetch information from NCBI on each HMM

Let's use Biopython v. 1.75 (Cock et al. 2009) to access NCBI's taxonomy database for retrieving interesting taxonomic information about the viruses which contributed their proteins in the HMMs used. Biopython comes also with a handy parser of the XML format data that is returned from NCBI on each request.

4.1 Fetch taxonomic data

```
IN="../analyses/HMMsearch/validation_and_further_analyses/10239_members_annotations.tsv"
SCRIPT="../scripts/fetch_taxon_data.py"
AWK_PARSER="../scripts/tabinator.awk"
OUTPUT="../analyses/HMMsearch/validation_and_further_analyses/lineages.tsv"
# Retain fields of interest with AWK, process with python by augmenting the data with
# taxon info from NCBI and print out in a nicely machine readable format with AWK
tail -n+2 $IN | awk -F"\t" 'BEGIN{OFS=","}{print $2,$5}' |\
python3 $SCRIPT | awk -F"@" -f $AWK_PARSER > $OUTPUT
```

In this returned file `/analyses/HMMsearch/validation_and_further_analyses/lineages.tsv` there are some issues that should be fixed before it will be read with R and processed further. The issues are:

1. There are unnecessary spaces after `;` in lineages column, e.g. `Viruses; Caudovirales; Myoviridae; unclassified Myoviridae,Viruses; Caudovirales; Myoviridae; unclassified Myoviridae`.
2. For 350 records root taxon was downloaded as the first taxon. This is of no interest for us so we need to get rid of it. This task involves not only removing `root` from scientific names (2nd field) but all NAs in other taxonomic columns and leading commas in lineages fields (the 13th one).
3. For 350 other records generic Bacteria taxon was downloaded as the first taxon. This seems to be some type of incorrect relic and not a valid contributor organism for the HMMs. This task involves not only removing `Bacteria` from scientific names (2nd field) but all NAs in other taxonomic columns except `no_rank` (9th field) where `cellular organisms` should be removed and finally in lineages field (the 13th one) removing the leading `cellular organisms,` text.

Let's fix these issues in this order before we move on.

4.2 Remove spaces and initial `,` from lineages field

```
IN="../analyses/HMMsearch/validation_and_further_analyses/lineages.tsv"
OUT="../analyses/HMMsearch/validation_and_further_analyses/taxon_data.tsv"
AWK_SCRIPT="../scripts/fixLineages.awk"
awk -F"\t" -f $AWK_SCRIPT $IN > $OUT
#sed -n '3'p $OUT
#tail -n 100 $TEMP | awk -F"\t" -f $AWK_SCRIPT | awk -F"\t" '{print $13}'
```

4.3 Remove `root` taxons from some records

```
export AWKPATH="../scripts/awk_lib"
FILE="../analyses/HMMsearch/validation_and_further_analyses/taxon_data.tsv"
TEMP="../analyses/HMMsearch/validation_and_further_analyses/temp.tsv"
AWK_SCRIPT="../scripts/removeRoots.awk"
cp $FILE $TEMP
awk -F"\t" -f $AWK_SCRIPT $TEMP > $FILE
rm $TEMP
```

4.4 Remove `Bacteria` taxons from some records

```
FILE="../analyses/HMMsearch/validation_and_further_analyses/taxon_data.tsv"
TEMP="../analyses/HMMsearch/validation_and_further_analyses/temp.tsv"
AWK_SCRIPT="../scripts/removeBacteria.awk"
cp $FILE $TEMP
awk -F"\t" -f $AWK_SCRIPT $TEMP > $FILE
#sed -n '19,22'p $TEMP | awk -F"\t" -f $AWK_SCRIPT
rm $TEMP
```

4.5 The output file

The retrieved data looks now the following: (the lines have been stripped so it looks nicer in `pdf`/`html`-formats)

```
FILE="../analyses/HMMsearch/validation_and_further_analyses/taxon_data.tsv"
head -n 5 $FILE | cut -c-83
```

```
## 4QAIH    Escherichia phage phAPEC8;Enterobacteria phage phi92    Viruses,Viruses Caudovir
## 4QAII    Paenibacillus phage phiIBB_Pl23;Bacteriophage Lily  Viruses,Viruses Caudoviral
## 4QAIJ    Synechococcus phage S-CRM01;Mycobacterium phage Shauna1;Mycobacterium phage S
## 4QAIK    Synechococcus phage S-CRM01;Mycobacterium virus BBPiebs31;Mycobacterium virus
## 4QAIM    Aeromonas phage CC2;Aeromonas virus 65  Viruses,Viruses Caudovirales,Caudovira
```

5 Augment existing data with taxonomic information

5.1 Read in the data

```
# The files where the data is
taxon_data_file <- "../analyses/HMMsearch/validation_and_further_analyses/taxon_data.tsv"
hmm_data_file <- "../analyses/HMMsearch/validation_and_further_analyses/10239_members_annotations.tsv"
fOTUs_data_file <- "../analyses/HMMsearch/max_fOTUsHMM.csv"

# read in the data
hmm_data <- read_tsv(file = hmm_data_file,
                     col_names = TRUE,
                     col_types = c("fcfccc"),
                     na = "NA")

taxon_data <- read_tsv(file = taxon_data_file,
                     col_names = c("HMM",
                                   "sci_names",
                                   "superkingdoms",
                                   "orders",
                                   "families",
                                   "subfamilies",
                                   "genuses",
                                   "species",
                                   "no_ranks",
                                   "phylums",
                                   "classes",
                                   "unclassified",
                                   "lineages"),
                     col_types = c("ccccccccccccc"),
                     na = "NA")

columns <- paste("c", strrep("d",8315), sep = "")
fOTUsHMM <- read_csv(file = fOTUs_data_file,
                     col_names = TRUE,
                     col_types = columns)
```

5.2 Join the data into one tibble

```
joined_data <- hmm_data %>%
  left_join(.,
            taxon_data,
            by = "HMM")
```

5.3 Create summary

What we'd like to do now is create a tsv file for each fOTU such that it contains all the significant matches with HMMs as rows and the relevant HMM data gathered above joined to the rows.

First let's define some functions

```
# This function returns FALSE for columns/vectors
# where all elements are NA
not_all_na <- function(x){
  !all(is.na(x))
}

# This function removes columns from the tibble where all the elements
# are NA and returns the reduced tibble
remove_na_cols <- function(this_tbl){
  filtered <- this_tbl %>%
    select_if(not_all_na)
}
```

Now we'll simply iterate over each row in the tibble with fOTUs and $-log_{10}$(e-val):s, then remove all the columns (HMMs) (in each row) which didn't have any matches and pivot the HMMs with some $-log_{10}$(e-val):s as rows, join all the previously gathered data to each HMM and finally write this data to a `tsv`-file with the current fOTU name as file name.

```
path <- "../analyses/HMMsearch/validation_and_further_analyses/fOTUwise_taxon_HMM_data/"
for (row in 1:nrow(fOTUsHMM)) {
  # Scrape the current fOTU name for use when writing files
  fOTU_name <- as.character(fOTUsHMM[row,][1])
  # Create the file name with path
  file_name <- paste(path,fOTU_name,".tsv",sep = "")
  fOTUsHMM[row,] %>%
    remove_na_cols() %>%
    select(-fOTU_name) %>%
    gather("HMM","e-value") %>%
    left_join(.,
              joined_data,
              by = "HMM") %>%
    write_tsv(.,
             path = file_name,
             append = FALSE,
             col_names = TRUE)
}
```

6 Session info

```r
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 9 (stretch)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib/libopenblasp-r0.2.19.so
##
## locale:
##  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C           LC_TIME=C.UTF-8
##  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8    LC_MESSAGES=C
##  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
## [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] forcats_0.4.0   stringr_1.4.0   dplyr_0.8.3     purrr_0.3.2
## [5] readr_1.3.1     tidyr_0.8.3     tibble_2.1.3    ggplot2_3.2.1
## [9] tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.2       cellranger_1.1.0 pillar_1.4.2     compiler_3.6.1
##  [5] tools_3.6.1      zeallot_0.1.0    digest_0.6.20    lubridate_1.7.4
##  [9] jsonlite_1.6     evaluate_0.14    nlme_3.1-140     gtable_0.3.0
## [13] lattice_0.20-38  pkgconfig_2.0.2  rlang_0.4.0      cli_1.1.0
## [17] rstudioapi_0.10  yaml_2.2.0       haven_2.1.1      xfun_0.9
## [21] withr_2.1.2      xml2_1.2.2       httr_1.4.1       knitr_1.24
## [25] vctrs_0.2.0      generics_0.0.2   hms_0.5.1        grid_3.6.1
## [29] tidyselect_0.2.5 glue_1.3.1       R6_2.4.0         readxl_1.3.1
## [33] rmarkdown_1.15   bookdown_0.13    modelr_0.1.5     magrittr_1.5
## [37] backports_1.1.4  scales_1.0.0     htmltools_0.3.6  rvest_0.3.4
## [41] assertthat_0.2.1 colorspace_1.4-1 stringi_1.4.3    lazyeval_0.2.2
## [45] munsell_0.5.0    broom_0.5.2      crayon_1.3.4
```

References

Cock, P. J. A., Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, et al. 2009. "Biopython: freely available Python tools for computational molecular biology and bioinformatics." *Bioinformatics* 25 (11): 1422–3. https://doi.org/10.1093/bioinformatics/btp163.

Huerta-Cepas, Jaime, Damian Szklarczyk, Davide Heller, Ana Hernández-Plaza, Sofia K Forslund, Helen Cook, Daniel R Mende, et al. 2019. "eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses." *Nucleic Acids Research* 47 (D1): D309–D314. https://doi.org/10.1093/nar/gky1085.