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 k and m sequences are similar and are equal to N.

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 find probability of sequences k and m to bind same epitope maximum and minimum weighted alignments were found for sequences k and m with condition to not look for maximum (minimum) weighted substitution in positions where k and m are similar:

$$S_{(k,m)\,\max} = \sum_{i=1}^{N} Q_{(k,m)_i n_i},$$

where $n_i = (k,m)_i$, if k_i=m_i, or $n_i = \underset{x \in aminoacids}{argmax}\, Q_{(k,m)_i x}$ if $k_i \neq m_i$

$$S_{(k,m)\,\min} = \sum_{i=1}^{N} Q_{(k,m)_i n_i},$$

where $n_i = (k,m)_i$, if k_i=m_i, or $n_i = \underset{x \in aminoacids}{argmin}\, Q_{(k,m)_i x}$ if $k_i \neq m_i$

The probability of sequences k and m to bind same epitope are:

$$P((k\&m) \in sameepitope) = \frac{S_{km} - S_{k\,\min}}{2(S_{k\,\max} - S_{k\,\min})} + \frac{S_{km} - S_{m\,\min}}{2(S_{m\,\max} - S_{m\,\min})}.$$

Those probabilities were computed by using script "ch_roc_curves.py". If somehow either $S_m\,\max = S_m\,\min$ or $S_k\,\max = S_k\,\min$ then those alignments were omitted (Didn't see any, though). Files with probabilites are in folder "seqdata/HomoSapiens/scores/".

Load data from folder:

```
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) < 175 cases (sameantigen 1).
## Area under the curve: 0.8545
## [1] "------------------"
```

```
## [1] "2_0_0_2"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 200 controls (sameantigen 0) < 433 cases (sameantigen 1).
## Area under the curve: 0.8434
## [1] "------------------"
```

```
## [1] "3_0_0_3"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 898 controls (sameantigen 0) < 673 cases (sameantigen 1).
## Area under the curve: 0.796
## [1] "------------------"
```

```
## [1] "4_0_0_4"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 3196 controls (sameantigen 0) < 993 cases (sameantigen 1).
## Area under the curve: 0.7714
## [1] "------------------"
```

```
## [1] "5_0_0_5"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 9115 controls (sameantigen 0) < 1460 cases (sameantigen 1).
```

```
## Area under the curve: 0.7269
## [1] "------------------"
## [1] "6_0_0_6"
##
## Call:
## roc.formula(formula = sameantigen ~ score, data = df)
##
## Data: score in 20649 controls (sameantigen 0) < 2170 cases (sameantigen 1).
## Area under the curve: 0.6796
## [1] "------------------"
```