HMM search output analyses with avg e-val:s

LJM

2019-12-21

Contents

Chapter 1

Load libraries

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(vegan)
```

```
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.5-6
```

```
library(ggbiplot)
```

```
## Loading required package: plyr
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
```

```
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact

## Loading required package: scales

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

## Loading required package: grid
```

Chapter 2

Read intermediary `csv` file

In the intermediary `csv` file the $-log_{10}$(e-val):s are the average e-values of all matches with same HMM profiles to each fOTU.

```
# Create a character string which determines the column types
columns <- paste("c", strrep("d",8315), sep = "")
fOTUsHMM <- read_csv(file = "../analyses/HMMsearch/avg_fOTUsHMM.csv",
                     col_types = columns)
```

Chapter 3

Exchange NA, Inf and -Inf with 0

```
is.na(fOTUsHMM) <- sapply(fOTUsHMM, is.infinite)
fOTUsHMM[is.na(fOTUsHMM)] <- 0
```

Chapter 4

Move fOTU names to row names and execute PCA

```
fOTUsHMM_pca <- fOTUsHMM %>%
  column_to_rownames(var = "fOTU_name") %>%
  prcomp(.,
         center = TRUE)
```

Chapter 5

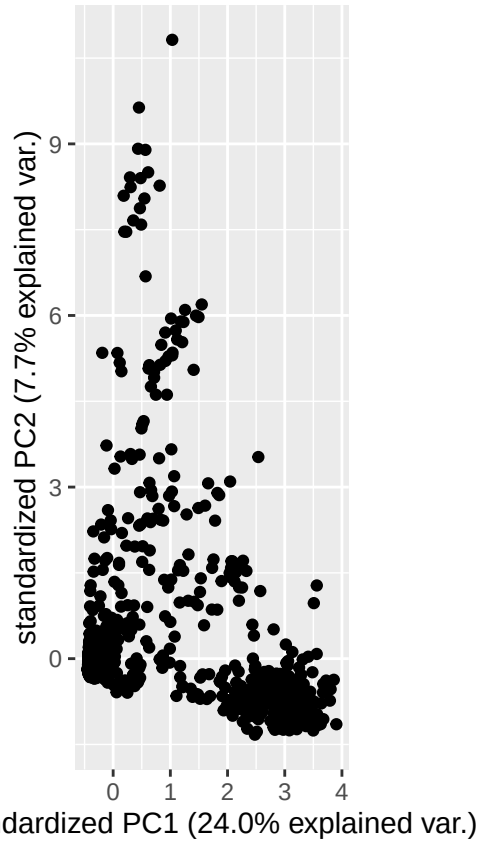
Visualise the results

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.

```
file_name <- "../visualisations/HMMsearch/ggbiplot_avg_PC1-2.png"
png(filename = file_name)
ggbiplot(fOTUsHMM_pca, var.axes = F, choices = 1:2)
dev.off()
```

```
## pdf
##   2
```

```
ggbiplot(fOTUsHMM_pca, var.axes = F, choices = 1:2)
```

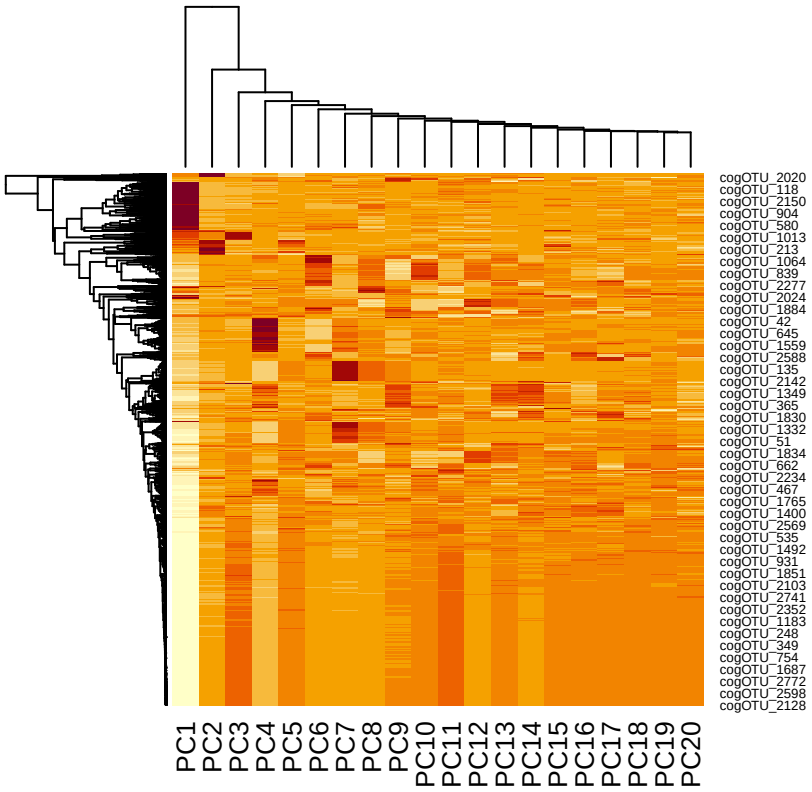


Here are some heatmaps to visualise how e-values are spread out by principal components and by HMMs.

```
prediction <- predict(fOTUsHMM_pca)[,1:20]
file_name <- "../visualisations/HMMsearch/heatmap_avg_20_first_PCs.png"
png(filename = file_name)
heatmap(as.matrix(prediction))
dev.off()
```

```
## pdf
##   2
```

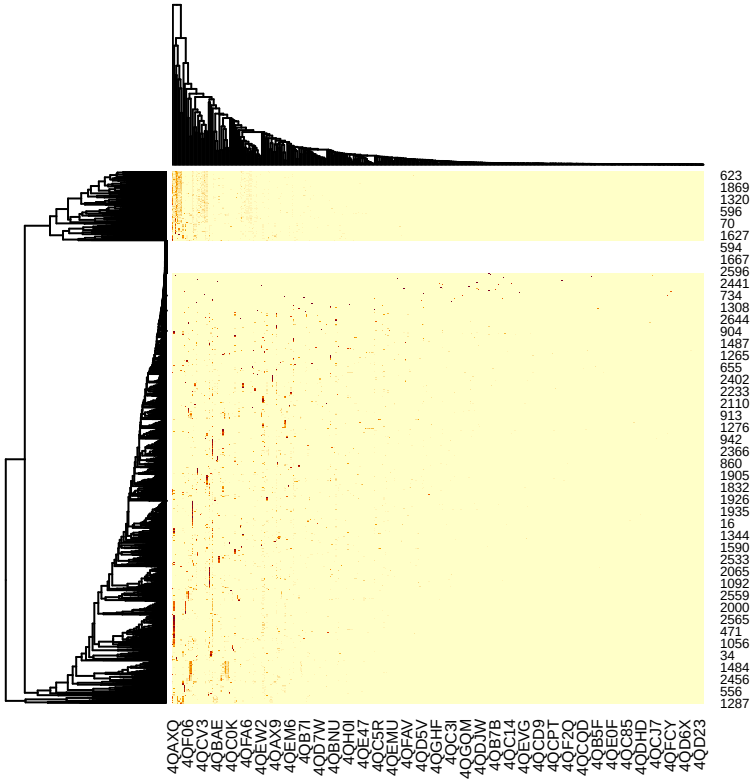
```
heatmap(as.matrix(prediction))
```




```
file_name <- "../visualisations/HMMsearch/heatmap_avg_400_first_fOTUs.png"
png(filename = file_name)
heatmap(as.matrix(fOTUsHMM[,-1])[,1:400])
dev.off()
```

```
## pdf
##   2
```

```
heatmap(as.matrix(fOTUsHMM[,-1])[,1:400])
```


623
1869
1320
596
70
1627
594
1667
2596
2441
734
1308
2644
904
1487
1265
655
2402
2233
2110
913
1276
942
2366
860
1905
1832
1926
1935
16
1344
1590
2533
2065
1092
2559
2000
2565
471
1056
34
1484
2456
556
1287
4QAXQ
4QF06
4QCV3
4QBAE
4QC0K
4QFA6
4QEW2
4QAX9
4QEM6
4QB7I
4QD7W
4QBNU
4QH0I
4QE47
4QC5R
4QEMU
4QFAV
4QD5V
4QGHF
4QC3I
4QGQM
4QDJW
4QB7B
4QC14
4QEVG
4QCD9
4QCPT
4QF2Q
4QCQD
4QB5F
4QE0F
4QC85
4QDHD
4QCCJ7
4QFCY
4QD6X
4QD23

Chapter 6

NMDS

Let's perform non-metric multidimensional scaling:

```
width <- dim(fOTUsHMM)[2]
ord_avg <- metaMDS(fOTUsHMM[,2:width])
```

Chapter 7

Save/Load R data

Save or load the data once again to speed-up things.

```
#saveRDS(ord_avg, file = "../analyses/HMMsearch/R_data/avg_ord.rds")
ord_avg <- readRDS(file = "../analyses/HMMsearch/R_data/avg_ord.rds")
```

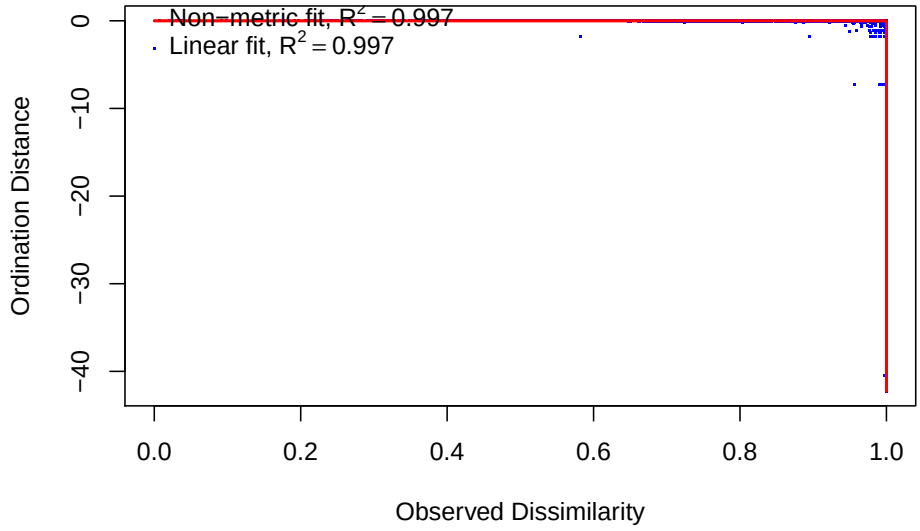
How well does the 2-D representation of multidimensional space represent the actual multidimensional space is done by using statistic called stress. Stress < 0.1 indicates that the 2-D representation is a good repreresentation of the multi-D space. Let's check that out now:

```
ord_avg$stress
```

```
## [1] 0.05112436
```

This is quite low which is great! Let's now plot out the relationship between the ordination dissimilarities and the distances in the ordination space:

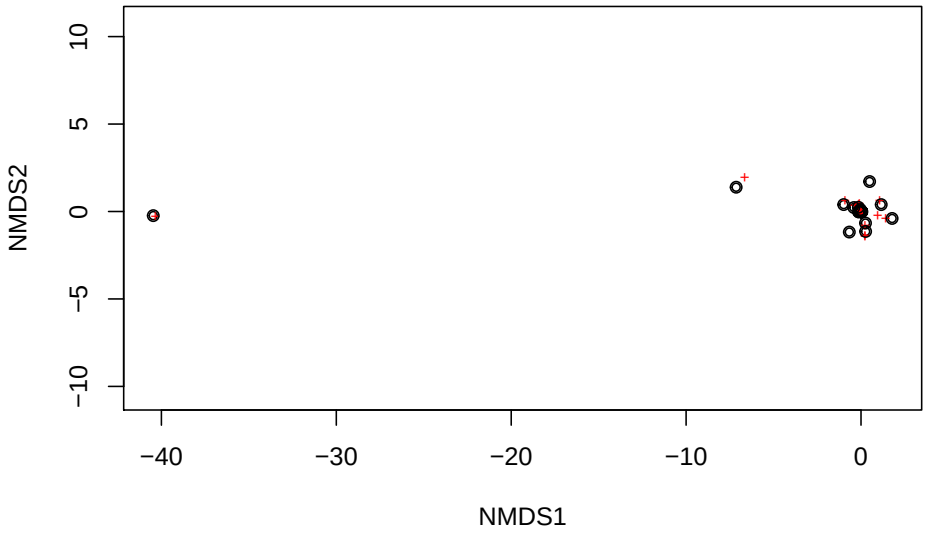
```
stressplot(ord_avg)
```



Observed dissimilarity of 1 means the fOTU doesn't have any HMMs incommon with the others.

Let's plot out the ordination space in 2-d:

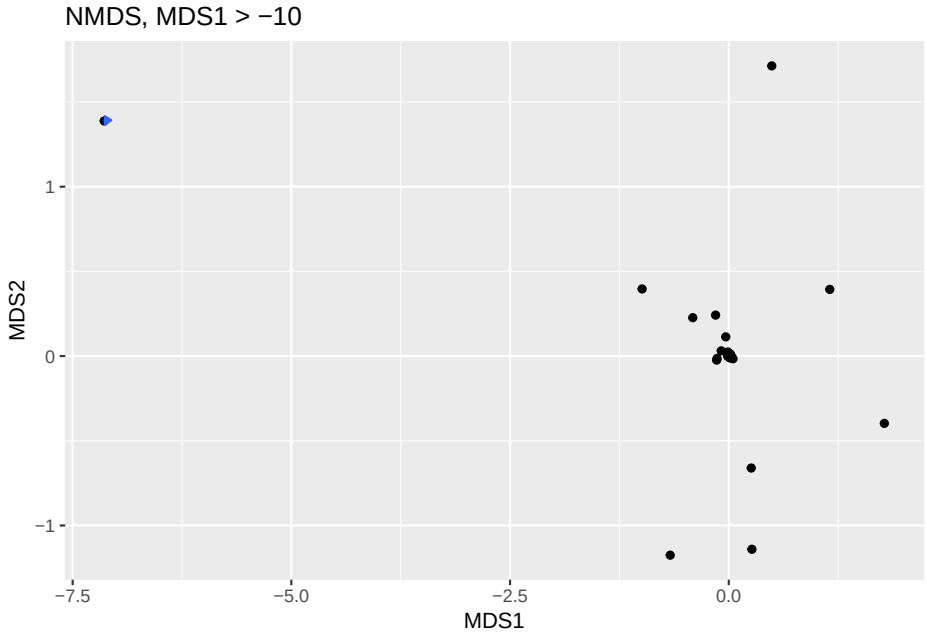
```
ordiplot(ord_avg)
points(ord_avg)
```



and let's filter away the outliers and make new plot from above:

```
NMDS_points1 <- as_tibble(ord_avg$points) %>%
  filter(MDS1 > -10)
```

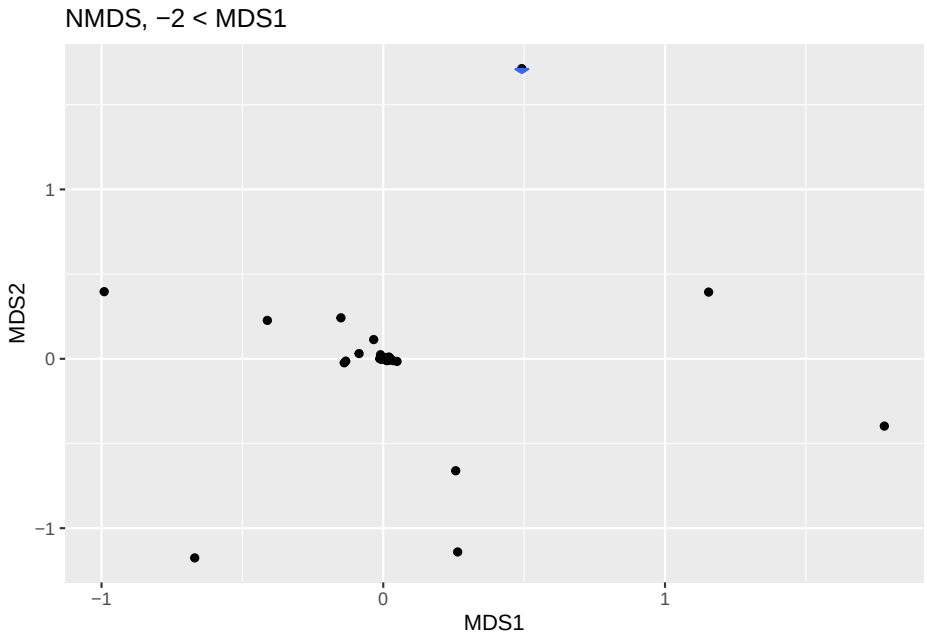
```
title <- "NMDS, MDS1 > -10"
qplot(MDS1,
      MDS2,
      data = NMDS_points1,
      geom = c("point", "density2d")) + ggtitle(title)
```




```
path <- "../visualisations/HMMsearch/nmds_hmmsearch_avg_MDS1_greater_than_-10.png"

ggsave(filename = path,
       device = png,
       width = 1000,
       height = 750,
       units = "in",
       limitsize = FALSE,
       dpi = 320)
```

Let's filter away slightly more:

```
NMDS_points2 <- as_tibble(ord_avg$points) %>%
  filter(MDS1 > -2)

title <- "NMDS, -2 < MDS1"
qplot(MDS1,
      MDS2,
      data = NMDS_points2,
      geom = c("point", "density2d")) + ggtitle(title)
```




```
path <- "../visualisations/HMMsearch/nmds_hmmsearch_avg_MDS1_greater_than_-2.png"

ggsave(filename = path,
       device = png,
       width = 1000,
       height = 750,
       units = "in",
       limitsize = FALSE,
       dpi = 320)
```

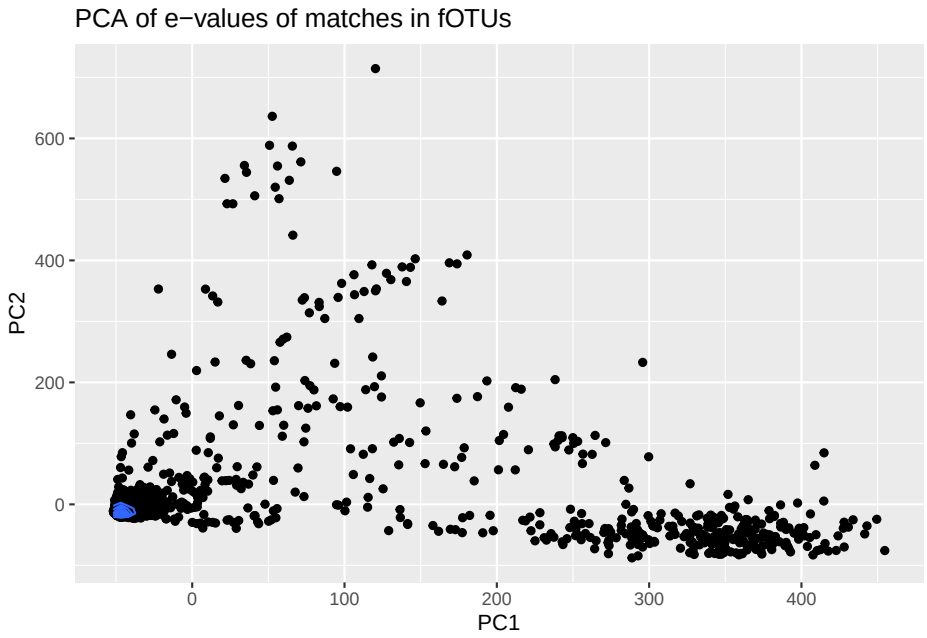
Chapter 8

Create lists of fOTUs of interest

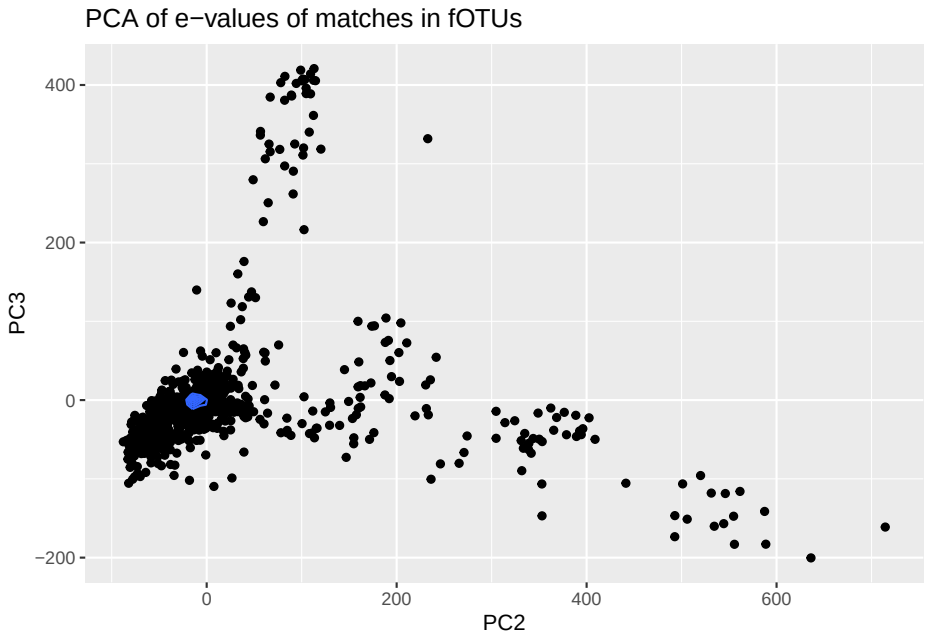
8.1 Preview PCA plots

These PCA plots can be used to visually determine potential groupings.

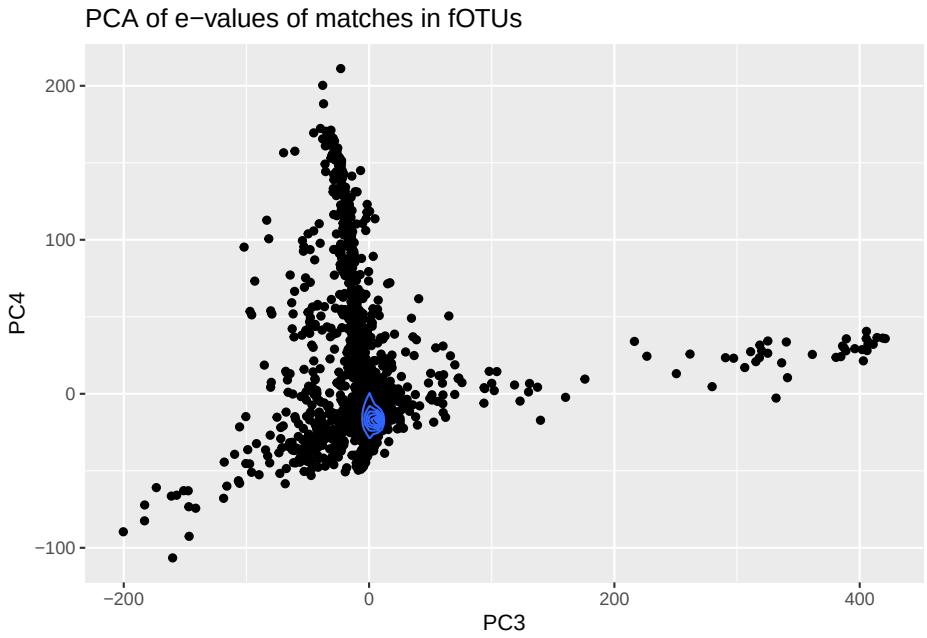
```
PC12s <- tibble(fOTUs = names(fOTUsHMM_pca$x[,1]),
                PC1 = unname(fOTUsHMM_pca$x[,1]),
                PC2 = unname(fOTUsHMM_pca$x[,2]))

title <- "PCA of e-values of matches in fOTUs"
qplot(PC1,
      PC2,
      data = PC12s,
      geom = c("point", "density2d")) + ggtitle(title)
```




```
ggsave(filename = "../visualisations/HMMsearch/pca_hmmsearch_avg_PC1-2.png",
       device = png,
       width = 1000,
       height = 750,
       units = "in",
       limitsize = FALSE,
       dpi = 320)

PC23s <- tibble(fOTUs = names(fOTUsHMM_pca$x[,1]),
                PC2 = unname(fOTUsHMM_pca$x[,2]),
                PC3 = unname(fOTUsHMM_pca$x[,3]))

title <- "PCA of e-values of matches in fOTUs"
qplot(PC2,
      PC3,
      data = PC23s,
      geom = c("point", "density2d")) + ggtitle(title)
```




```
ggsave(filename = "../visualisations/HMMsearch/pca_hmmsearch_avg_PC2-3.png",
       device = png,
       width = 1000,
       height = 750,
       units = "in",
       limitsize = FALSE,
       dpi = 320)

PC34s <- tibble(fOTUs = names(fOTUsHMM_pca$x[,1]),
                PC3 = unname(fOTUsHMM_pca$x[,3]),
                PC4 = unname(fOTUsHMM_pca$x[,4]))

title <- "PCA of e-values of matches in fOTUs"
qplot(PC3,
      PC4,
      data = PC34s,
      geom = c("point", "density2d")) + ggtitle(title)
```




```
ggsave(filename = "../visualisations/HMMsearch/pca_hmmsearch_avg_PC3-4.png",
       device = png,
       width = 1000,
       height = 750,
       units = "in",
       limitsize = FALSE,
       dpi = 320)
```

8.2 Create a file for each list

By comparing visually the PCA data, we can see that they are located essentially in same locations. Thus it is unnecessary to create new lists based on the average e-value points when there already are points based on the maximum e-values.

Chapter 9

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] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ggbiplot_0.55   scales_1.0.0    plyr_1.8.4      vegan_2.5-6
##  [5] lattice_0.20-38 permute_0.9-5   forcats_0.4.0   stringr_1.4.0
##  [9] dplyr_0.8.3     purrr_0.3.2     readr_1.3.1     tidyr_0.8.3
## [13] tibble_2.1.3    ggplot2_3.2.1   tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.5 xfun_0.9         splines_3.6.1    haven_2.1.1
##  [5] colorspace_1.4-1 generics_0.0.2   vctrs_0.2.0      htmltools_0.3.6
##  [9] mgcv_1.8-28      yaml_2.2.0       rlang_0.4.0      pillar_1.4.2
```

```
## [13] glue_1.3.1       withr_2.1.2      modelr_0.1.5     readxl_1.3.1
## [17] munsell_0.5.0    gtable_0.3.0     cellranger_1.1.0 rvest_0.3.4
## [21] evaluate_0.14    labeling_0.3     knitr_1.24       parallel_3.6.1
## [25] broom_0.5.2      Rcpp_1.0.2       backports_1.1.4  jsonlite_1.6
## [29] hms_0.5.1        digest_0.6.20    stringi_1.4.3    bookdown_0.13
## [33] cli_1.1.0        tools_3.6.1      magrittr_1.5     lazyeval_0.2.2
## [37] cluster_2.1.0    crayon_1.3.4     pkgconfig_2.0.2  zeallot_0.1.0
## [41] Matrix_1.2-17    MASS_7.3-51.4    xml2_1.2.2       lubridate_1.7.4
## [45] assertthat_0.2.1 rmarkdown_1.15   httr_1.4.1       rstudioapi_0.10
## [49] R6_2.4.0         nlme_3.1-140     compiler_3.6.1
```

Chapter 10

References