ch VDJdb sandbox. ROC curves

Here $Q_{k_i m_j}$ is Q-score for amino acid substitution in aligned sequences k and m at i and j positions respectively. $Q_{k_i m_j}$ represents log_{10} of odds ratios for probabilities to see substitution of amino acid in same and distinct antigens. lengths of sequences k and m are similar and are equal to N. No indels allowed.

To get the weight of an alignment we sum those Q-scores as follows:

$$S_{km} = \sum_{i=1}^{N} Q_{k_i m_i}$$

To compute predicted value for sequences k and m to bind same epitope weight of an alignment was divided by it's length N_{km}:

$$W_{km} = \frac{S_{km}}{N_{km}}$$

Those values were computed using script "ch_roc_curves.py".

Files with values are in folder "seqdata/HomoSapiens/scores/". Scripts are in folder "scripts/".

Point to note: Cysteines were omitted while lengths and weights were calculated

Load data from folder and calculate ROC curves:

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

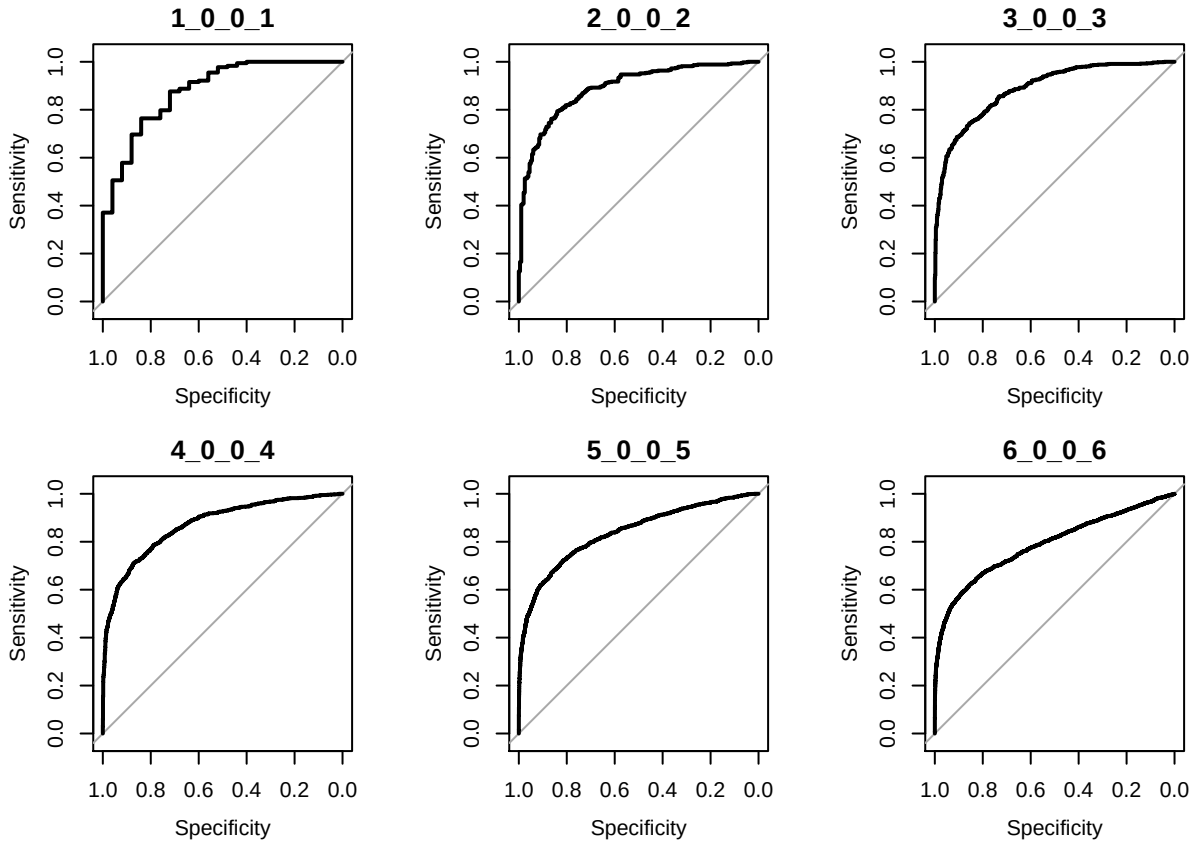
```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
read_file_for_ROC <- function(folder, sub, del, ins, tot, name) {
  suffix <- paste(sub, del, ins, tot, sep="_")
  inpname <- paste(paste(name, suffix, sep="_"), '.txt', sep='')
  df <- read.csv(paste(folder, inpname, sep =""), sep = '\t', stringsAsFactors = F)
  colnames(df) <- c("sameantigen", "score", "pair")
  df$dataset <- suffix
  return(df)
}

folder <- "seqdata/HomoSapiens/scores/"
par(mfrow=c(2,3))
for (s in 1:6) {
  df <- read_file_for_ROC(folder, s, 0, 0, s, 'pairscores')
  dfroc <- roc(sameantigen ~ score, df)
  print(paste(s,0,0,s,sep='_'))
  print(dfroc)
  print('-----------------')
  plot(dfroc, main=paste(s,0,0,s,sep='_'))
}
```

```
## [1] "1_0_0_1"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 25 controls (sameantigen 0) < 178 cases (sameantigen 1).
## Area under the curve: 0.8796
## [1] "-----------------"
## [1] "2_0_0_2"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 200 controls (sameantigen 0) < 436 cases (sameantigen 1).
## Area under the curve: 0.8901
## [1] "-----------------"
## [1] "3_0_0_3"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 898 controls (sameantigen 0) < 676 cases (sameantigen 1).
## Area under the curve: 0.8869
## [1] "-----------------"
## [1] "4_0_0_4"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 3196 controls (sameantigen 0) < 996 cases (sameantigen 1).
## Area under the curve: 0.8706
## [1] "-----------------"
## [1] "5_0_0_5"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 9115 controls (sameantigen 0) < 1463 cases (sameantigen 1).
## Area under the curve: 0.8372
## [1] "-----------------"
## [1] "6_0_0_6"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 20649 controls (sameantigen 0) < 2173 cases (sameantigen 1).
## Area under the curve: 0.7885
## [1] "-----------------"
```


1_0_0_1
2_0_0_2
3_0_0_3
4_0_0_4
5_0_0_5
6_0_0_6
Sensitivity
Specificity
0.0
0.2
0.4
0.6
0.8
1.0