HMM search using eggNOG viral data and parsing of output

LJM

2019-12-22

Contents

1 Searching with HMMs 2

2 Counting number of bins in each fOTU 3

3 Parse delimited files from hmmsearch output files 4

4 Create a csv containing number of sequences per bin 5

5 Testing new utilities 6

5.1 Utility script for parsing eggNOG output files 6

5.2 Utility script for changing delimiters 6

5.3 Helper chunk for moving already studied fOTUs 7

6 Session info 8

7 References 9

Chapter 1

Searching with HMMs

Essentially the following code will be run in a sbatch script:

```
#ml bioinfo-tools hmmer/3.2.1
DATA="../analyses/HMMsearch/eggNOG_data/HMMs/10239/"
fOTUproteoms_DIR="../analyses/fOTU_proteomes/"
HMMer_OUTPUT_DIR="../analyses/HMMsearch/hmmsearch_out_viruses/"
TAXID_NO="10239"
read -r -a fOTUproteoms <<< $( find $fOTUproteoms_DIR -name "*.faa" -and -type f -print0 | x
echo "fOTUprot" ${#fOTUproteoms[@]}
# Take time of this run
SECONDS=0
read -r -a HMMs <<< $( find $DATA -name "*.hmm" -and -type f -print0 | xargs -0 echo )
for fOTUproteom in "${fOTUproteoms[@]}"; do
  fOTU_FILE=$(echo $(basename "$fOTUproteom"))
  fOTU=$(echo $(basename "$fOTUproteom") | awk -F "." '{print $1}')
  for HMM in "${HMMs[@]}"; do
    HMM_FILE=$(echo $(basename "$HMM"))
    HMM_VARIANT=$(echo $(basename "$HMM") | awk -F "." '{print $1}')
    hmmsearch --cpu 4 --noali --notextw -E 0.1 --domE 0.1 --incE 0.01 --incdomE 0.01 "$HMM"
  done
done
duration=$SECONDS
echo "$(($duration / 60)) minutes and $(($duration % 60)) seconds elapsed."
```

Chapter 2

Counting number of bins in each fOTU

```
fOTUs="../data/fOTUs.csv"
SCRIPT="../scripts/binCounter.awk"
OUTPUT="../analyses/numBinsfOTU.csv"
cat $fOTUs | awk -F, -v delimiter="," -f $SCRIPT > $OUTPUT
```

Chapter 3

Parse delimited files from hmmsearch output files

```
SCRIPT="../scripts/hmmsearchOutputParser.awk"
eggNOGs="../analyses/HMMsearch/hmmsearch_out_viruses/"
PARSED="../analyses/HMMsearch/hmmsearch_out_parsed/"
read -r -a eggNOGarray <<< $( find $eggNOGs -name "*.out" -and -type f -print0 | xargs -0 ec

SECONDS=0
for eggNOGfile in "${eggNOGarray[@]}"; do
  FILE=$(echo $(basename "$eggNOGfile"))
  fOTU=$(echo $FILE | awk -F "-with-" '{print $2}' | awk -F "." '{print $1}')
  awk -v filename="$FILE" -v delimiter="\t" -v last_col=10 -f "$SCRIPT" "$eggNOGfile" > "$PA
done

duration=$SECONDS
echo "$(($duration / 60)) minutes and $(($duration % 60)) seconds elapsed."
```

Chapter 4

Create a csv containing number of sequences per bin

```
BINS="../data/proteoms/"
OUTPUT="../analyses/numSeqsBin.csv"

read -r -a BINS_ARRAY <<< $( find $BINS -name "*.faa" -and -type f -print0 | xargs -0 echo )

# Here the Bin ID is named this way because it's faster to join this data to another later
printf "%s\n" "Target_Bin_id,Num_of_seqs" >> $OUTPUT

SECONDS=0
for BIN in "${BINS_ARRAY[@]}"; do
  FILE=$(echo $(basename "$BIN"))
  BIN_ID=$(echo $(basename "$FILE") | awk -F "." '{print $1}')
  NUM_SEQS=$(grep -c '>' $BIN)
  printf "%s," "$BIN_ID" >> $OUTPUT
  printf "%s\n" "$NUM_SEQS" >> $OUTPUT
done

duration=$SECONDS
echo "$(($duration / 60)) minutes and $(($duration % 60)) seconds elapsed."
```

Chapter 5

Testing new utilities

5.1 Utility script for parsing eggNOG output files

```
OUTPUT="../analyses/HMMsearch/hmmsearch_out_viruses/"
FILE="10239-with-cogOTU_1165.out"
SCRIPT="../scripts/hmmsearchOutputParser.awk"
#pcre2grep -M -B 19 -A 11 "^    \d.+$\n(.*\n)*?^Internal pipeline statistics summary:" $OUT
SECONDS=0
#pcre2grep -M -B 19 -A 11 "^    \d.+$\n(.*\n)*?^Internal pipeline statistics summary:" $OUT
awk -v filename="$FILE" -v delimiter="\t" -v last_col=10 -f $SCRIPT "$OUTPUT""$FILE"
duration=$SECONDS
echo "$(($duration / 60)) minutes and $(($duration % 60)) seconds elapsed."
```

5.2 Utility script for changing delimiters

The changing of delimiters is from space delimited to a user given delimiter

```
SCRIPT="../scripts/eggNOGdeliminator.awk"
TSV_FILE="Fuselloviridae_10474_annotations.tsv"
# A regex for the linus that should be skipped when adding delimiters in the middle
REGEXP="^#.+"

cat $TSV_FILE | awk -v delimiter=";" -v last_col="4" -v excluded_recs="$REGEXP" -f $SCRIPT
```

5.3 Helper chunk for moving already studied fOTUs

```
STUDIED_fOTUs="../analyses/HMMsearch/hmmsearch_out_viruses/"
fOTUs="../analyses/fOTU_proteomes1/"
TEMP_STORAGE="../analyses/fOTU_proteomes_studied/"

read -r -a STUDIED_fOTUs_ARRAY <<< $( find $STUDIED_fOTUs -name "*.out" -and -type f -print0
for STUDIED_fOTU in "${STUDIED_fOTUs_ARRAY[@]}"; do
  fOTU=$(echo $(basename "$STUDIED_fOTU") | awk -F "." '{print $1}' | awk -F "-with-" '{prin
  if [ -f "$fOTUs""$fOTU"".faa" ]; then
    mv "$fOTUs""$fOTU"".faa" "$TEMP_STORAGE"
  fi
done
```

Chapter 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     
## 
## loaded via a namespace (and not attached):
##  [1] compiler_3.6.1  magrittr_1.5    bookdown_0.13   tools_3.6.1    
##  [5] htmltools_0.3.6 yaml_2.2.0      Rcpp_1.0.2      stringi_1.4.3  
##  [9] rmarkdown_1.15  knitr_1.24      stringr_1.4.0   xfun_0.9       
## [13] digest_0.6.20   evaluate_0.14
```

Chapter 7

References