Most promising manual cluster

Looking more closely to see for what has been matched with HMMer and filtering previous results

LJM

2020-01-06

Contents

	A	B	C	D	E	F	G	
1	Cluster_name	Number_of_fOTUs	Total_length	Total_number_of_contigs	VirFinder_viral_content	PlasFlow_plasmid_content	GC_percentage	Tc
2	fOTU_PC1-and-2_x-+250-to-+400-and-y--100-to-0	118	2812146112	575896	Low	bacterial_chromosome	0.5535630476	
3	fOTU_PC1-and-2_x--100-to-0-and-y--100-to-+100	1187	216325975	30627	High	mixed	0.5149915908	
4	fOTU_PC2-and-3_x--100-to-+50-and-y--100-to-+100	1345	3580266181	694797	Low	bacterial_chromosome	0.5400976601	
5	fOTU_PC3-and-4_x--50-to-+50-and-y--50-to-+75	1228	2650819249	527049	Low	mixed	0.532501696	
6	fOTU_max_NMDS_x-+0_01-to-+0_02	4	219991	39	Low	mixed	0.4764740376	
7	fOTU_max_NMDS_x-+0_05-to-+9999	5	177797	47	Low	mixed	0.4857224813	
8	fOTU_max_NMDS_x--0_02-to-0	12	92667399	13672	Low	bacterial_chromosome	0.4650113574	
9	fOTU_max_NMDS_x--0_2-to--0.05	2	101128	12	Low	bacterial_chromosome	0.5871568705	
10	fOTU_max_NMDS_x--1-to--0_2	1	64539	12	Low	bacterial_chromosome	0.6087636933	
11	fOTU_max_NMDS_x--10-to--7	1	37228	8	Low	unclassified	0.5650316966	
12	fOTU_max_NMDS_x-0-to-+0_01	2770	7384890653	1496266	Low	bacterial_chromosome	0.5364252607	
13	fOTU_max_PC1-and-2_x-+200-to-+500-and-y--100-to-0	247	6234808593	1262644	Low	bacterial_chromosome	0.549887481	
14	fOTU_max_PC1-and-2_x--100-to-0-and-y--50-to-+50	2310	416931387	56807	High	mixed	0.5125171255	
15	fOTU_max_PC2-and-3_x-+100-to-+250-and-y--80-to-+125	50	89039608	15792	High	mixed	0.3463442135	
16	fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0	32	82431866	15232	Very_high	mixed	0.4182298142	
17	fOTU_max_PC2-and-3_x-+400-to-+650-and-y--210-to--80	18	23871254	3526	Very_high	bacterial_chromosome	0.3760687645	
18	fOTU_max_PC2-and-3_x--100-to-+75-and-y--150-to-+80	2640	7152309547	1453417	Low	bacterial_chromosome	0.5420871371	
19	fOTU_max_PC2-and-3_x-0-to-+200-and-y-+200-to-+450	37	72733390	12297	Medium	mixed	0.4128331431	
20	fOTU_max_PC3-and-4_x-+200-to-+450-and-y--25-to-+50	38	77042853	12953	Medium	mixed	0.4104644074	
21	fOTU_max_PC3-and-4_x--50-to-0-and-y-+75-to-+175	185	250756574	50745	Low	bacterial_chromosome	0.4901496381	
22	fOTU_max_PC3-and-4_x--80-to-+80-and-y--50-to-+75	2499	5937723174	1171809	Low	bacterial_chromosome	0.5413042496	

Figure 1: This figure shows some summary details of the putatively most promising viral cluster of the fOTUs. This will be studied further in this document.

1 Introduction

In this document will be studied further what features are present in manually clustered fOTU group `fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0` (row 16 in Figure 1). Further, e-value filtering (of 10^{-2}) and *Function unknown* of COG categories are filtered from these following clusters:

1. VirFinder viral content is low and PlasFlow result suggests bacterial chromosomal material (putatively predominantly bacterial chromosomal content).
2. VirFinder viral content is high or very high and PlasFlow result suggests bacterial chromosomal material or mixed (putatively predominantly viral content)

These estimation as based on previewing `summary_statistics.csv` file. The filtering of *Function unknown* of COG categories will help to see differences better between what is already there when it comes to viral material. 4

2 Features of interest that will be studied further

From the fOTU wise data files it might be interesting to see what is included in the matches with HMMs. These features are checked further in this document:

- COGs functional category
- Description of the COG (in eggNOG database)
- Taxonomic order
- Taxonomic family
- Taxonomic species
- Scientific name

2.1 Gather grouped data files

In this section new filtered (with respect to e-value and *Function unknown* in COG categories) data files are produced. In addition, `fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0` cluster will be studied separately. In order that the filtering will be implemented some adjustments to the AWK scripts used, need to be made.

```
OUT_DIR_BACT="../analyses/HMMsearch/validation_and_further_analyses/bacts_lns/"
OUT_DIR_VIR="../analyses/HMMsearch/validation_and_further_analyses/virs_lns/"
OUT="../analyses/HMMsearch/validation_and_further_analyses/cluster_data/"
SCRIPT_COGcat="../scripts/groupCOGcat.awk"
SCRIPT_Names="../scripts/groupNames.awk"
SCRIPT_Descs="../scripts/groupDescs.awk"
SCRIPT_Taxons="../scripts/groupTaxons.awk"
EVALUE="2.0"

awk -F"\t" -v e_val=$EVALUE -f "$SCRIPT_COGcat" "$OUT_DIR_BACT"*".tsv" \
>> "$OUT""bact_COGcats_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -v header="sci_names" -v field="8" \
-f "$SCRIPT_Names" "$OUT_DIR_BACT"*".tsv" \
>> "$OUT""bact_Sci_Names_filtered.txt"
awk -F"\t" -v e_val=$EVALUE -v header="species" -v field="14" \
-f "$SCRIPT_Names" "$OUT_DIR_BACT"*".tsv" \
>> "$OUT""bact_Spec_Names_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -f "$SCRIPT_Descs" "$OUT_DIR_BACT"*".tsv" \
>> "$OUT""bact_Descs_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -v header="orders" -v field="10" \
-f "$SCRIPT_Taxons" "$OUT_DIR_BACT"*".tsv" \
>> "$OUT""bact_Orders_filtered.txt"
awk -F"\t" -v e_val=$EVALUE -v header="families" -v field="11" \
-f "$SCRIPT_Taxons" "$OUT_DIR_BACT"*".tsv" \
>> "$OUT""bact_Families_filtered.txt"



awk -F"\t" -v e_val=$EVALUE -f "$SCRIPT_COGcat" "$OUT_DIR_VIR"*".tsv" \
>> "$OUT""vir_COGcats_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -v header="sci_names" -v field="8" \
-f "$SCRIPT_Names" "$OUT_DIR_VIR"*".tsv" \
```

```
>> "$OUT""vir_Sci_Names_filtered.txt"
awk -F"\t" -v e_val=$EVALUE -v header="species" -v field="14" \
-f "$SCRIPT_Names" "$OUT_DIR_VIR"*".tsv" \
>> "$OUT""vir_Spec_Names_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -f "$SCRIPT_Descs" "$OUT_DIR_VIR"*".tsv" \
>> "$OUT""vir_Descs_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -v header="orders" -v field="10" \
-f "$SCRIPT_Taxons" "$OUT_DIR_VIR"*".tsv" \
>> "$OUT""vir_Orders_filtered.txt"
awk -F"\t" -v e_val=$EVALUE -v header="families" -v field="11" \
-f "$SCRIPT_Taxons" "$OUT_DIR_VIR"*".tsv" \
>> "$OUT""vir_Families_filtered.txt"
```

Lastly let's run the previous groupings on `fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0` cluster but before that can be done let's create a directory with symbolic links to the relevant fOTUs.

```
VIR_LIST="../../lists_of_clusters/fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0.txt"
OUT_DIR_ROOT_VIR="../analyses/HMMsearch/validation_and_further_analyses/"
OUT_DIR_VIR="fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_lns/"
SCRIPT="../../../../scripts/groupSymLinkify.awk"
cd "$OUT_DIR_ROOT_VIR""$OUT_DIR_VIR"
awk -f "$SCRIPT" "$VIR_LIST"
```

And now the text files are created...

```
OUT_ROOT_DIR_VIR="../analyses/HMMsearch/validation_and_further_analyses/"
DIR="fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_lns/"
OUT="../analyses/HMMsearch/validation_and_further_analyses/cluster_data/"
SCRIPT_COGcat="../scripts/groupCOGcat.awk"
SCRIPT_Names="../scripts/groupNames.awk"
SCRIPT_Descs="../scripts/groupDescs.awk"
SCRIPT_Taxons="../scripts/groupTaxons.awk"
EVALUE="2.0"

awk -F"\t" -v e_val=$EVALUE -f "$SCRIPT_COGcat" "$OUT_ROOT_DIR_VIR""$DIR"*".tsv" \
>> "$OUT""fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_vir_COGcats_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -v header="sci_names" -v field="8" \
-f "$SCRIPT_Names" "$OUT_ROOT_DIR_VIR""$DIR"*".tsv" \
>> "$OUT""fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_vir_Sci_Names_filtered.txt"
awk -F"\t" -v e_val=$EVALUE -v header="species" -v field="14" \
-f "$SCRIPT_Names" "$OUT_ROOT_DIR_VIR""$DIR"*".tsv" \
>> "$OUT""fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_vir_Spec_Names_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -f "$SCRIPT_Descs" "$OUT_ROOT_DIR_VIR""$DIR"*".tsv" \
>> "$OUT""fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_vir_Descs_filtered.txt"

awk -F"\t" -v e_val=$EVALUE -v header="orders" -v field="10" \
-f "$SCRIPT_Taxons" "$OUT_ROOT_DIR_VIR""$DIR"*".tsv" \
>> "$OUT""fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_vir_Orders_filtered.txt"
awk -F"\t" -v e_val=$EVALUE -v header="families" -v field="11" \
-f "$SCRIPT_Taxons" "$OUT_ROOT_DIR_VIR""$DIR"*".tsv" \
>> "$OUT""fOTU_max_PC2-and-3_x-+250-to-+425-and-y--200-to-0_vir_Families_filtered.txt"
```

3 Check what is in the clusters

Now that we have these twelve text files. Let's read them in a form wordclouds from them.

```
#install.packages("wordcloud")
#install.packages("RColorBrewer")
#install.packages("tm")
library(wordcloud)
library(RColorBrewer)
library(tm)
library(tidyverse)
```

Let's define some functions. These functions will do the hard work of producing the visualisations.

```
de_underscore <- function(underscored) {
  de_underscored <- gsub("_",
                         " ",
                         underscored)
  de_underscored
  }

wordCloudify <- function(input_filename, output_filename){
  raw_text <- readLines(input_filename,
                        warn = F)
  docs <- Corpus(VectorSource(raw_text))
  # build a term-document matrix:
  tdm <- TermDocumentMatrix(docs)
  mat <- as.matrix(tdm)
  named_num <- sort(rowSums(mat),decreasing=TRUE)
  freq_data <- tibble(entity = names(named_num),
                      freq=named_num) %>%
    mutate_at(vars(entity),
              funs(de_underscore))

  #head(d, 10)

  # generate the wordcloud:
  png(filename = output_filename,
      width = 1000,
      height = 1000,)
  wordcloud(words = freq_data$entity,
            freq = freq_data$freq,
            min.freq = 1,
            max.words=200,
            random.order=FALSE,
            rot.per=0.35,
            colors=brewer.pal(8, "Dark2"))
  dev.off()
}
```

The input files are stored next.

```
dir_path <- "../analyses/HMMsearch/validation_and_further_analyses/cluster_data/"
files <- list.files(path = dir_path,
                    pattern = ".+_filtered.txt")
```

Lastly, let's create word clouds to visualise which entities are most common with respect to the aforementioned aspects.

```
set.seed(1234)
output_dir <- "../visualisations/HMMsearch/wordClouds/"
for (text_file in files) {
  in_file_name <- paste(dir_path,
                        text_file,
                        sep = "")
  extensionless <- gsub(".txt","",text_file)
  out_file_name <- paste(output_dir,
                         extensionless,
                         ".png",
                         sep = "")
  wordCloudify(in_file_name, out_file_name)
}
```