Logistic regression for Hbond analysis

Dizhou Wu

3/24/2022

```
suppressMessages(library(caret))
suppressMessages(library(rlang))
suppressMessages(library(ggplot2))
suppressMessages(library(ggmosaic))
suppressMessages(library(glmnet))
```

TM56 TM456

```
# Read data
hbond_stride100_trajectory=read.csv("/deac/salsburyGrp/wud18/md/TM/hbond/hbond_stride100_trajectory_sel
```

```
# Replace headers
Colnames1 <- c("LEU11-HIS152","ARG12-GLU16","ARG12-ASP22","ARG12-MET47","GLU16-GLU16","LYS17-ARG12","SE
Colnames2 <- c("CYS399-ALA406","ALA406-CYS399","PHE15-ARG12","LYS18-PHE15","SER41-VAL198","MET47-GLU44"
Colnames <- c(Colnames1,Colnames2)
colnames(hbond_stride100_trajectory) <- Colnames
```

```
# Add column TM4
a=rep(1, 8000)
b=rep(0, 8000)
c <- c(a,b)
hbond_stride100_trajectory$TM4 <- c
levels(as.factor(hbond_stride100_trajectory$TM4))
```

```
## [1] "0" "1"
```

```
# Train the logistic regression model
m2 <- glm(formula = TM4 ~ ., data=hbond_stride100_trajectory, family='binomial')
```

```
# Save coef
coef <- data.frame(m2$coefficients)
write.csv(coef, "/deac/salsburyGrp/wud18/md/TM/logistic_regression_Hbond/logistic_regression.csv", row.
```

```
# Coef in order
coef_sorted <- sort(m2$coefficients, decreasing = TRUE)
knitr::kable(c(coef_sorted[1:10],coef_sorted[303:312]))
```

	x
(Intercept)	7.932972
LEU132-ARG216	4.187012
ARG72-LEU141	3.357247
PHE275-TRP263	2.892360
THR1-ASP71	2.835278
ALA404-PRO401	2.561503
ARG104-GLU61	2.295686
SER241-SER262	1.670896
ARG104-TRP177	1.540038
TRP263-SER262	1.535242
THR277-ASP135	-1.269049
ILE114-ASN380	-1.305841
ARG72-ASP71	-1.338281
LEU132-ASN131	-1.418311
GLY393-ASP374	-1.636489
TRP86-GLU238	-1.690950
TYR32-LYS248	-1.884502
ASP133-ASN127	-2.432484
ARG12-ASP22	-2.563261
HIS123-PRO124	-3.027473

```
# Elastic net
tuningGrid <- data.frame("alpha" = c(0), "lambda"= c(0.0034375))
set.seed(100)
TM_final_model = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory,
  method = "glmnet",
  lambda=0.0034375,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
)
```

```
TM_final_model
```

```
## glmnet
##
## 16000 samples
##   311 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9563125  0.912625
##
## Tuning parameter 'alpha' was held constant at a value of 0
## Tuning
##  parameter 'lambda' was held constant at a value of 0.0034375
```

```
# Parameters
Beta1 <- m2$coefficients
Beta2 <- as.numeric(coef(TM_final_model$finalModel))
```

```
# Compare them
Comparison <- data.frame("LogisticRegression" = Beta1, "Elastic_net" = Beta2)
rownames(Comparison) <- c('(Intercept)',Colnames)
tmp1 <- Comparison[order(-Comparison$Elastic_net)[1:10],]
tmp2 <- Comparison[order(-Comparison$Elastic_net)[303:312],]
Betas=rbind(tmp1,tmp2)
knitr::kable(Betas)
```

	LogisticRegression	Elastic_net
(Intercept)	7.932972	4.9992872
LEU132-ARG216	4.187012	2.4382781
ARG72-LEU141	3.357247	2.0266173
THR1-ASP71	2.835278	1.9819484
PHE275-TRP263	2.892360	1.9583735
ALA404-PRO401	2.561503	1.7581794
ARG104-GLU61	2.295686	1.4828349
SER241-SER262	1.670896	1.3299225
ARG104-TRP177	1.540038	1.0395606
TRP263-SER262	1.535242	1.0394007
THR277-ASP135	-1.269049	-0.8436615
GLY393-ASP374	-1.636489	-0.8692914
TYR32-LYS248	-1.884502	-0.8792411
ILE114-ASN380	-1.305841	-0.9240794
LEU132-ASN131	-1.418311	-0.9848677
ARG72-ASP71	-1.338281	-1.0737675
TRP86-GLU238	-1.690950	-1.2358593
ARG12-ASP22	-2.563261	-1.2892059
ASP133-ASN127	-2.432484	-1.7013552
HIS123-PRO124	-3.027473	-1.8964858

```
# Elastic net (top 9)
Top9_TM <- c("LEU132-ARG216","ARG72-LEU141","THR1-ASP71","PHE275-TRP263","HIS123-PRO124","ALA404-PRO401
tuningGrid <- data.frame("alpha" = c(0), "lambda"= c(0.0034375))
set.seed(100)
TM_final_model_top9 = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory[ , names(hbond_stride100_trajectory) %in%
    c(Top9_TM,"TM4")],
  method = "glmnet",
  lambda=0.0034375,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
)
```

```
TM_final_model_top9
```

```
## glmnet
##
```

```
## 16000 samples
##     9 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.881625  0.76325
##
## Tuning parameter 'alpha' was held constant at a value of 0
## Tuning
##  parameter 'lambda' was held constant at a value of 0.0034375
```

```
# Elastic net (top 10)
Top10_TM <- c("LEU132-ARG216","ARG72-LEU141","THR1-ASP71","PHE275-TRP263","HIS123-PRO124","ALA404-PRO40
tuningGrid <- data.frame("alpha" = c(0), "lambda"= c(0.0034375))
set.seed(100)
TM_final_model_top10 = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory[ , names(hbond_stride100_trajectory) %in%
    c(Top10_TM,"TM4")],
  method = "glmnet",
  lambda=0.0034375,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
)
```

```
TM_final_model_top10
```

```
## glmnet
##
## 16000 samples
##    10 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8993125  0.798625
##
## Tuning parameter 'alpha' was held constant at a value of 0
## Tuning
##  parameter 'lambda' was held constant at a value of 0.0034375
```

```
# Elastic net (top 11)
Top11_TM <- c("LEU132-ARG216","ARG72-LEU141","THR1-ASP71","PHE275-TRP263","HIS123-PRO124","ALA404-PRO40
tuningGrid <- data.frame("alpha" = c(0), "lambda"= c(0.0034375))
set.seed(100)
```

```
TM_final_model_top11 = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory[ , names(hbond_stride100_trajectory) %in%
    c(Top11_TM,"TM4")],
  method = "glmnet",
  lambda=0.0034375,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
)
```

```
TM_final_model_top11
```

```
## glmnet
##
## 16000 samples
##    11 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.895625  0.79125
##
## Tuning parameter 'alpha' was held constant at a value of 0
## Tuning
##  parameter 'lambda' was held constant at a value of 0.0034375
```

Thrombin TM56

```
## Thrombin TM56
# Read data
hbond_stride100_trajectory=read.csv("/deac/salsburyGrp/wud18/md/TM/hbond/hbond_stride100_thrombin_TM56_
```

```
# Replace headers
Colnames1 <- c("ARG12-GLU16","ARG12-ASP22","ARG12-MET47","GLU16-GLU16","LYS17-ARG12","LYS18-PHE15","SER
Colnames2 <- c("PHE293-VAL289","GLY294-ILE290","THR1-ASP291","ILE37-ASN179","TRP50-SER48","ARG56-SER58"
Colnames <- c(Colnames1,Colnames2)
colnames(hbond_stride100_trajectory) <- Colnames
```

```
# Add column TM4
a=rep(0, 8000)
b=rep(1, 8000)
d <- c(a,b)
hbond_stride100_trajectory$TM4 <- d
levels(as.factor(hbond_stride100_trajectory$TM4))
```

```
## [1] "0" "1"
```

```r
# Train the logistic regression model
m2 <- glm(formula = TM4 ~ ., data=hbond_stride100_trajectory, family='binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
# Elastic net
tuningGrid <- data.frame("alpha" = c(0.875), "lambda"= c(0.0009375))
set.seed(100)
thrombin_TM56_final_model = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory,
  method = "glmnet",
  lambda=0.0009375,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
)
thrombin_TM56_final_model
```

```
## glmnet
##
## 16000 samples
##   275 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.9505    0.901
##
## Tuning parameter 'alpha' was held constant at a value of 0.875
## Tuning
##  parameter 'lambda' was held constant at a value of 0.0009375
```

```r
# Parameters
Beta1 <- m2$coefficients
Beta2 <- as.numeric(coef(thrombin_TM56_final_model$finalModel))
```

```r
# Compare them
Comparison <- data.frame("LogisticRegression" = Beta1, "Elastic_net" = Beta2)
rownames(Comparison) <- c('(Intercept)',Colnames)
tmp1 <- Comparison[order(-Comparison$Elastic_net)[1:10],]
tmp2 <- Comparison[order(-Comparison$Elastic_net)[267:276],]
Betas=rbind(tmp1,tmp2)
knitr::kable(Betas)
```

	LogisticRegression	Elastic_net
TYR32-LYS248	4.469139	3.2015528
HIS123-PRO124	2.923938	2.1713034

	LogisticRegression	Elastic_net
GLU112-GLU108	2.247867	1.8513930
TYR32-LYS171	2.755436	1.7241562
THR213-CYS209	1.949372	1.3960396
ARG173-ASP22	1.932521	1.3372303
GLN60-LYS57	1.637322	1.3292156
ARG109-TYR107	1.492354	1.1848056
ILE37-ASN179	1.397393	1.1198920
ARG56-SER58	1.401726	1.0616207
CYS267-GLU182	-1.156008	-0.8804728
GLU61-ARG56	-1.236893	-0.9165794
SER115-MET116	-1.156650	-0.9234520
ARG104-GLU61	-1.683693	-1.1067488
THR1-GLU6	-1.482651	-1.2037194
ARG233-GLU39	-2.410700	-1.4078448
PHE275-TRP263	-1.955056	-1.4776470
SER58-GLN60	-20.497746	-4.7790560
ARG56-LYS57	-21.755899	-5.4480523
(Intercept)	-14.709829	-10.9355491

```
# Elastic net (top 11)
Top11_thrombin_TM56 <- c("ARG56-LYS57","SER58-GLN60","TYR32-LYS248","HIS123-PRO124","GLU112-GLU108","TY
tuningGrid <- data.frame("alpha" = c(0.875), "lambda"= c(0.0009375))
set.seed(100)
thrombin_TM56_final_model_top11 = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory[ , names(hbond_stride100_trajectory) %in%
    c(Top11_thrombin_TM56,"TM4")],
  method = "glmnet",
  lambda=0.0009375,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
)
thrombin_TM56_final_model_top11
```

```
## glmnet
##
## 16000 samples
##    11 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.8765    0.753
##
## Tuning parameter 'alpha' was held constant at a value of 0.875
## Tuning
##  parameter 'lambda' was held constant at a value of 0.0009375
```

Thrombin TM456

```
## Thrombin TM456
# Read data
hbond_stride100_trajectory=read.csv("/deac/salsburyGrp/wud18/md/TM/hbond/hbond_stride100_thrombin_TM456_
```

```
# Replace headers
Colnames1 <- c("THR1-GLU295","ARG12-GLU16","ARG12-ASP22","ARG12-MET47","GLU16-GLU16","LYS17-ARG12","LYS
Colnames2 <- c("LYS283-GLN287","LYS283-PRO157","ILE286-LEU282","GLN287-LYS283","VAL289-TRP285","ASP291-
Colnames <- c(Colnames1,Colnames2)
colnames(hbond_stride100_trajectory) <- Colnames
```

```
# Add column TM4
a=rep(0, 8000)
b=rep(1, 8000)
d <- c(a,b)
hbond_stride100_trajectory$TM4 <- d
levels(as.factor(hbond_stride100_trajectory$TM4))
```

```
## [1] "0" "1"
```

```
# Train the logistic regression model
m2 <- glm(formula = TM4 ~ ., data=hbond_stride100_trajectory, family='binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# Elastic net
tuningGrid <- data.frame("alpha" = c(0.1875), "lambda"= c(0.00171875))
set.seed(100)
thrombin_TM456_final_model = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory,
  method = "glmnet",
  lambda=0.00171875,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
)
thrombin_TM456_final_model
```

```
## glmnet
##
## 16000 samples
##   279 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy   Kappa
```

```
##   0.9558125  0.911625
##
## Tuning parameter 'alpha' was held constant at a value of 0.1875
##
## Tuning parameter 'lambda' was held constant at a value of 0.00171875
```

```
# Parameters
Beta1 <- m2$coefficients
Beta2 <- as.numeric(coef(thrombin_TM456_final_model$finalModel))
```

```
# Compare them
Comparison <- data.frame("LogisticRegression" = Beta1, "Elastic_net" = Beta2)
rownames(Comparison) <- c('(Intercept)',Colnames)
tmp1 <- Comparison[order(-Comparison$Elastic_net)[1:10],]
tmp2 <- Comparison[order(-Comparison$Elastic_net)[267:276],]
Betas=rbind(tmp1,tmp2)
knitr::kable(Betas)
```

	LogisticRegression	Elastic_net
LEU132-ARG216	9.8208858	4.9690138
TYR32-LYS248	2.9820157	2.2040669
GLU112-GLU108	2.2378330	1.8041989
GLN60-LYS57	1.7403578	1.4286418
GLU205-GLU205	2.3775723	1.4264926
TYR32-LYS171	1.9641646	1.3801081
SER31-LEU28	1.7727834	1.3272187
ARG173-ASP22	2.4027397	1.3036729
TYR83-LYS88	1.5831850	1.2945960
TRP263-SER262	1.6524507	1.2899741
ARG72-ASP71	-0.9524432	-0.7112814
ARG206-MET221	-0.9375861	-0.7222318
LEU132-ASN131	-1.0116711	-0.7838471
ASP219-ASP219	-1.0293187	-0.8100814
ARG173-ASN200	-1.3848588	-0.9610759
HIS123-PRO124	-1.3479261	-0.9842277
ASP133-ASN127	-1.4059140	-1.0001466
SER241-HIS79	-1.3938273	-1.0073551
LYS227-GLU205	-2.0220531	-1.1113055
ARG56-LYS57	-1.9439311	-1.4536160

```
# Elastic net (top 11)
Top11_thrombin_TM456 <- c("LEU132-ARG216","TYR32-LYS248","GLU112-GLU108","ARG56-LYS57","GLN60-LYS57","GL
tuningGrid <- data.frame("alpha" = c(0.875), "lambda"= c(0.0009375))
set.seed(100)
thrombin_TM456_final_model_top11 = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory[ , names(hbond_stride100_trajectory) %in%
    c(Top11_thrombin_TM456,"TM4")],
  method = "glmnet",
  lambda=0.0009375,
  tuneGrid = tuningGrid,
  trControl = trainControl(method = "cv", number = 10)
```

```
)
thrombin_TM456_final_model_top11
```

```
## glmnet
##
## 16000 samples
##    11 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 14400, 14400, 14400, 14400, 14400, 14400, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.824     0.648
##
## Tuning parameter 'alpha' was held constant at a value of 0.875
## Tuning
##  parameter 'lambda' was held constant at a value of 0.0009375
```

Thrombin TM56 TM456

```
## Thrombin TM56 TM456
# Read data
hbond_stride100_trajectory=read.csv("/deac/salsburyGrp/wud18/md/TM/hbond/hbond_stride100_thrombin_TM56_
```

```
# Replace headers
Colnames1 <- c("THR1-GLU295","ARG12-GLU16","ARG12-ASP22","ARG12-MET47","GLU16-GLU16","LYS17-ARG12","LYS
Colnames2 <- c("ILE69-TRP73","SER70-TRP73","ASP71-SER70","ASP71-ASP71","ARG72-GLU295","ARG72-LYS142","L
Colnames <- c(Colnames1,Colnames2)
colnames(hbond_stride100_trajectory) <- Colnames
```

```
# Add column TM4
a=rep(0, 8000)
b=rep(1, 8000)
c=rep(2, 8000)
d <- c(a,b,c)
hbond_stride100_trajectory$TM4 <- d
levels(as.factor(hbond_stride100_trajectory$TM4))
```

```
## [1] "0" "1" "2"
```

```
tuningGrid <- data.frame("alpha" = c(0.875), "lambda"= c(0.00046875))
set.seed(100)
thrombin_TM56_TM456_final_model = train(
  as.factor(TM4) ~ ., data = hbond_stride100_trajectory,
  method = "glmnet",
  lambda = 0.00046875,
  tuneGrid = tuningGrid,
```

```
  family = "multinomial",
  trControl = trainControl(method = "cv", number = 10)
)
```

```
thrombin_TM56_TM456_final_model
```

```
## glmnet
##
## 24000 samples
##   330 predictor
##     3 classes: '0', '1', '2'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 21600, 21600, 21600, 21600, 21600, 21600, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9253333  0.888
##
## Tuning parameter 'alpha' was held constant at a value of 0.875
## Tuning
##  parameter 'lambda' was held constant at a value of 0.00046875
```

```
thrombin_TM56_TM456_final_model$results
```

```
##   alpha     lambda  Accuracy Kappa  AccuracySD    KappaSD
## 1 0.875 0.00046875 0.9253333 0.888 0.002912033 0.00436805
```

```
thrombin_TM56_TM456_final_model$control$p
```

```
## [1] 0.75
```

```
# Parameters
Beta <- coef(thrombin_TM56_TM456_final_model$finalModel)
```

```
Beta1=Beta$`0`[,1]
Beta2=Beta$`1`[,1]
Beta3=Beta$`2`[,1]
```

```
beta1 <- data.frame("thrombin" = c(Beta1[order(-Beta1), drop = FALSE][1:10], Beta1[order(-Beta1), drop =
beta2 <- data.frame("TM56" = c(Beta2[order(-Beta2), drop = FALSE][1:10], Beta2[order(-Beta2), drop = FA
beta3 <- data.frame("TM456" = c(Beta3[order(-Beta3), drop = FALSE][1:10], Beta3[order(-Beta3), drop = FA
```

```
knitr::kable(beta1)
```

thrombin
10.2726496

	thrombin
SER58-GLN60	6.5558926
ARG56-LYS57	1.4561542
ARG173-ASN200	0.8060454
ARG216-ASN131	0.7631101
GLU61-ARG56	0.7324076
ARG233-ASP270	0.7054907
ARG206-MET221	0.6984261
LYS210-ASP211	0.6397046
SER115-MET116	0.6363297
ARG109-TYR107	-1.2669558
ARG173-ASP22	-1.2750293
TYR83-LYS88	-1.2965472
ARG56-GLU61	-1.4336868
GLN60-LYS57	-1.4512492
ARG98-ARG106	-1.6407794
TYR32-LYS171	-1.9679916
GLU112-GLU108	-2.0037005
TYR32-LYS248	-3.2210340
LEU132-ARG216	-3.7497827

```
knitr::kable(beta2)
```

	TM56
TYR126-LEU82	1.9676586
HIS123-PRO124	1.6544617
THR277-SER262	1.0367660
ARG12-ASP22	1.0160817
THR213-CYS209	0.9587330
TRP86-GLU238	0.9110832
TYR32-LYS248	0.7427249
GLU230-LYS227	0.7160818
ARG134-ASP219	0.7004929
ARG254-CYS9	0.6514090
ASP34-GLU30	-1.1071272
THR1-GLU6	-1.1312028
THR1-ASP71	-1.1533160
ARG104-GLU61	-1.1923157
ARG216-ASP211	-1.4141040
ARG233-GLU39	-1.7735465
PHE275-TRP263	-1.8829956
ARG56-PRO59	-2.9858577
ARG56-LYS57	-4.6820139
	-7.2626679

```
knitr::kable(beta3)
```

	TM456
LEU132-ARG216	3.2611183
GLY35-GLU30	1.6518615
THR1-ASP71	1.5314980
ARG72-LEU141	1.3348677
LEU180-GLN192	1.1637130
GLU205-GLU205	1.1630424
TRP263-SER262	0.9019599
SER241-SER262	0.8327754
ARG173-MET47	0.8174286
HIS278-MET258	0.6298286
ARG269-GLU182	-0.8022216
LEU132-ASN131	-0.8079871
LYS283-GLU6	-0.8086607
LYS227-GLU205	-0.8924000
ARG12-ASP22	-0.9133427
ARG134-ASP219	-1.0849946
ASP133-ASN127	-1.2414923
THR277-ASP135	-1.3865522
ARG104-PRO193	-1.7627703
	-3.0099817