Mapping specificity, entropy, allosteric changes and substrates in blood coagulation proteases by a high-throughput protease screen

Load all required functions, packages and a ggplot theme for plotting

```
## Loading required package: seqinr

## Warning: package 'seqinr' was built under R version 3.6.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.6.2

## Loading required package: pheatmap

## Warning: package 'pheatmap' was built under R version 3.6.3

## Loading required package: reshape2

## Warning: package 'reshape2' was built under R version 3.6.2

## Loading required package: ggsci

## Warning: package 'ggsci' was built under R version 3.6.3

## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggsci_2.9       reshape2_1.4.3  pheatmap_1.0.12 ggplot2_3.2.1
## [5] seqinr_3.6-1
```

```
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.3         knitr_1.28         magrittr_1.5       MASS_7.3-51.4
##  [5] munsell_0.5.0      colorspace_1.4-1   R6_2.4.1           rlang_0.4.2
##  [9] plyr_1.8.5         stringr_1.4.0      tools_3.6.1        grid_3.6.1
## [13] gtable_0.3.0       xfun_0.14          withr_2.1.2        htmltools_0.4.0
## [17] yaml_2.2.1         ade4_1.7-15        lazyeval_0.2.2     digest_0.6.23
## [21] tibble_2.1.3       lifecycle_0.1.0    crayon_1.3.4       RColorBrewer_1.1-2
## [25] evaluate_0.14      rmarkdown_2.3      stringi_1.4.4      compiler_3.6.1
## [29] pillar_1.4.3       scales_1.1.0       pkgconfig_2.0.3
```

Prepare data for analysis and generate cleavage file

Here the file peptides.txt derived from a maxQuant search can be loaded. In the provided txt file we benchmarked HTPS to characterize trypsin specificity, entropy and block entropy. C terminal and N terminal cleavage sequences are combined and filtered to obtain a list representing unique cleavage windows.
To provide supplementary plots on identified peptide such as MQ SCORE and PEP set debugs_plot=TRUE.

```
input<-preparing_input("peptides.txt", debugs_plot=FALSE)


input_1<-split_cleavage(input, 'Intensity.1', debugs_plot=FALSE)
input_2<-split_cleavage(input, 'Intensity.2', debugs_plot=FALSE)
input_3<-split_cleavage(input, 'Intensity.3', debugs_plot=FALSE)



unique<-data.frame(rbind(input_1,input_2, input_3))
unique<-data.frame(unique(unique))
# identified 4260 unique cleavages

# write cleavages file
cleavages <- cbind.fill(input_1, input_2, input_3)
# change location output file
write.csv(cleavages, "cleavages.csv")
cleavages<-data.frame(cleavages)
```

Load HTPS_db.fasta file

```
HTPS_DB<-read.fasta("HTPS_db.fasta", seqtype = c("AA"), strip.desc = TRUE, seqonly = TRUE)
HTPS_DB<-data.frame(HTPS_DB)
HTPS_DB<-sapply(HTPS_DB, as.character)
```

Retrieve specificity and entropy information for the studied protease and for a control.

A frequency matrix for the studied protease is generated from the frequency position of amino acids in the cleavage windows. For control, a random frequency matrix is generated by sampling the same number of amino acids as contained in the frequency matrix from the natural distribution of amino acids in HTPS_DB.fasta

```
cleavages<-start_fun(cleavages, 'aa_freq.csv')

cleavages_1 <- cleavages[[1]][!is.na(cleavages[[1]])]
cleavages_2 <- cleavages[[2]][!is.na(cleavages[[2]])]
cleavages_3 <- cleavages[[3]][!is.na(cleavages[[3]])]

comb1<-specificity_entropy_tda(cleavages_1, HTPS_DB)
```

```
## Loading required package: reshape

## Warning: package 'reshape' was built under R version 3.6.2

##
## Attaching package: 'reshape'

## The following objects are masked from 'package:reshape2':
##
##     colsplit, melt, recast
```

```
comb2<-specificity_entropy_tda(cleavages_2, HTPS_DB)
comb3<-specificity_entropy_tda(cleavages_3, HTPS_DB)
```

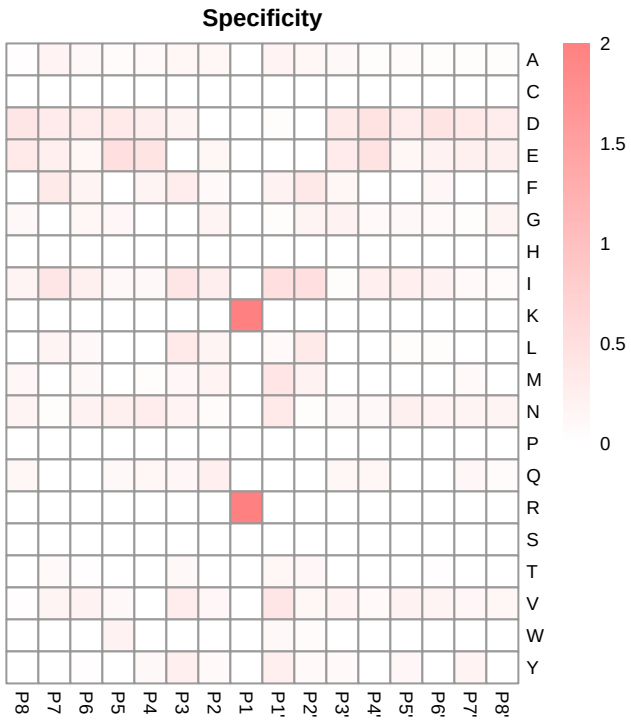
Plot and Output table

Protease specificity

```
speci_combi<-cbind(comb1[['targ_spec']], comb2[['targ_spec']], comb3[['targ_spec']],
                   comb1[['dec_spec']], comb2[['dec_spec']], comb3[['dec_spec']])
speci_combi<-data.frame(cbind(speci_combi$ID, log2(speci_combi[,c(3,7,11,15,19,23)])))
colnames(speci_combi)[1]<-c("ID")
specificity<-calc_stats(speci_combi)
colnames(specificity)<-c("ID","specificity_1", "specificity_2", "specificity_3",
                         "specificity_CTRL_1", "specificity_CTRL_2", "specificity_CTRL_3",
                         "median_specificity", "median_CTRL_specificity", "FC", "pval",
                         "p_val_adj","FC_sign","FC_sign_pos")
specificity$ID<-gsub('X', '', specificity$ID)
write.csv(specificity, "specificity.csv")

pos<-c("P8","P7","P6","P5", "P4", "P3","P2", "P1",
            "P1'","P2'","P3'","P4'","P5'","P6'","P7'","P8'")

tmp0<-specificity[,c(1,14)]
tmp0character<-as.character(tmp0[,1])
tmp0character<-data.frame((strsplit(tmp0character, "_")))
tmp0character<-data.frame(t(tmp0character))
tmp0<-data.frame(cbind(tmp0character, tmp0[,2]))
colnames(tmp0)<-c("AA", "pos", "FC")
# plot
m<-dcast(tmp0, AA~pos)
```

```
## Using FC as value column: use value.var to override.
```

```
rownames(m)<-m$AA
m<-m[,c(2,10:17,3:9)]
colnames(m)<-pos

bk <- c(seq(0,2, length=100))
my_palette <-colorRampPalette(c('grey100','red')) (n=201)

# plot specificity
# change location output folder
pheatmap<-pheatmap(m, cluster_rows = F, cluster_cols = F,
                   breaks =bk, color =my_palette,
                   cellheight=12, cellwidth=12,
                   legend=T,
                   main = "Specificity", fontsize =7)
```



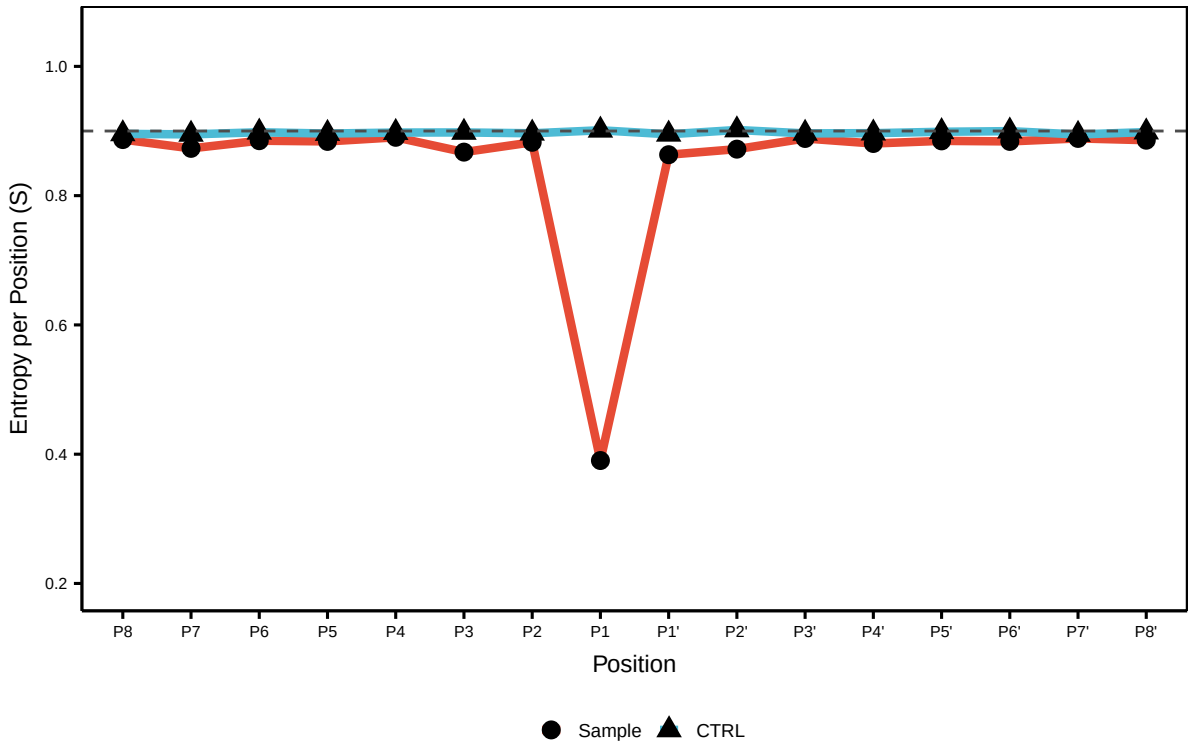
Protease entropy

```
entropy<-cbind(comb1[['targ_entr']], comb2[['targ_entr']], comb3[['targ_entr']],
               comb1[['dec_entr']], comb2[['dec_entr']], comb3[['dec_entr']])
entropy_1<-data.frame(applyBy(entropy, 3, apply, 1, median))
entropy<-cbind(entropy, entropy_1)
```

```
entropy$FC<-entropy$X1-entropy$X2
colnames(entropy)<-c("entropy_1", "entropy_2", "entropy_3",
                     "entropy_CTRL_1", "entropy_CTRL_2", "entropy_CTRL_3",
                     "median_entropy", "median_CTRL_entropy", "FC")
entropy$ID<-rownames(entropy)
entropy<-entropy[,c(10,1:9)]
entropy$ID<-gsub('X', '', entropy$ID)
write.csv(entropy, "entropy.csv")
tmp20<-data.frame(cbind(entropy$median_entropy, entropy$median_CTRL_entropy))
colnames(tmp20)<-c("Sample", "CTRL")
tmp20<-data.frame(tmp20)

tmp20$pos<-pos
tmp21<-melt(tmp20, id=c("pos"))
i<- ggplot(tmp21, aes(x=pos,y= value, group = variable))
i<-i+geom_line(aes(color=variable), size =1.5)
i<-i+geom_point(aes(shape=variable), size =3)
i<-i+scale_color_npg() + scale_x_discrete(name = "Position",
                                          limits = pos)
i<-i+xlab('Position')+ylab('Entropy per Position (S)')
i<-i+theme_Publication()+rmback
```

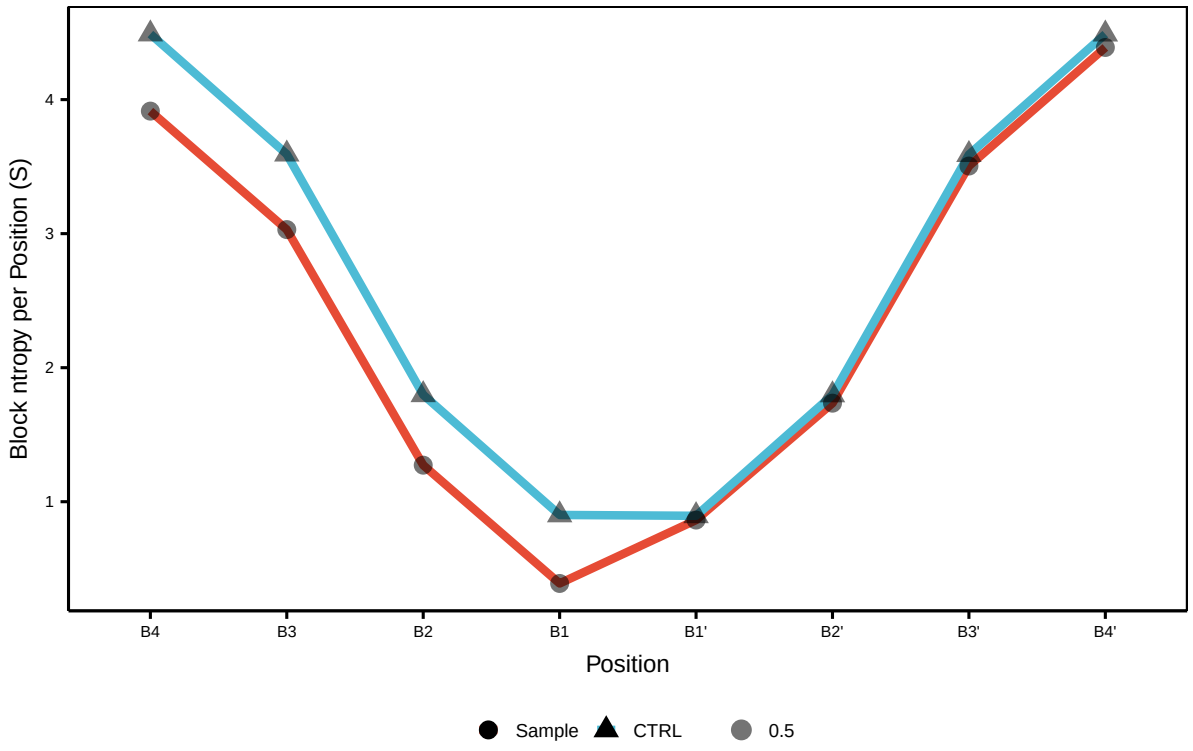
```
## Warning: package 'ggthemes' was built under R version 3.6.3
```

```
i<-i+geom_hline(yintercept = 0.9, linetype = "dashed", color = "grey30")
i<-i+ylim(0.2,1.05)
plot(i)
```



Protease block entropy

```
block_entropy<-block_entropy_function(entropy, debugs_plot=FALSE)
write.csv(block_entropy, "blockentropy.csv")
tmp5<-data.frame(t(block_entropy[,1:8]))

tmp30<-data.frame(tmp5$protease,tmp5$ctrl)

colnames(tmp30)<-c("Sample", "CTRL")
tmp30$pos<-c("B4","B3","B2","B1","B1'","B2'","B3'","B4'")

tmp31<-melt(tmp30, id=c("pos"))
# plot block entropy

i<- ggplot(tmp31, aes(x=pos,y= value, group = variable))
i<-i+geom_line(aes(color=variable), size =1.5)
i<-i+geom_point(aes(shape=variable, alpha=0.5), size =3)
i<-i + scale_x_discrete(name = "Position", limits =c ("B4","B3","B2","B1",
                                                      "B1'","B2'","B3'","B4'"))
i<-i+scale_color_npg()
i<-i+xlab('Position')+ylab('Block ntropy per Position (S)')
i<-i+theme_Publication()+rmback
plot(i)
```


Block ntropy per Position (S)
4
3
2
1
B4
B3
B2
B1
B1'
B2'
B3'
B4'
Position
Sample
CTRL
0.5