HMM search output analyses

LJM

2019-11-26

Contents

Chapter 1

Analysing hmm output data

```
library(tidyverse)

hit_percentage <- vector()

path_to_data <- "../analyses/HMM_scan_using_eggNOG_HMMs/hmmsearch_out_parsed/"

files <- list.files(path = path_to_data, pattern = "*.tsv", recursive=FALSE)

fOTUbinNums <- read_csv(file = "../analyses/numBinsfOTU.csv",
                        col_types = "ci",
                        col_names = c("fOTU_name",
                                      "num_bins"))
egg_nog_cats <- read_tsv(file = "../data/annotations/10239_annotations.tsv",
                         col_types = "fcfc",
                         col_names = c("taxid","hmm_profile_id",
                                       "egg_nog_category",
                                       "hmm_description")) %>%
  # Drop taxid because it's uninteresting
  select(.,-taxid)

# Read in some data
tsv_names <- c("fOTU_name",
               "hmm_profile_id",
               "inside_inclusion_threshold",
               "Target_Bin_id",
               "Target_Seq_id",
               "full_sequence_e-value",
               "full_sequence_score",
               "full_sequence_bias",
```

```
                "best_one_domain_e-value",
                "best_one_domain_score",
                "best_one_domain_bias",
                "exp",
                "N",
                "description")

for(fOTU_file in files) {

  path_file <- paste(path_to_data,fOTU_file, sep = "")

  fOTU <- read_tsv(file = path_file,
                   col_types = "cclffddddddddic",
                   col_names = tsv_names)

  # Check if there were empty bins
  fOTU_first_term <- fOTU %>%
    pull(fOTU_name)
  fOTU_first_term <- fOTU_first_term[1]
  # The empty bins had "dummy" as the first word in the first line
  if(fOTU_first_term == "dummy"){
    hit_percentage <- c(hit_percentage,0.0)
    next
  }

  # Make a big data table out of these three tibbles by left joining
  fOTU <- left_join(fOTU, egg_nog_cats, by = "hmm_profile_id") %>%
    left_join(., fOTUbinNums, by = "fOTU_name") %>%
    # Remove values that were outside inclusion threshold
    filter(.,inside_inclusion_threshold) %>%
    # Drop inside_inclusion_threshold now that it has done its duty
    select(.,-inside_inclusion_threshold)

  # Find how many hits there are to each bin
  numHits <- fOTU %>%
    count(.,Target_Bin_id) %>%
    rename(.,Num_hits = n)

  # Add num hits beside each bin
  fOTU <- left_join(fOTU, numHits, by = "Target_Bin_id")

  # Count how many unique hits to bins there are
  fOTU_vec_len <- fOTU %>%
    distinct(.,Target_Bin_id) %>%
    pull(Target_Bin_id) %>%
```

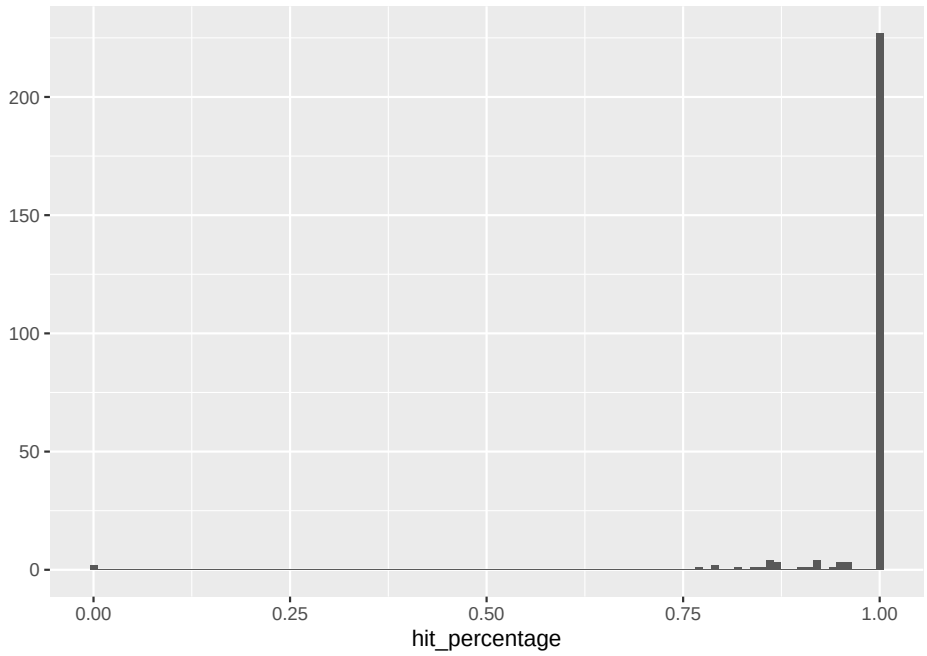
```
    length()

  # Grab the total number of bins in the fOTU
  fOTU_tot_num_bins <- fOTU %>%
    pull(num_bins)
  fOTU_tot_num_bins <- fOTU_vec_len/fOTU_tot_num_bins[1]

  # Append the value to a vector
  hit_percentage <- c(hit_percentage,fOTU_tot_num_bins)
}

# Create finally a tibble from the vector
hits <- tibble(hit_percentage)
```

Chapter 2

Visualise the data

The following graph depicts how large percentage of bins in fOTU have at least one hit from viral HMM profiles with e-value less than 0.01.

```
qplot(hit_percentage, data = hits, binwidth = 0.01)
```



Chapter 3

Session info

```
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 labeling_0.3     stringi_1.4.3
## [45] lazyeval_0.2.2   munsell_0.5.0    broom_0.5.2      crayon_1.3.4
```

Chapter 4

References