Find A Gene Project: POMC

Taylor Darby

11/12/2021

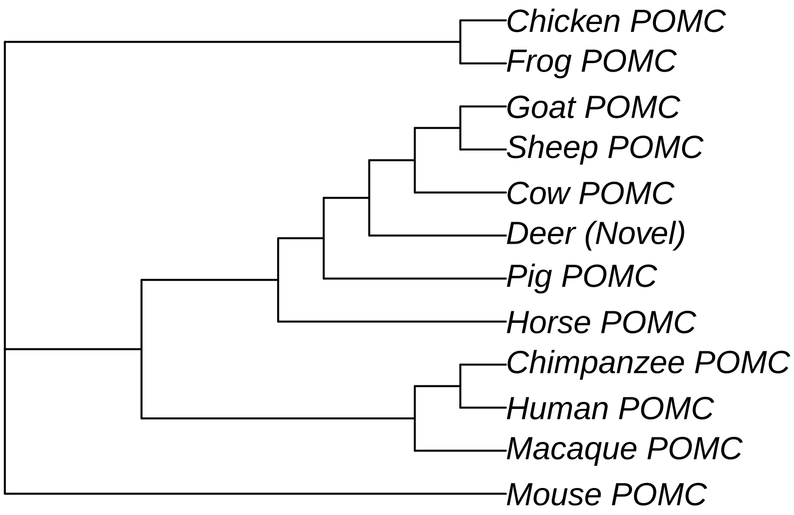
[Q6] Create a phylogenetic tree, using either a parsimony or distance-based approach. Bootstrapping and tree rooting are optional. Use "simple phylogeny" online from the EBI or any respected phylogeny program (such as MEGA, PAUP, or Phylip). Paste an image of your Cladogram or tree output in your report.

```r
#install.packages("remotes")
#remotes::install_github("GuangchuangYu/treeio")
#install.packages("seqinr")
#BiocManager::install("ggtree")
#BiocManager::install("DECIPHER")
#install.packages("ape")
#install.packages("adegenet")

#library(seqinr)
library(adegenet)
library(ape)
library(ggtree)
library(DECIPHER)
library(viridis)
library(ggplot2)
library(phangorn)
```

```r
read <- read.aa("POMC_aligned.fasta", format = "fasta")
read_phyDat <- phyDat(read, type = "AA")
read.dist <- dist.ml(read_phyDat, model="JC69")
POMC_UPGMA <- upgma(read.dist)
POMC_optim <- optim.parsimony(POMC_UPGMA, read_phyDat)
```

```
## Final p-score 325 after  3 nni operations
```

```r
plot(POMC_optim)
```



Other options:

```
#fit <- pml(POMC_UPGMA,read_phyDat)
#fitB <- optim.pml(fit, model = "Blosum62")
#fitBoot <- bootstrap.pml(fitB, bs=1000, optNni=TRUE, control = pml.control(trace=0))
#plotBS(midpoint(fitB$tree), fitBoot, type="p", cex = 0.5)
```

```
#seqs <- readAAStringSet("POMC_protein.fasta", format = "fasta")
#aligned <- AlignSeqs(seqs)
#writeXStringSet(aligned, file="POMC_aligned.fasta")
#tree.data <- read.alignment("POMC_aligned.fasta", format = "fasta")
#tree.dist <- dist.alignment(tree.data, matrix = "identity")
#temp.dist <- as.data.frame(as.matrix(tree.dist))
#table.paint(temp.dist, cleg=.75) + scale_color_viridis()
```

```
#nj.tree <- nj(tree.dist)
#nj.tree <- ladderize(nj.tree)
#new.nj <- nj.tree
#new.nj$tip.label <- aligned@ranges@NAMES
#msaplot(p=ggtree(new.nj, options(ignore.negative.edge=TRUE)), fasta="POMC_aligned.fasta", offset = 0.1
```

[Q7] Generate a sequence identity based heatmap of your aligned sequences using R. If necessary convert your sequence alignment to the ubiquitous FASTA format (Seaview can read in clustal format and “Save as” FASTA format for example). Read this FASTA format alignment into R with the help of functions in the Bio3D package. Calculate a sequence identity matrix (again using a function within the Bio3D package).

Then generate a heatmap plot and add to your report. Do make sure your labels are visible and not cut at the figure margins.

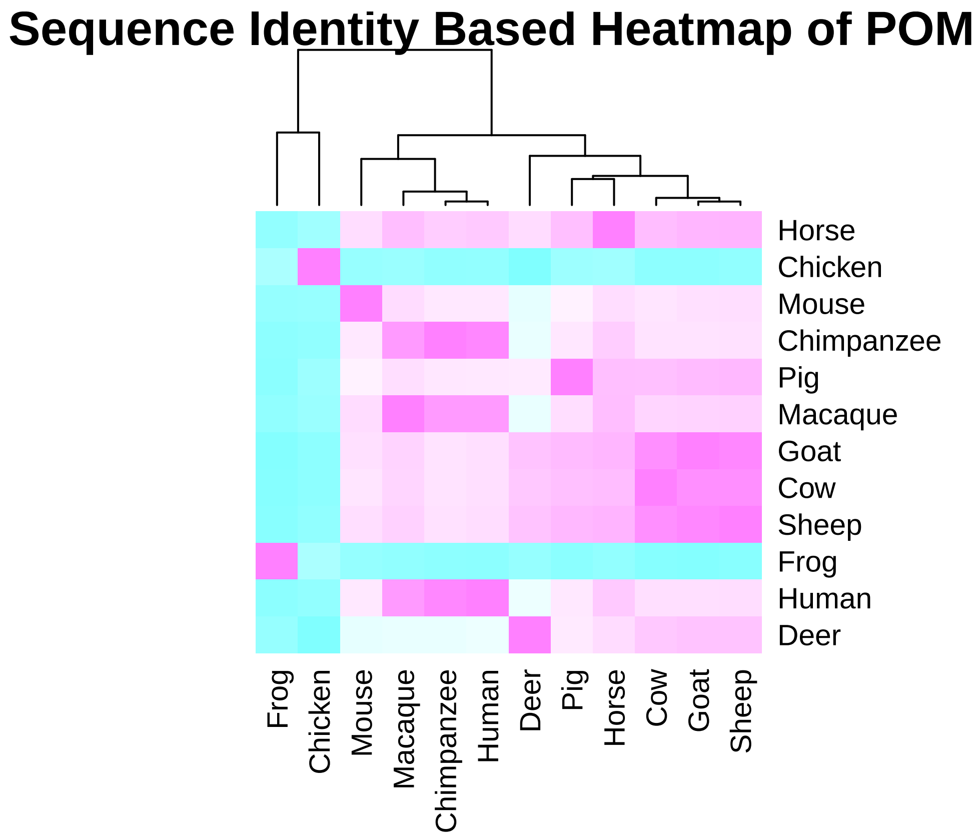
```r
# I will use the function 'seqaln()' which requires a MUSCLE download. Downloaded MUSCLE in the 'Termin
# I will also use the bio3d package which is already installed
library(bio3d)
```

```
##
## Attaching package: 'bio3d'

## The following object is masked from 'package:Biostrings':
##
##     mask

## The following object is masked from 'package:IRanges':
##
##     trim

## The following object is masked from 'package:ape':
##
##     consensus
```

```r
POMC_fasta <- read.fasta("POMC_protein.fasta")
POMC_align <- seqaln(POMC_fasta)
POMC_seqID <- seqidentity(POMC_align)
POMC_seqID
```

```
##             Deer Human  Frog Sheep   Cow  Goat Macaque   Pig Chimpanzee Mouse
## Deer       1.000 0.738 0.581 0.876 0.868 0.876   0.730 0.806      0.730 0.725
## Human      0.738 1.000 0.563 0.829 0.826 0.826   0.951 0.810      0.985 0.810
## Frog       0.581 0.563 1.000 0.556 0.552 0.548   0.572 0.560      0.565 0.579
## Sheep      0.876 0.829 0.556 1.000 0.970 0.985   0.852 0.897      0.822 0.828
## Cow        0.868 0.826 0.552 0.970 1.000 0.970   0.844 0.882      0.819 0.815
## Goat       0.876 0.826 0.548 0.985 0.970 1.000   0.848 0.890      0.819 0.824
## Macaque    0.730 0.951 0.572 0.852 0.844 0.848   1.000 0.828      0.951 0.832
## Pig        0.806 0.810 0.560 0.897 0.882 0.890   0.828 1.000      0.811 0.791
## Chimpanzee 0.730 0.985 0.565 0.822 0.819 0.819   0.951 0.811      1.000 0.810
## Mouse      0.725 0.810 0.579 0.828 0.815 0.824   0.832 0.791      0.810 1.000
## Chicken    0.538 0.574 0.619 0.572 0.564 0.564   0.588 0.593      0.571 0.583
## Horse      0.832 0.865 0.573 0.904 0.888 0.900   0.886 0.884      0.858 0.829
##            Chicken Horse
## Deer         0.538 0.832
## Human        0.574 0.865
## Frog         0.619 0.573
## Sheep        0.572 0.904
## Cow          0.564 0.888
## Goat         0.564 0.900
## Macaque      0.588 0.886
## Pig          0.593 0.884
## Chimpanzee   0.571 0.858
## Mouse        0.583 0.829
## Chicken      1.000 0.599
## Horse        0.599 1.000
```

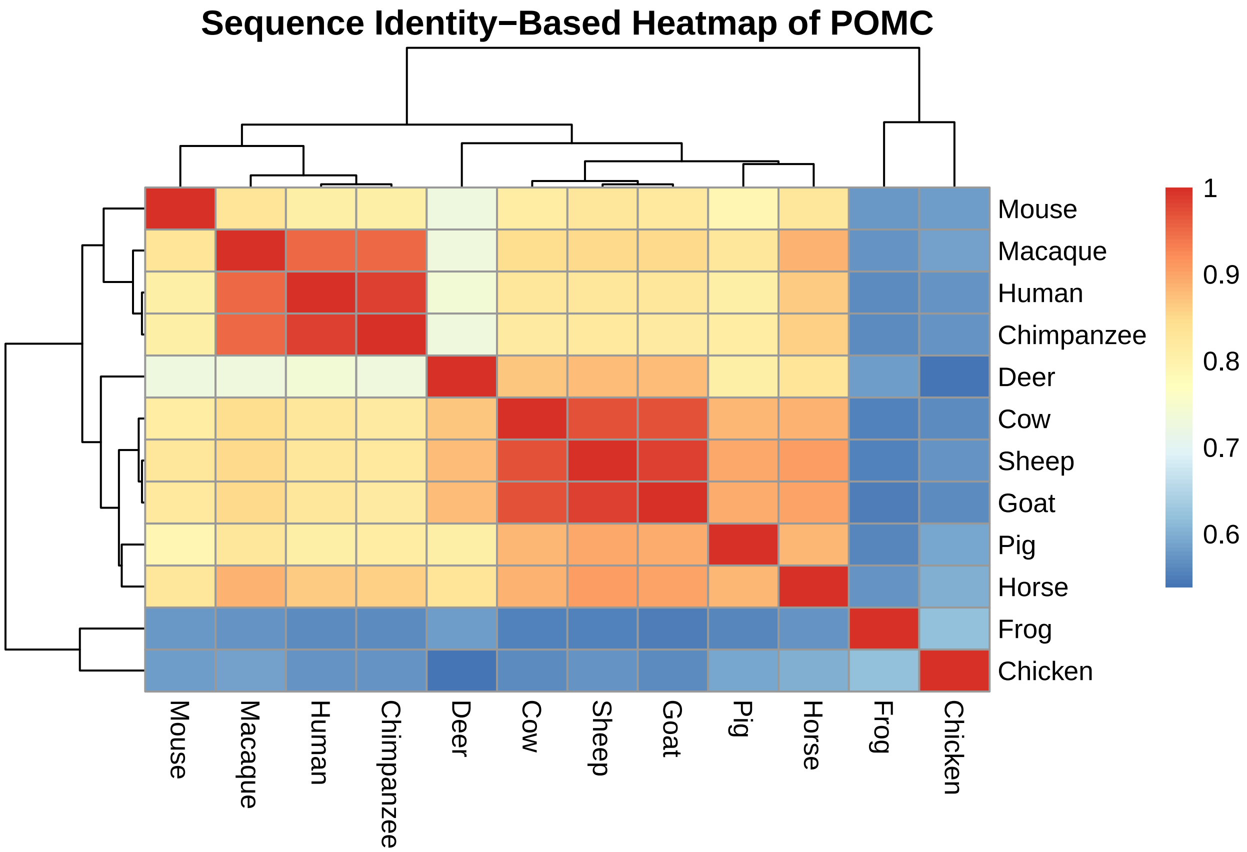
```
POMC_heatmap <- heatmap(POMC_seqID, Rowv= NA, col= cm.colors(256), scale = "none", main = "Sequence Ide
```



```
POMC_heatmap
```

```
## $rowInd
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
##
## $colInd
##  [1]  3 11 10  7  9  2  1  8 12  5  6  4
##
## $Rowv
## NULL
##
## $Colv
## NULL
```

```
# install.packages("pheatmap")
library(pheatmap)
pheatmap(POMC_seqID, scale = "none", main = "Sequence Identity-Based Heatmap of POMC")
```



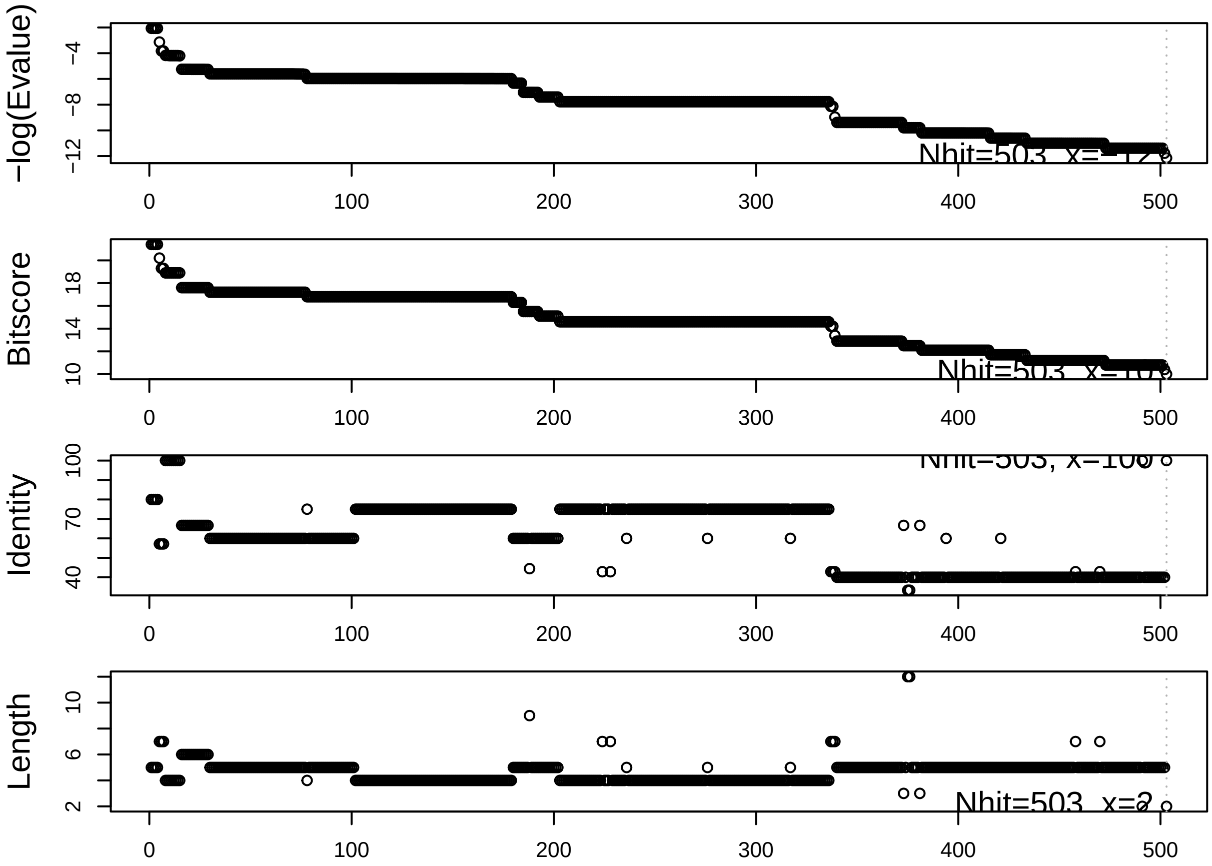
[**Q8**] Using R/Bio3D (or an online blast server if you prefer), search the main protein structure database for the most similar atomic resolution structures to your aligned sequences. List the top 3 unique hits (i.e. not hits representing different chains from the same structure) along with their Evalue and sequence identity to your query. Please also add annotation details of these structures. For example include the annotation terms PDB identifier (structureId), Method used to solve the structure (experimentalTechnique), resolution (resolution), and source organism (source). HINT: You can use a single sequence from your alignment or generate a consensus sequence from your alignment using the Bio3D function consensus(). The Bio3D functions blast.pdb(), plot.blast() and pdb.annotate() are likely to be of most relevance for completing this task. Note that the results of blast.pdb() contain the hits PDB identifier (or pdb.id) as well as Evalue and identity. The results of pdb.annotate() contain the other annotation terms noted above. Note that if your consensus sequence has lots of gap positions then it will be better to use an original sequence from the alignment for your search of the PDB. In this case you could chose the sequence with the highest identity to all others in your alignment by calculating the row-wise maximum from your sequence identity matrix.

```
#library(bio3d)
POMC_blast <- blast.pdb("POMC.pdb")
```

```
##  Searching ... please wait (updates every 5 seconds) RID = UFFGFXG2016
##  ..
##  Reporting 503 hits
```

```
POMC_table <- head(POMC_blast$hit.tbl, n=3)
hits <- plot.blast(POMC_blast, cutoff = -13)
```

```
##   * Possible cutoff values:    -13
##              Yielding Nhits:     503
##
##    * Chosen cutoff value of:     -13
##              Yielding Nhits:     503
```



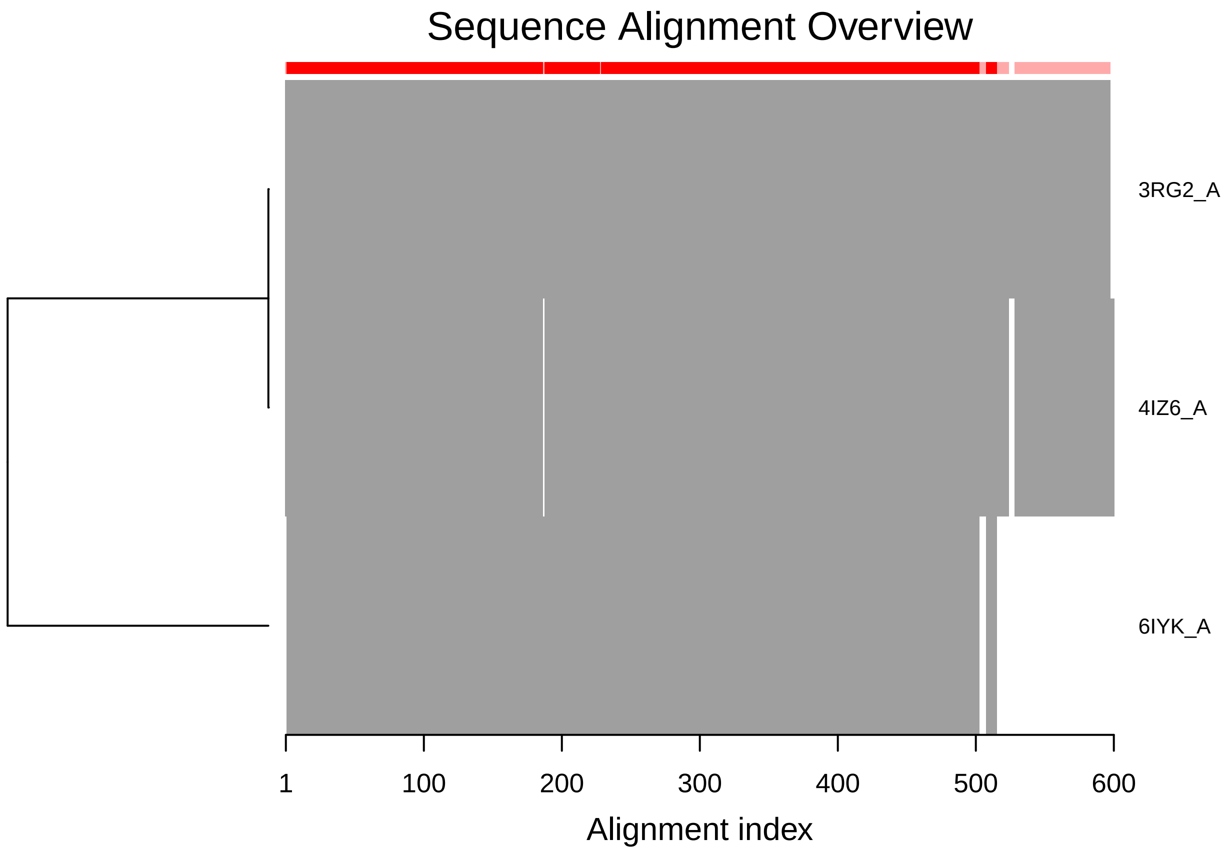
```r
write.csv(POMC_table, "POMC_top3_hits.csv")
```

```r
top3_ids <- hits$pdb.id[1:3]

# Download related pdb files
files <- get.pdb(top3_ids, path = "pdbs", split = TRUE, gzip = TRUE)
```

```
## Warning in get.pdb(top3_ids, path = "pdbs", split = TRUE, gzip = TRUE): pdbs/
## 4IZ6.pdb exists. Skipping download

## Warning in get.pdb(top3_ids, path = "pdbs", split = TRUE, gzip = TRUE): pdbs/
## 3RG2.pdb exists. Skipping download

## Warning in get.pdb(top3_ids, path = "pdbs", split = TRUE, gzip = TRUE): pdbs/
## 6IYK.pdb exists. Skipping download

##   |                                                                               |
# Align related PDBs
pdbs <- pdbaln(files, fit = TRUE)
```

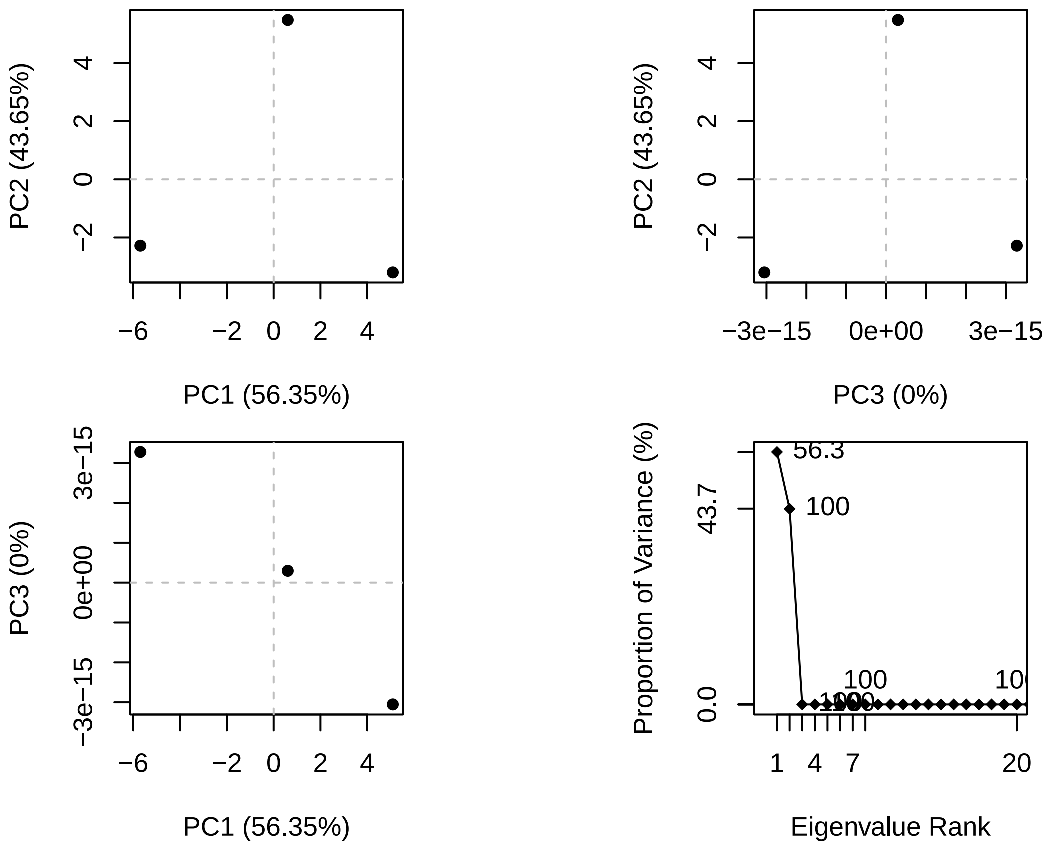
```
## Reading PDB files:
## pdbs/split_chain/4IZ6_A.pdb
## pdbs/split_chain/3RG2_A.pdb
## pdbs/split_chain/6IYK_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## ...
##
## Extracting sequences
##
## pdb/seq: 1   name: pdbs/split_chain/4IZ6_A.pdb
##    PDB has ALT records, taking A only, rm.alt=TRUE
## pdb/seq: 2   name: pdbs/split_chain/3RG2_A.pdb
## pdb/seq: 3   name: pdbs/split_chain/6IYK_A.pdb
```

```
# Vector containing PDB codes for figure axis
ids <- basename.pdb(pdbs$id)

# Draw schematic alignment
plot(pdbs, labels=ids)
```



```
top3_ids <- hits$pdb.id[1:3]
top3_annot <- pdb.annotate(top3_ids)
write.csv(top3_annot, "POMC_top3_annot.csv")
```

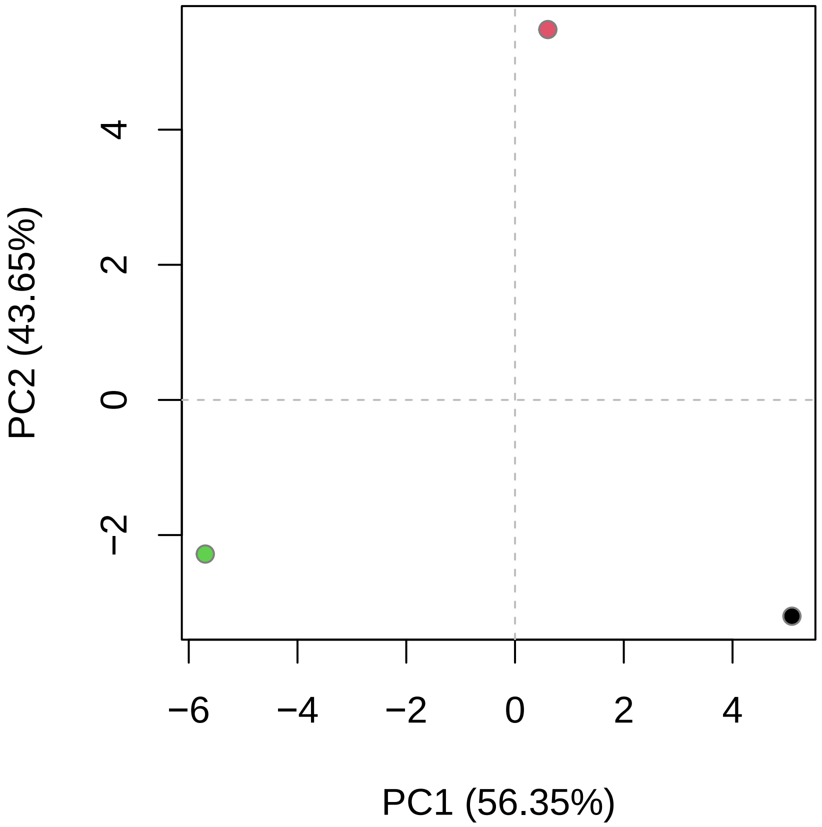
```
# Perform PCA
top3_pca <- pca(pdbs)
plot(top3_pca)
```



```
# Calculate RMSD
rmsd <- rmsd(pdbs)
```

```
## Warning in rmsd(pdbs): No indices provided, using the 520 non NA positions
```

```
# Structure-based clustering
hc.rmsd <- hclust(dist(rmsd))
grps.rmsd <- cutree(hc.rmsd, k=3)

plot(top3_pca, 1:2, col="grey50", bg=grps.rmsd, pch=21, cex=1)
```



```
# Visualize first principal component
top3_pc1 <- mktrj(top3_pca, pc=1, file="POMC_top3_pc1.pdb")
```