SIS NMF All Together Now

November 27, 2014

1 Preparation

```
######################################################################## LIBRARIES
options(java.parameters = "-Xmx4G")

library(survival)

## Loading required package:  splines

library(energy)
library(NMF)

## Loading required package:  methods
## Loading required package:  pkgmaker
## Loading required package:  registry
## Loading required package:  rngtools
## Loading required package:  cluster
## NMF - BioConductor layer [OK] | Shared memory capabilities [NO: bigmemory] | Cores 31/32
##  To enable shared memory capabilities, try:  install.extras('
## NMF
## ')

library(glmulti)

## Loading required package:  rJava
##
## Attaching package:  'glmulti'
##
## The following object is masked from 'package:NMF':
##
##    consensus

library(glmnet)

## Loading required package:  Matrix
## Loaded glmnet 1.9-8

library(RColorBrewer)
library(gplots)

##
## Attaching package:  'gplots'
##
## The following object is masked from 'package:stats':
##
##    lowess
```

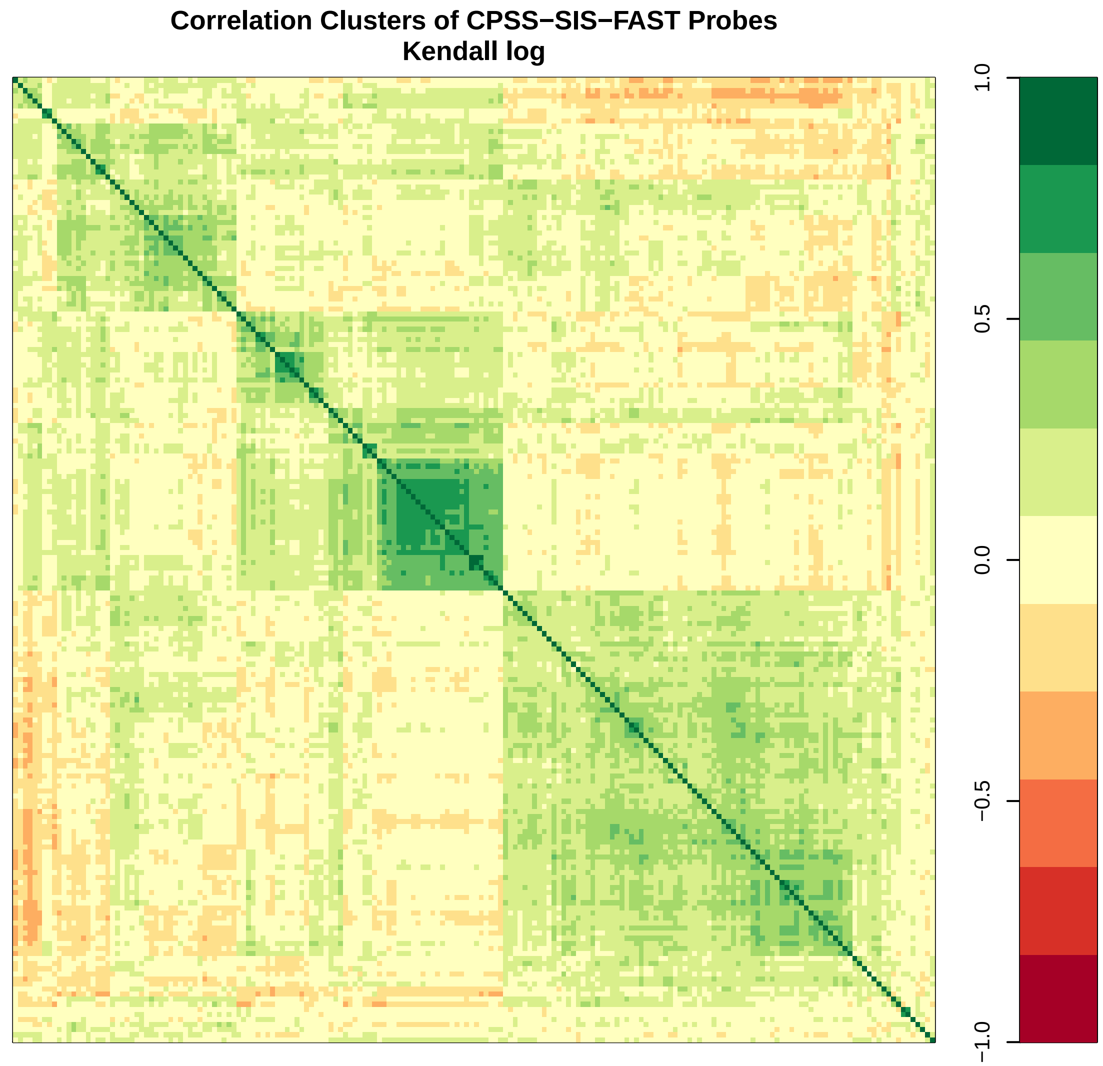
```r
library(xtable)
library(stargazer)

##
## Please cite as:
##
##  Hlavac, Marek (2014).  stargazer:  LaTeX code and ASCII text for well-formatted regression and summary statistics tables.
##  R package version 5.1.  http://CRAN.R-project.org/package=stargazer

load("image.rda")
```

2 Probe selection

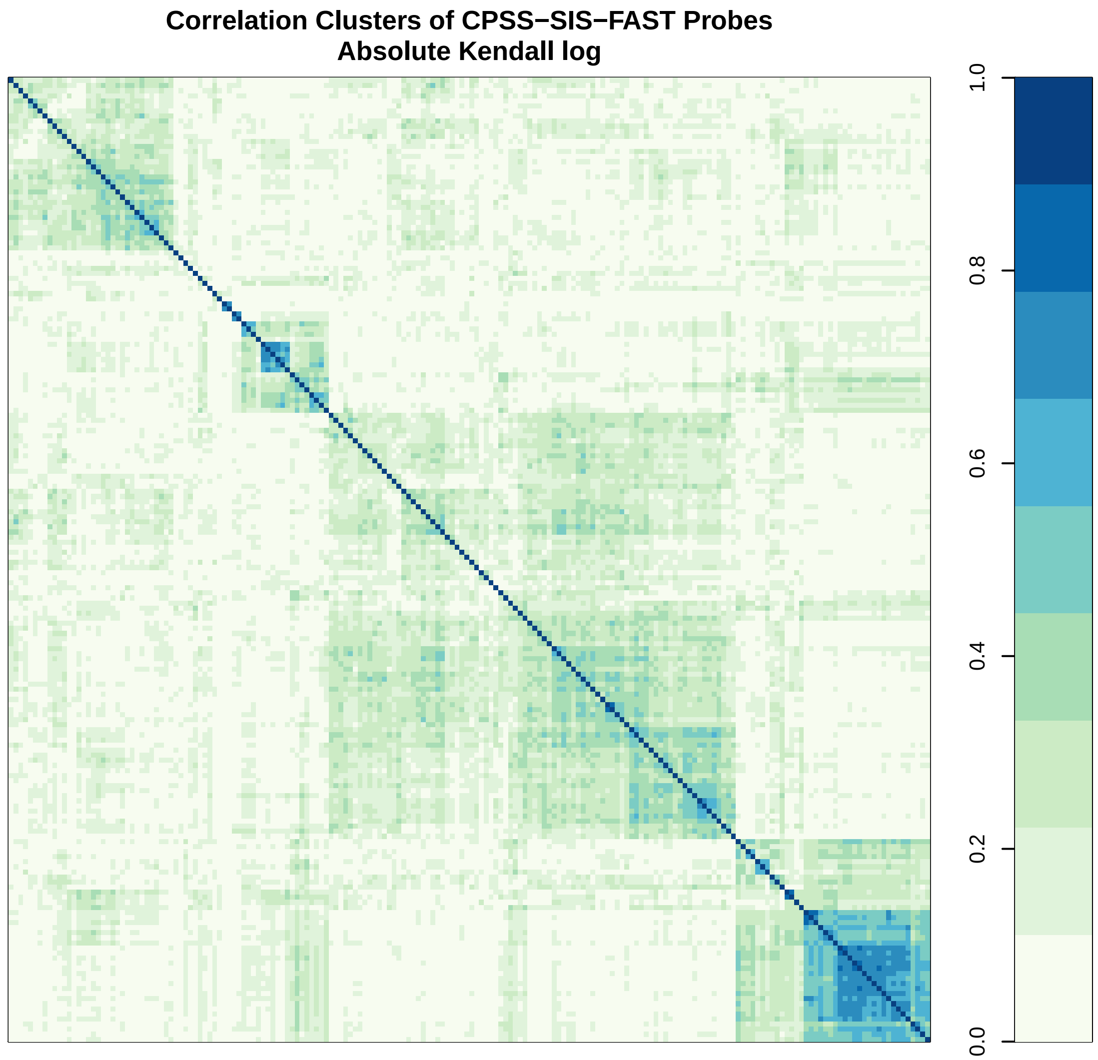
```r
table(cpss.sis$sel)

##
## FALSE  TRUE
## 12810   190

mean(cpss.sis$sel)

## [1] 0.01462
```

3 Expression correlation

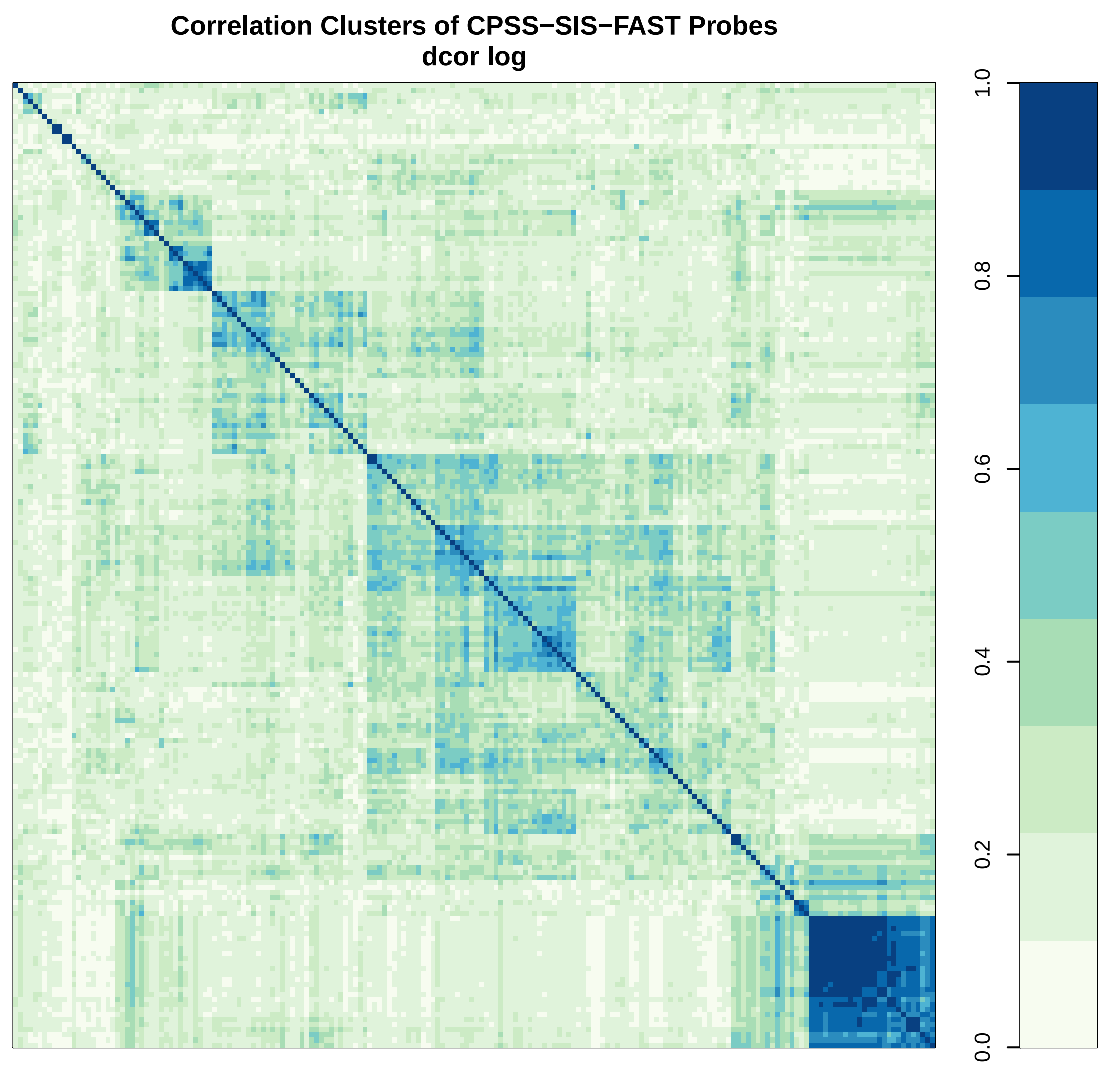
```r
corPlot(x.sel.kcor, main = "Correlation Clusters of CPSS-SIS-FAST Probes\nKendall log",
    useRaster = FALSE)
```




```
corPlot(abs(x.sel.kcor), zlim = c(0, 1), pal = "GnBu", main = "Correlation Clusters of CPSS-SIS-FAST Pro
    useRaster = FALSE)
```

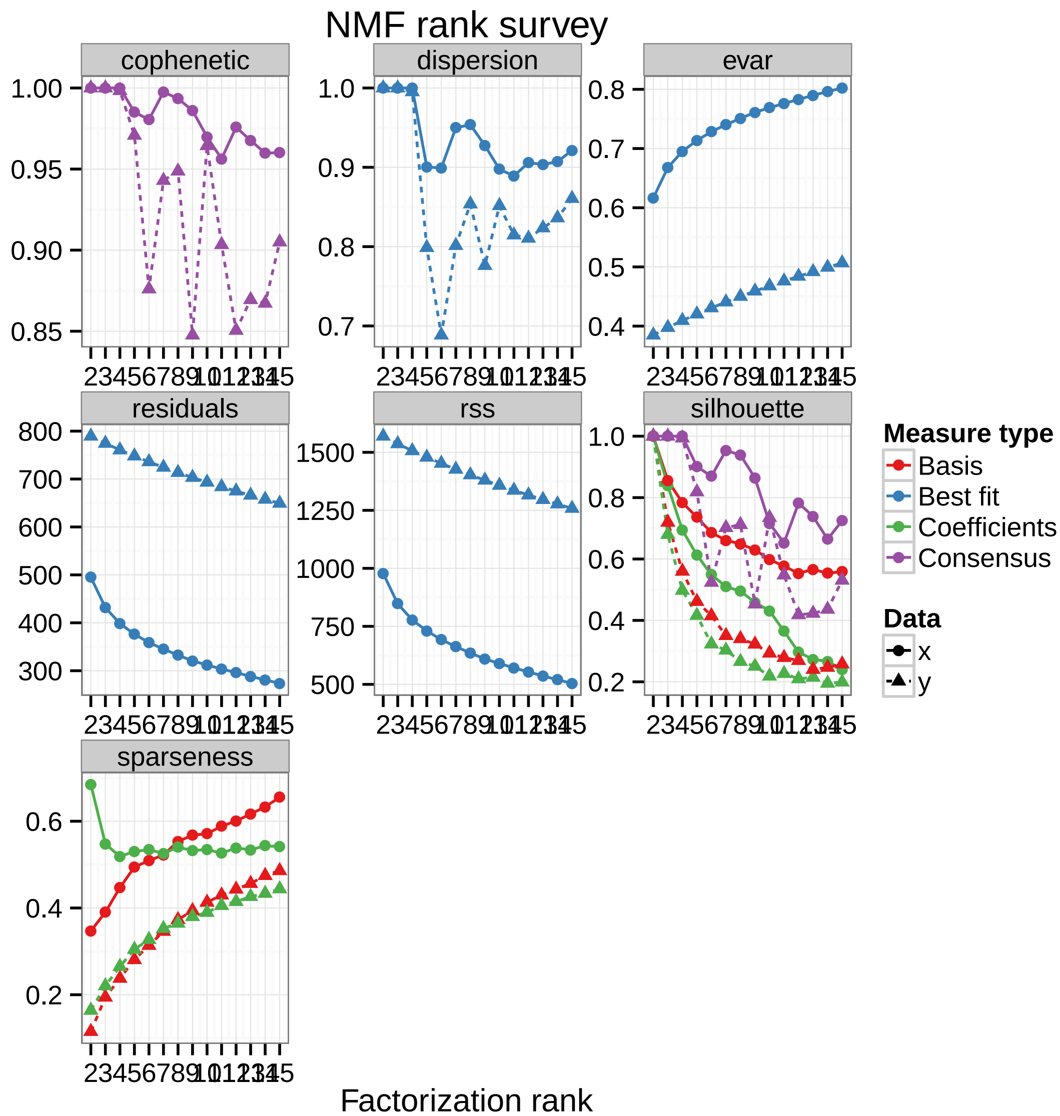



```
corPlot(x.sel.dcor, zlim = c(0, 1), pal = "GnBu", main = "Correlation Clusters of CPSS-SIS-FAST Probes\n
    useRaster = FALSE)
```

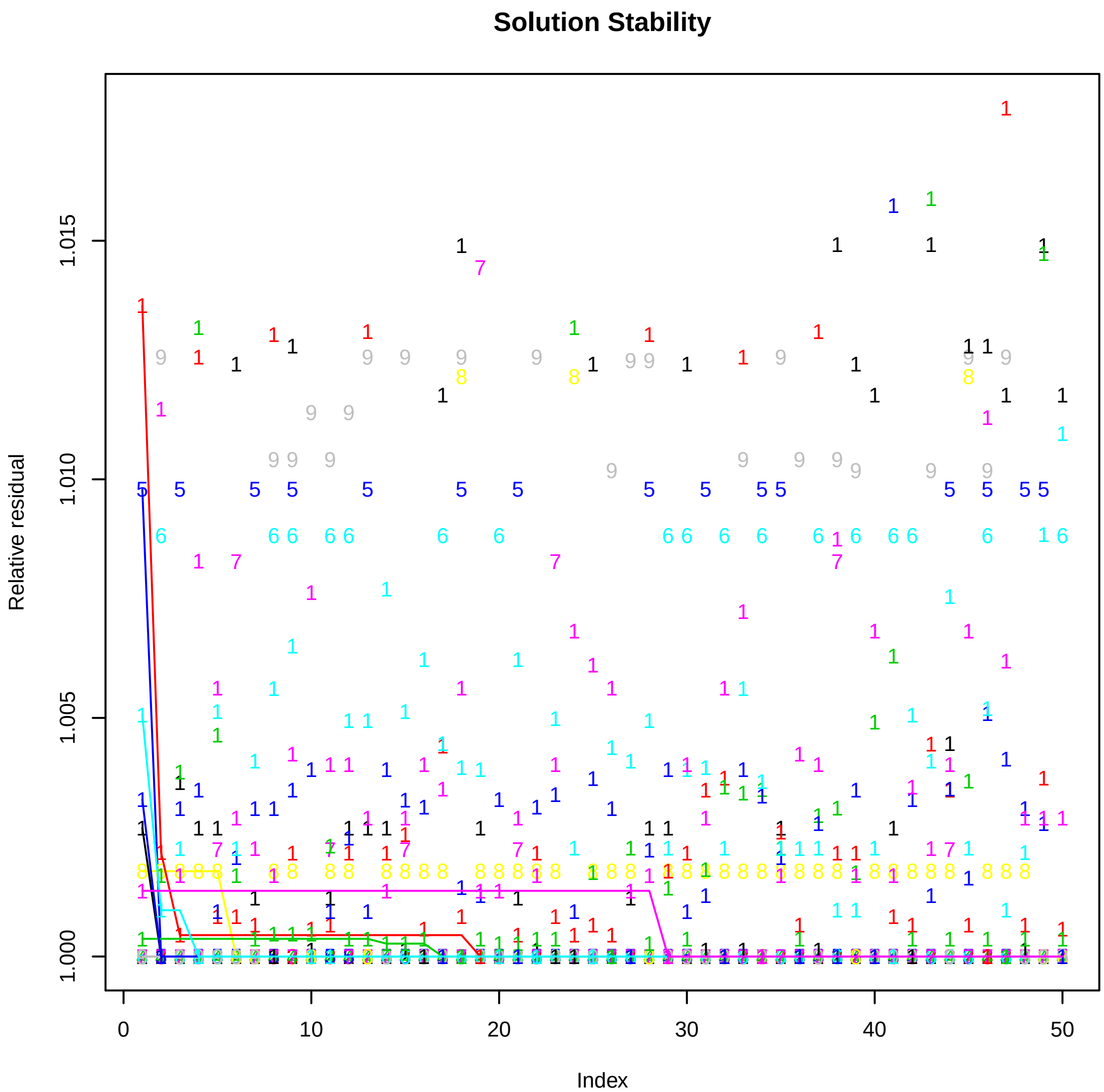


4 Factorization

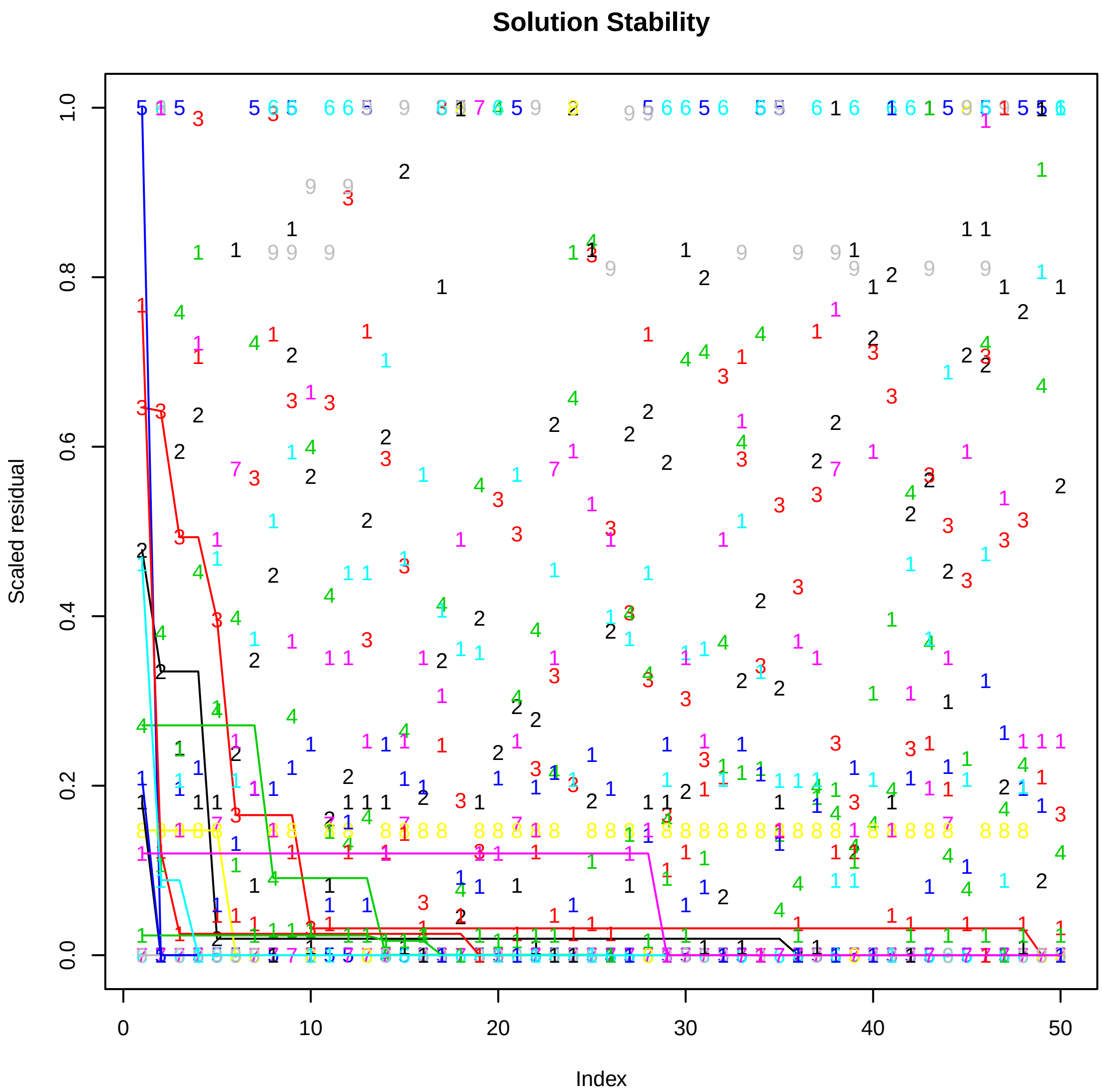
```
plot(temp.nmf.rank, temp.nmf.rank.random[[1]])
```




```
# for (i in min(nmf.rankrange):10) {
# consensusmap(temp.nmf.rank£fit[[which(nmf.rankrange == i)]]) }
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_rel),
    ylab = "Relative residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_rel[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_rel[, i]), col = i)
}
```

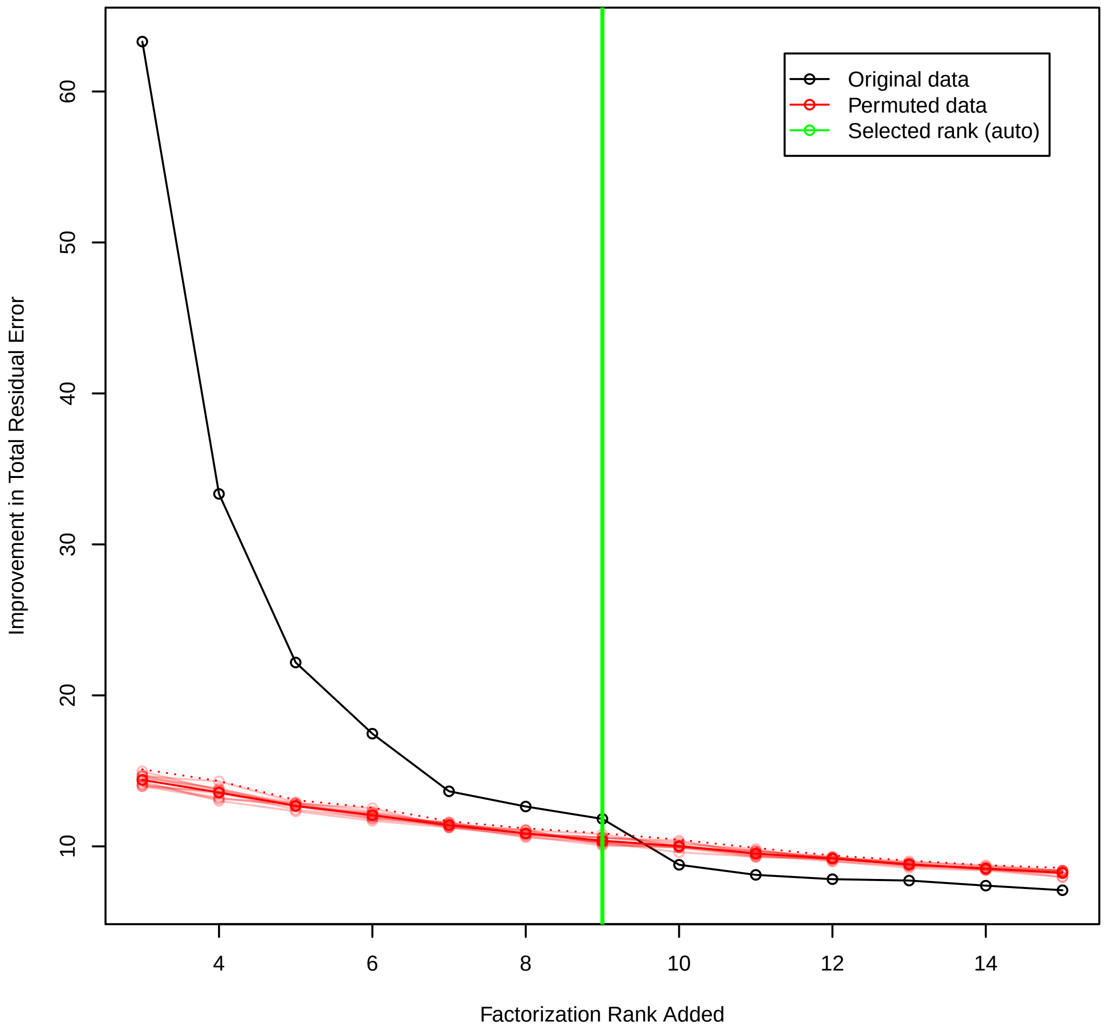



```r
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_scaled),
    ylab = "Scaled residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_scaled[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_scaled[, i]), col = i)
}
```



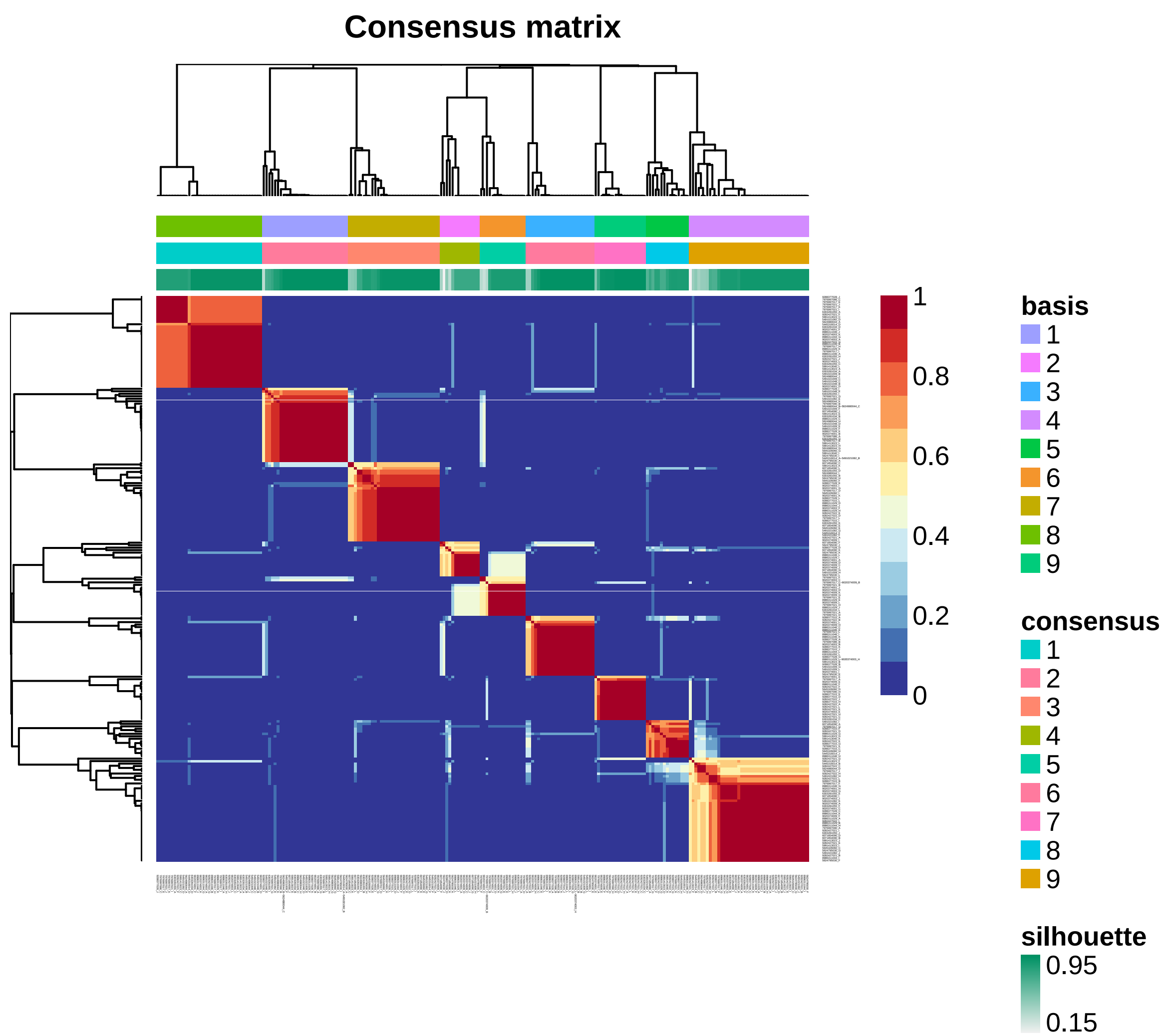

```r
plot(nmf.rankrange[-1], -temp.orig_resids.delta, type = "o", col = "black",
    pch = 21, ylim = range(-c(temp.orig_resids.delta, temp.perm_resids.delta.mean)),
    xlab = "Factorization Rank Added", ylab = "Improvement in Total Residual Error")
lines(nmf.rankrange[-1], -temp.perm_resids.delta.mean, col = "red", type = "o",
    pch = 21, lwd = 1)
for (i in 1:ncol(temp.perm_resids)) {
    lines(nmf.rankrange[-1], -temp.perm_resids.delta[, i], type = "o", col = rgb(1,
        0, 0, 0.25))
}
lines(nmf.rankrange[-1], -temp.perm_resids.delta.threshold, col = "red", lty = "dotted")
if (nmf.rank.wasauto == TRUE) {
    temp.col = "green"
} else {
    temp.col = "blue"
}
abline(v = nmf.rank, col = temp.col, lwd = 2)
legend("topright", legend = c("Original data", "Permuted data", sprintf("Selected rank (%s)",
```

```
    ifelse(nmf.rank.wasauto == TRUE, "auto", "fixed"))), col = c("black", "red",
    temp.col), lty = "solid", pch = 21, inset = 0.05)
```

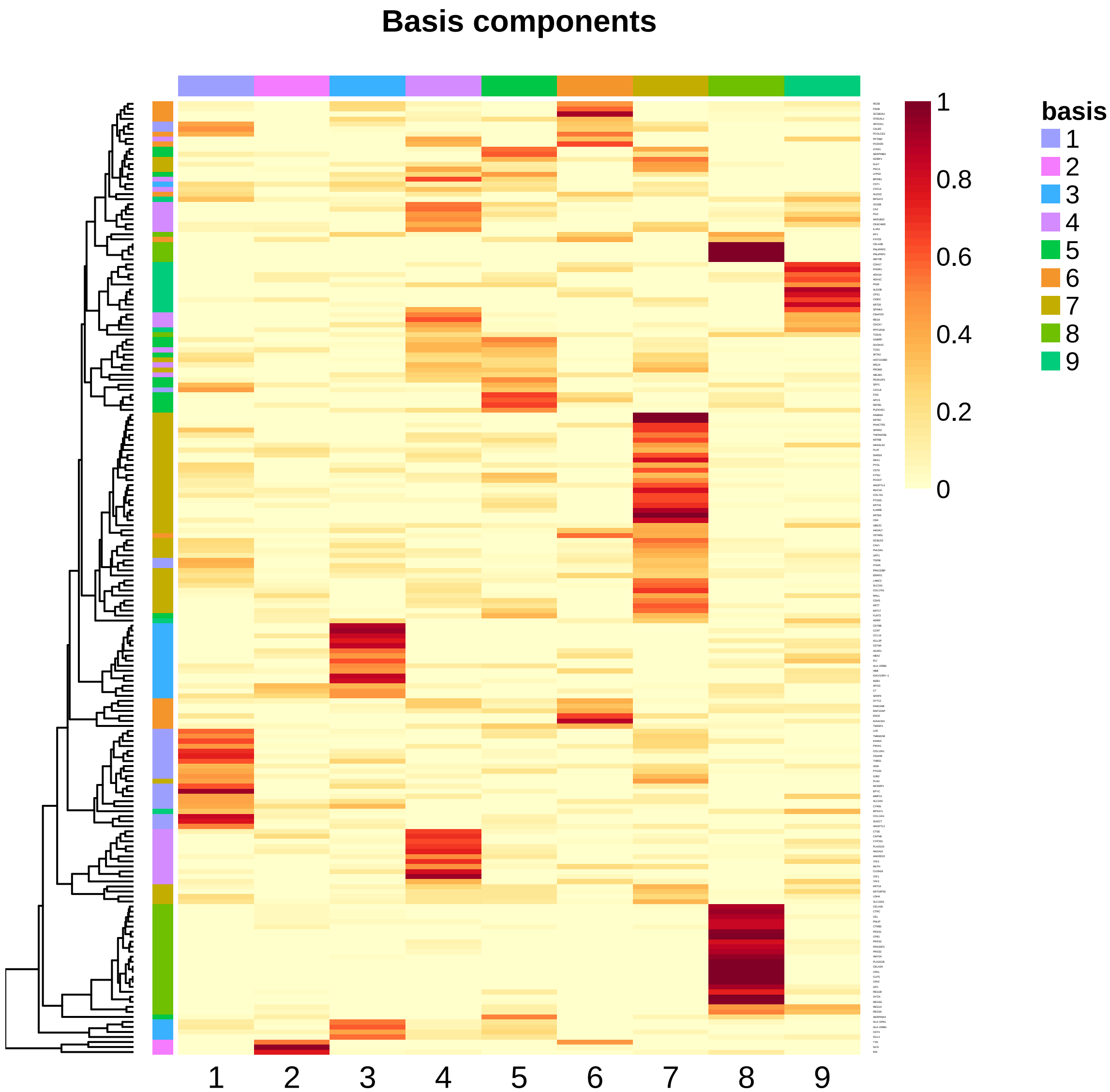


4.1 Fit

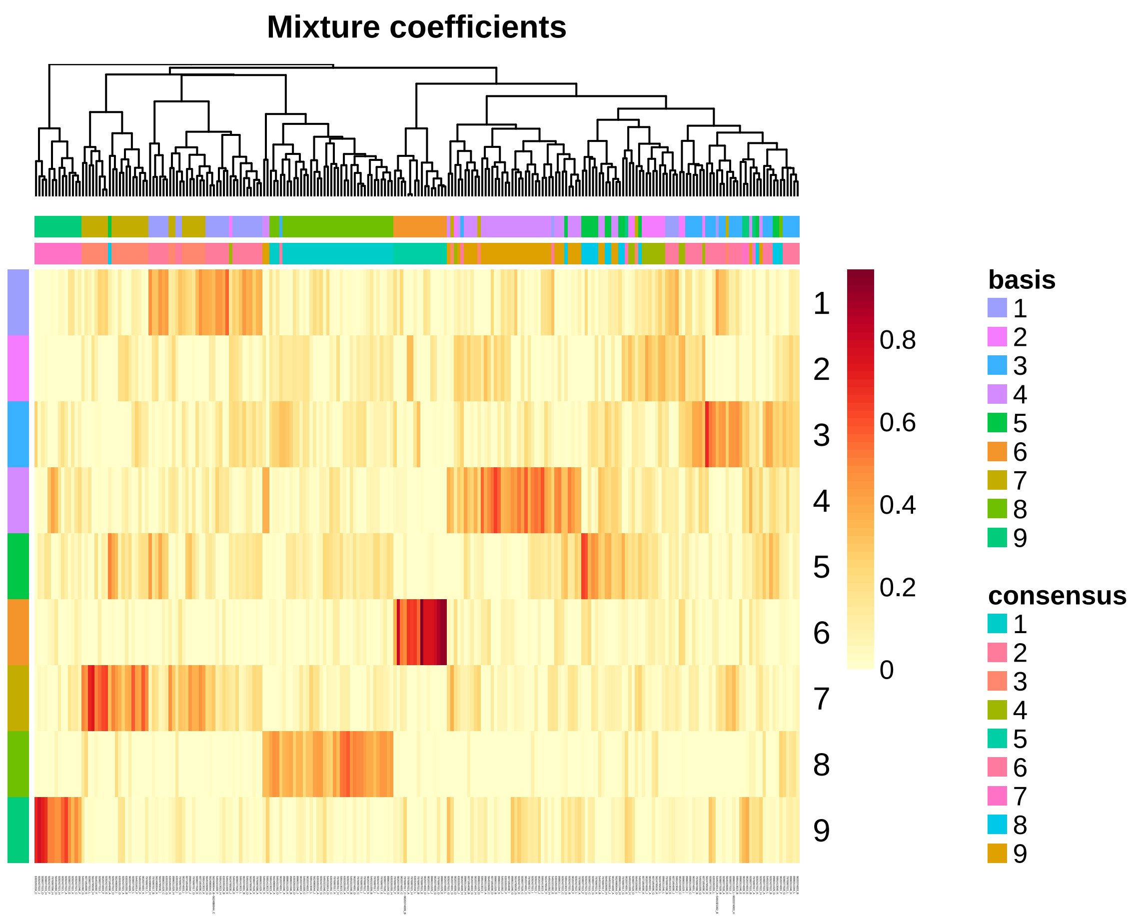
```
consensusmap(xlin.scaled.sel.nmf)
```



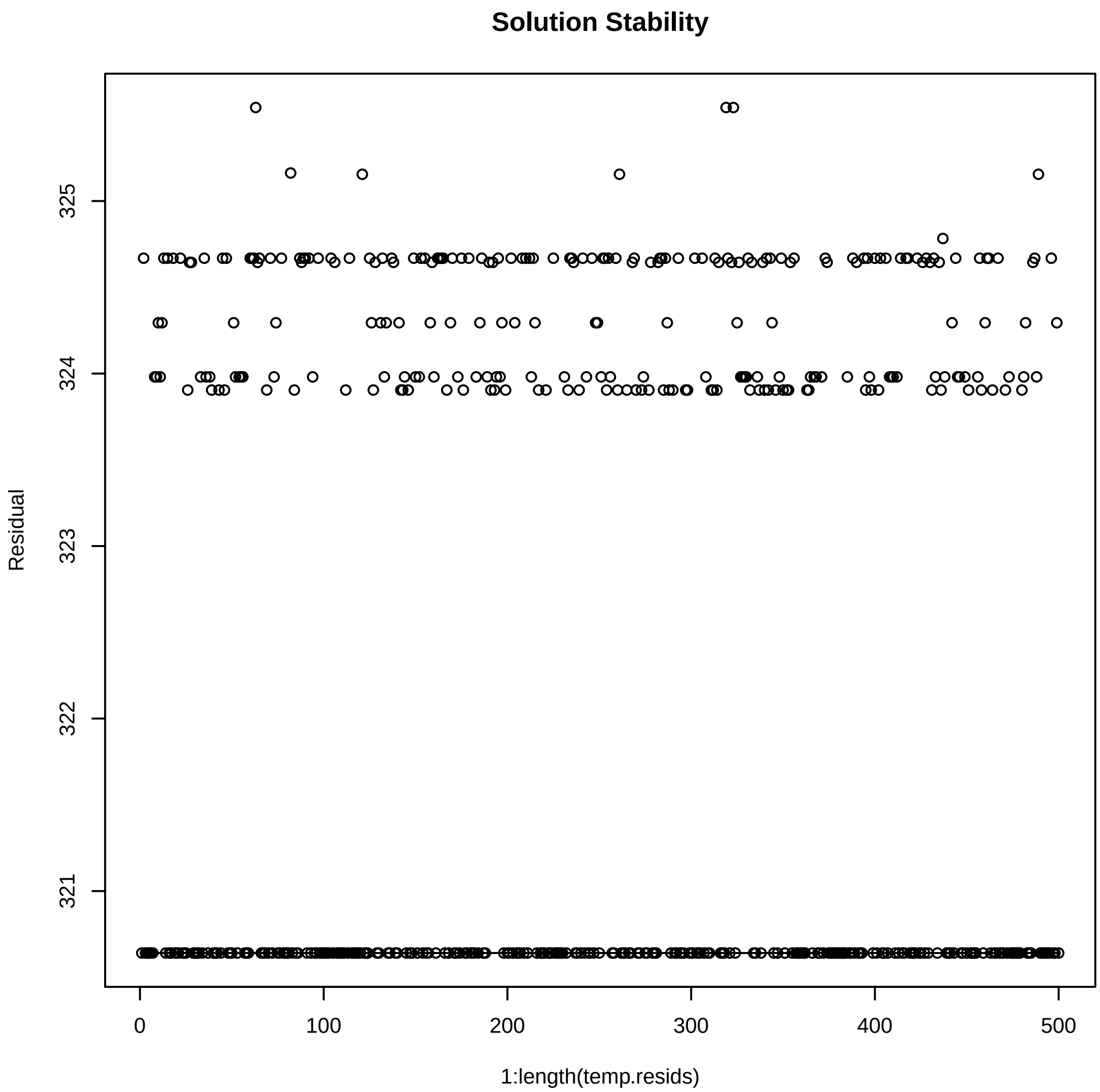

```
basismap(xlin.scaled.sel.nmf)
```




```
coefmap(xlin.scaled.sel.nmf)
```




```
temp.resids = sapply(xlin.scaled.sel.nmf, residuals)
plot(1:length(temp.resids), temp.resids, ylab = "Residual", main = "Solution Stability")
lines(1:length(temp.resids), cummin(temp.resids))
```



4.2 Component CPV associations

4.2.1 Outcome: Diagnosis to recurrence

```
for (i in 1:ncol(coefs.diag_rec)) {
    print(summary(coxph(y.diag_rec ~ coefs.diag_rec[, i])))
}
```

```
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                         coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_rec[, i]   4.72    112.61     1.62 2.92   0.0035
##
##                       exp(coef) exp(-coef) lower .95 upper .95
```

```
## coefs.diag_rec[, i]       113    0.00888      4.71      2690
##
## Concordance= 0.576  (se = 0.036 )
## Rsquare= 0.072   (max possible= 0.997 )
## Likelihood ratio test= 7.73  on 1 df,   p=0.00544
## Wald test            = 8.51  on 1 df,   p=0.00353
## Score (logrank) test = 8.7  on 1 df,   p=0.00319
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                      coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_rec[, i]  2.33     10.25     2.15 1.08     0.28
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      10.3     0.0975     0.152       692
##
## Concordance= 0.529  (se = 0.036 )
## Rsquare= 0.011   (max possible= 0.997 )
## Likelihood ratio test= 1.16  on 1 df,   p=0.282
## Wald test            = 1.17  on 1 df,   p=0.279
## Score (logrank) test = 1.18  on 1 df,   p=0.278
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_rec[, i] -3.1418    0.0432   1.8776 -1.67    0.094
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]    0.0432       23.1   0.00109      1.71
##
## Concordance= 0.565  (se = 0.036 )
## Rsquare= 0.028   (max possible= 0.997 )
## Likelihood ratio test= 2.95  on 1 df,   p=0.0861
## Wald test            = 2.8  on 1 df,   p=0.0943
## Score (logrank) test = 2.82  on 1 df,   p=0.0929
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                       coef exp(coef) se(coef)  z Pr(>|z|)
## coefs.diag_rec[, i] -1.530     0.217    1.531 -1     0.32
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.217       4.62    0.0108      4.36
##
```

```
## Concordance= 0.527  (se = 0.036 )
## Rsquare= 0.01   (max possible= 0.997 )
## Likelihood ratio test= 1.04  on 1 df,   p=0.308
## Wald test            = 1  on 1 df,   p=0.318
## Score (logrank) test = 1  on 1 df,   p=0.317
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                       coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_rec[, i]   5.65    284.95     1.84 3.07   0.0022
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]       285    0.00351      7.67     10580
##
## Concordance= 0.603  (se = 0.036 )
## Rsquare= 0.079   (max possible= 0.997 )
## Likelihood ratio test= 8.51  on 1 df,   p=0.00353
## Wald test            = 9.39  on 1 df,   p=0.00218
## Score (logrank) test = 9.51  on 1 df,   p=0.00204
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_rec[, i] -3.7057    0.0246   4.5383 -0.82     0.41
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]    0.0246       40.7  3.37e-06       179
##
## Concordance= 0.525  (se = 0.035 )
## Rsquare= 0.007   (max possible= 0.997 )
## Likelihood ratio test= 0.71  on 1 df,   p=0.398
## Wald test            = 0.67  on 1 df,   p=0.414
## Score (logrank) test = 0.67  on 1 df,   p=0.414
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                       coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_rec[, i]   6.10    443.90     1.52 4.02  5.8e-05
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]       444    0.00225      22.8      8661
##
## Concordance= 0.656  (se = 0.036 )
## Rsquare= 0.117   (max possible= 0.997 )
```

```
## Likelihood ratio test= 12.9  on 1 df,   p=0.000322
## Wald test            = 16.2  on 1 df,   p=5.79e-05
## Score (logrank) test = 16.7  on 1 df,   p=4.32e-05
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_rec[, i] -1.632     0.196    1.264 -1.29      0.2
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.196       5.11    0.0164      2.33
##
## Concordance= 0.565  (se = 0.035 )
## Rsquare= 0.017   (max possible= 0.997 )
## Likelihood ratio test= 1.8  on 1 df,   p=0.18
## Wald test            = 1.67  on 1 df,   p=0.197
## Score (logrank) test = 1.68  on 1 df,   p=0.195
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                           coef exp(coef)  se(coef)     z Pr(>|z|)
## coefs.diag_rec[, i] -8.058756  0.000316  4.089621 -1.97    0.049
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]  0.000316       3161  1.04e-07     0.958
##
## Concordance= 0.553  (se = 0.036 )
## Rsquare= 0.042   (max possible= 0.997 )
## Likelihood ratio test= 4.43  on 1 df,   p=0.0353
## Wald test            = 3.88  on 1 df,   p=0.0488
## Score (logrank) test = 3.92  on 1 df,   p=0.0478
```

4.2.2 Outcome: Diagnosis to disease-specific death

```
for (i in 1:ncol(coefs.diag_dsd)) {
    print(summary(coxph(y.diag_dsd ~ coefs.diag_dsd[, i])))
}
```

```
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                        coef exp(coef) se(coef)   z Pr(>|z|)
## coefs.diag_dsd[, i]   5.73    308.63     1.59 3.6  0.00032
##
```

```
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]       309    0.00324      13.6      6996
##
## Concordance= 0.603  (se = 0.038 )
## Rsquare= 0.098   (max possible= 0.995 )
## Likelihood ratio test= 11.3  on 1 df,   p=0.000766
## Wald test            = 13  on 1 df,   p=0.000319
## Score (logrank) test = 13.4  on 1 df,   p=0.000257
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                      coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_dsd[, i] 1.35      3.85     2.29 0.59     0.56
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      3.85       0.26    0.0432       343
##
## Concordance= 0.513  (se = 0.037 )
## Rsquare= 0.003   (max possible= 0.995 )
## Likelihood ratio test= 0.34  on 1 df,   p=0.557
## Wald test            = 0.35  on 1 df,   p=0.556
## Score (logrank) test = 0.35  on 1 df,   p=0.556
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                          coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_dsd[, i] -5.88825   0.00277  2.18235 -2.7    0.007
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]   0.00277        361  3.85e-05       0.2
##
## Concordance= 0.605  (se = 0.038 )
## Rsquare= 0.07   (max possible= 0.995 )
## Likelihood ratio test= 8.02  on 1 df,   p=0.00463
## Wald test            = 7.28  on 1 df,   p=0.00697
## Score (logrank) test = 7.45  on 1 df,   p=0.00635
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                         coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_dsd[, i] -3.8704    0.0208   1.8068 -2.14    0.032
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]    0.0208         48  0.000604      0.72
```

```
##
## Concordance= 0.585  (se = 0.038 )
## Rsquare= 0.045   (max possible= 0.995 )
## Likelihood ratio test= 5.11  on 1 df,   p=0.0238
## Wald test            = 4.59  on 1 df,   p=0.0322
## Score (logrank) test = 4.66  on 1 df,   p=0.031
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                     coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_dsd[, i] 2.17      8.80     1.89 1.15     0.25
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]       8.8      0.114     0.215       359
##
## Concordance= 0.538  (se = 0.037 )
## Rsquare= 0.011   (max possible= 0.995 )
## Likelihood ratio test= 1.27  on 1 df,   p=0.26
## Wald test            = 1.32  on 1 df,   p=0.251
## Score (logrank) test = 1.32  on 1 df,   p=0.25
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                         coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_dsd[, i] -5.97982   0.00253  5.21022 -1.15     0.25
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]   0.00253        395  9.29e-08      68.9
##
## Concordance= 0.559  (se = 0.037 )
## Rsquare= 0.013   (max possible= 0.995 )
## Likelihood ratio test= 1.46  on 1 df,   p=0.227
## Wald test            = 1.32  on 1 df,   p=0.251
## Score (logrank) test = 1.32  on 1 df,   p=0.25
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                        coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_dsd[, i]    7.67   2133.58     1.52 5.04  4.7e-07
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      2134   0.000469       108     42096
##
## Concordance= 0.672  (se = 0.037 )
## Rsquare= 0.164   (max possible= 0.995 )
```

```
## Likelihood ratio test= 19.7  on 1 df,   p=9.15e-06
## Wald test            = 25.4  on 1 df,   p=4.7e-07
## Score (logrank) test = 26.9  on 1 df,   p=2.14e-07
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_dsd[, i] -0.787     0.455    1.267 -0.62     0.53
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     0.455        2.2     0.038      5.45
##
## Concordance= 0.552  (se = 0.036 )
## Rsquare= 0.004   (max possible= 0.995 )
## Likelihood ratio test= 0.4  on 1 df,   p=0.527
## Wald test            = 0.39  on 1 df,   p=0.535
## Score (logrank) test = 0.39  on 1 df,   p=0.534
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_dsd[, i] -1.331     0.264    3.773 -0.35     0.72
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     0.264       3.79  0.000162       430
##
## Concordance= 0.484  (se = 0.037 )
## Rsquare= 0.001   (max possible= 0.995 )
## Likelihood ratio test= 0.13  on 1 df,   p=0.721
## Wald test            = 0.12  on 1 df,   p=0.724
## Score (logrank) test = 0.12  on 1 df,   p=0.724
```

4.2.3 Outcome: Recurrence to disease-specific death

```
for (i in 1:ncol(coefs.recr_dsd)) {
    print(summary(coxph(y.recr_dsd ~ coefs.recr_dsd[, i])))
}
```

```
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                       coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.recr_dsd[, i]  4.03     56.43     1.65 2.44    0.015
##
```

```
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      56.4     0.0177      2.21      1442
##
## Concordance= 0.593  (se = 0.041 )
## Rsquare= 0.064   (max possible= 0.997 )
## Likelihood ratio test= 5.4  on 1 df,   p=0.0202
## Wald test            = 5.95  on 1 df,   p=0.0147
## Score (logrank) test = 6.05  on 1 df,   p=0.0139
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                       coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.recr_dsd[, i] 0.371     1.449    2.402 0.15     0.88
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      1.45       0.69    0.0131       161
##
## Concordance= 0.483  (se = 0.041 )
## Rsquare= 0   (max possible= 0.997 )
## Likelihood ratio test= 0.02  on 1 df,   p=0.877
## Wald test            = 0.02  on 1 df,   p=0.877
## Score (logrank) test = 0.02  on 1 df,   p=0.877
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                          coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.recr_dsd[, i] -8.14606   0.00029  2.61554 -3.11   0.0018
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]   0.00029       3450  1.72e-06    0.0488
##
## Concordance= 0.621  (se = 0.041 )
## Rsquare= 0.124   (max possible= 0.997 )
## Likelihood ratio test= 10.8  on 1 df,   p=0.00104
## Wald test            = 9.7  on 1 df,   p=0.00184
## Score (logrank) test = 10.1  on 1 df,   p=0.00152
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                         coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.recr_dsd[, i] -3.0076    0.0494   1.9267 -1.56     0.12
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]    0.0494       20.2   0.00113      2.16
```

```
##
## Concordance= 0.564  (se = 0.041 )
## Rsquare= 0.032   (max possible= 0.997 )
## Likelihood ratio test= 2.62  on 1 df,   p=0.105
## Wald test            = 2.44  on 1 df,   p=0.119
## Score (logrank) test = 2.45  on 1 df,   p=0.118
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.recr_dsd[, i] -2.9819    0.0507   2.1425 -1.39     0.16
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]    0.0507       19.7  0.000761      3.38
##
## Concordance= 0.561  (se = 0.041 )
## Rsquare= 0.025   (max possible= 0.997 )
## Likelihood ratio test= 2.04  on 1 df,   p=0.154
## Wald test            = 1.94  on 1 df,   p=0.164
## Score (logrank) test = 1.94  on 1 df,   p=0.163
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.recr_dsd[, i] -3.5325    0.0292   5.6966 -0.62     0.54
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]    0.0292       34.2  4.14e-07      2064
##
## Concordance= 0.566  (se = 0.041 )
## Rsquare= 0.005   (max possible= 0.997 )
## Likelihood ratio test= 0.4  on 1 df,   p=0.527
## Wald test            = 0.38  on 1 df,   p=0.535
## Score (logrank) test = 0.38  on 1 df,   p=0.535
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                       coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.recr_dsd[, i]   5.17    175.82     1.75 2.95   0.0032
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]       176    0.00569      5.67      5453
##
## Concordance= 0.639  (se = 0.041 )
## Rsquare= 0.088   (max possible= 0.997 )
```

```
## Likelihood ratio test= 7.5  on 1 df,   p=0.00618
## Wald test            = 8.7  on 1 df,   p=0.00318
## Score (logrank) test = 8.94  on 1 df,   p=0.00279
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                        coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.recr_dsd[, i] 0.721     2.057    1.337 0.54     0.59
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      2.06      0.486      0.15      28.3
##
## Concordance= 0.484  (se = 0.039 )
## Rsquare= 0.003   (max possible= 0.997 )
## Likelihood ratio test= 0.28  on 1 df,   p=0.596
## Wald test            = 0.29  on 1 df,   p=0.589
## Score (logrank) test = 0.29  on 1 df,   p=0.589
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                           coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.recr_dsd[, i]    11.04  62057.78     3.73 2.96   0.0031
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]     62058   1.61e-05      41.2  93474988
##
## Concordance= 0.62  (se = 0.04 )
## Rsquare= 0.085   (max possible= 0.997 )
## Likelihood ratio test= 7.18  on 1 df,   p=0.00739
## Wald test            = 8.74  on 1 df,   p=0.00312
## Score (logrank) test = 9.08  on 1 df,   p=0.00258
```

4.2.4 Purity

```
apply(coefs, 2, function(xc) cor.test(samps$purity_qpure, xc, method = "kendall"))
```

```
## $mg.1
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -2.131, p-value = 0.03313
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.1204
```

```
##
##
## $mg.2
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.836, p-value = 0.0001248
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.2219
##
##
## $mg.3
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -5.8, p-value = 6.647e-09
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.3277
##
##
## $mg.4
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 2.066, p-value = 0.03885
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.1166
##
##
## $mg.5
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.469, p-value = 0.0005224
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## -0.197
##
##
## $mg.6
##
##  Kendall's rank correlation tau
##
```

```
## data:  samps$purity_qpure and xc
## z = 1.281, p-value = 0.2001
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## 0.07383
##
##
## $mg.7
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -0.8378, p-value = 0.4022
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.04738
##
##
## $mg.8
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.48, p-value = 0.0005021
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.2073
##
##
## $mg.9
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 1.855, p-value = 0.06365
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.1067
```

4.3 MTC P-values

```
xlin.scaled.sel.nmf.cpv.pvals = data.frame(surv.diag_rec.p = apply(coefs.diag_rec,
    2, function(xc) pchisq(2 * diff(coxph(y.diag_rec ~ xc)$loglik), df = 1,
        lower.tail = FALSE)), surv.diag_rec.c = apply(coefs.diag_rec, 2, function(xc) coef(coxph(y.diag_
    xc))), surv.diag_dsd.p = apply(coefs.diag_dsd, 2, function(xc) pchisq(2 *
    diff(coxph(y.diag_dsd ~ xc)$loglik), df = 1, lower.tail = FALSE)), surv.diag_dsd.c = apply(coefs.dia
    2, function(xc) coef(coxph(y.diag_dsd ~ xc))), surv.recr_dsd.p = apply(coefs.recr_dsd,
    2, function(xc) pchisq(2 * diff(coxph(y.recr_dsd ~ xc)$loglik), df = 1,
```

```
        lower.tail = FALSE)), surv.recr_dsd.c = apply(coefs.recr_dsd, 2, function(xc) coef(coxph(y.recr_
    xc))), pure.p = apply(coefs, 2, function(xc) cor.test(samps$purity_qpure,
    xc, method = "kendall")$p.value), pure.s = apply(coefs, 2, function(xc) cor.test(samps$purity_qpure,
    xc, method = "kendall")$statistic))
temp.pvals = as.matrix(xlin.scaled.sel.nmf.cpv.pvals[, grepl("\\.p$", colnames(xlin.scaled.sel.nmf.cpv.p
temp.pvals.FWER = matrix(p.adjust(as.vector(temp.pvals), "holm"), nrow = nrow(temp.pvals))
colnames(temp.pvals.FWER) = paste(colnames(temp.pvals), "Holm", sep = ".")
temp.pvals.BY = matrix(p.adjust(as.vector(temp.pvals), "BY"), nrow = nrow(temp.pvals))
colnames(temp.pvals.BY) = paste(colnames(temp.pvals), "BY", sep = ".")
xlin.scaled.sel.nmf.cpv.pvals = cbind(xlin.scaled.sel.nmf.cpv.pvals, temp.pvals.FWER,
    temp.pvals.BY)
xlin.scaled.sel.nmf.cpv.pvals = xlin.scaled.sel.nmf.cpv.pvals[, order(colnames(xlin.scaled.sel.nmf.cpv.p
xlin.scaled.sel.nmf.cpv.pvals
```

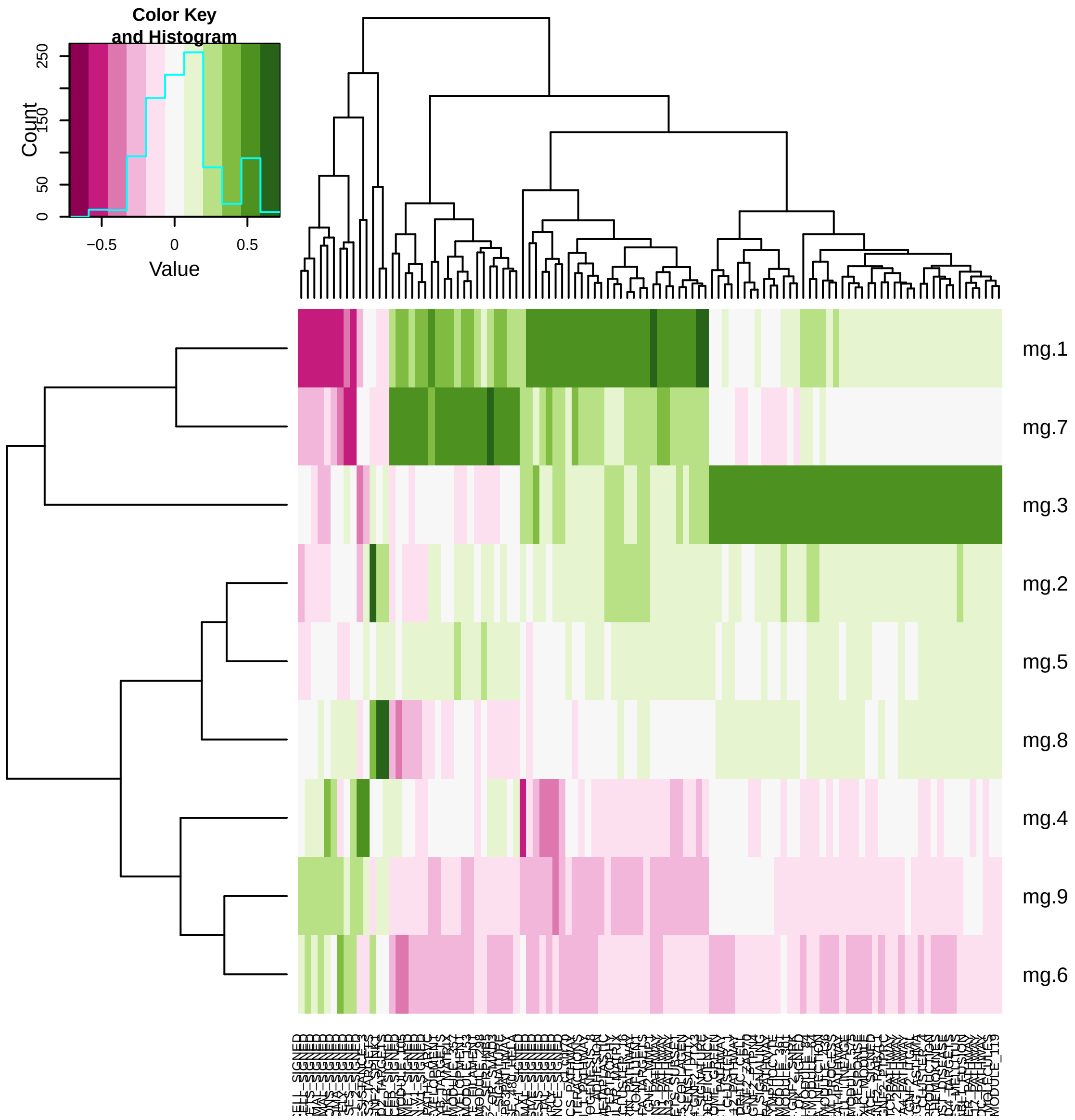
```
##         pure.p pure.p.BY pure.p.Holm  pure.s surv.diag_dsd.c
## mg.1 3.313e-02 3.112e-01   6.957e-01 -2.1305          5.7321
## mg.2 1.248e-04 6.254e-03   4.245e-03 -3.8364          1.3483
## mg.3 6.647e-09 9.990e-07   2.393e-07 -5.7996         -5.8883
## mg.4 3.885e-02 3.244e-01   7.381e-01  2.0658         -3.8704
## mg.5 5.224e-04 1.309e-02   1.620e-02 -3.4690          2.1745
## mg.6 2.001e-01 1.000e+00   1.000e+00  1.2813         -5.9798
## mg.7 4.022e-01 1.000e+00   1.000e+00 -0.8378          7.6656
## mg.8 5.021e-04 1.309e-02   1.607e-02 -3.4796         -0.7869
## mg.9 6.365e-02 5.035e-01   1.000e+00  1.8546         -1.3313
##      surv.diag_dsd.p surv.diag_dsd.p.BY surv.diag_dsd.p.Holm
## mg.1       7.655e-04          0.0164356            0.0229663
## mg.2       5.572e-01          1.0000000            1.0000000
## mg.3       4.628e-03          0.0695495            0.1249524
## mg.4       2.381e-02          0.2385380            0.5237913
## mg.5       2.602e-01          1.0000000            1.0000000
## mg.6       2.271e-01          1.0000000            1.0000000
## mg.7       9.152e-06          0.0006877            0.0003203
## mg.8       5.271e-01          1.0000000            1.0000000
## mg.9       7.211e-01          1.0000000            1.0000000
##      surv.diag_rec.c surv.diag_rec.p surv.diag_rec.p.BY
## mg.1           4.724       0.0054414            0.07434
## mg.2           2.327       0.2815081            1.00000
## mg.3          -3.142       0.0860601            0.64667
## mg.4          -1.530       0.3075327            1.00000
## mg.5           5.652       0.0035300            0.05894
## mg.6          -3.706       0.3983063            1.00000
## mg.7           6.096       0.0003216            0.01208
## mg.8          -1.632       0.1795556            1.00000
## mg.9          -8.059       0.0352772            0.31186
##      surv.diag_rec.p.Holm surv.recr_dsd.c surv.recr_dsd.p
## mg.1              0.14148          4.0330        0.020155
## mg.2              1.00000          0.3709        0.877312
## mg.3              1.00000         -8.1461        0.001041
## mg.4              1.00000         -3.0076        0.105463
## mg.5              0.09884         -2.9819        0.153580
## mg.6              1.00000         -3.5325        0.526554
## mg.7              0.01061          5.1694        0.006184
## mg.8              1.00000          0.7214        0.595577
## mg.9              0.70554         11.0358        0.007390
```

	pure.p.Holm	pure.s	surv.diag_dsd.c	surv.diag_dsd.p.Holm	surv.diag_rec.c	surv.diag_rec.p.Holm	surv.recr_dsd.c	surv.recr_dsd.p.Holm
mg.1	0.6957	-2.1305	5.7321	0.0230	4.724	0.1415	4.0330	0.4636
mg.2	0.0042	-3.8364	1.3483	1.0000	2.327	1.0000	0.3709	1.0000
mg.3	0.0000	-5.7996	-5.8883	0.1250	-3.142	1.0000	-8.1461	0.0302
mg.4	0.7381	2.0658	-3.8704	0.5238	-1.530	1.0000	-3.0076	1.0000
mg.5	0.0162	-3.4690	2.1745	1.0000	5.652	0.0988	-2.9819	1.0000
mg.6	1.0000	1.2813	-5.9798	1.0000	-3.706	1.0000	-3.5325	1.0000
mg.7	1.0000	-0.8378	7.6656	0.0003	6.096	0.0106	5.1694	0.1546
mg.8	0.0161	-3.4796	-0.7869	1.0000	-1.632	1.0000	0.7214	1.0000
mg.9	1.0000	1.8546	-1.3313	1.0000	-8.059	0.7055	11.0358	0.1774

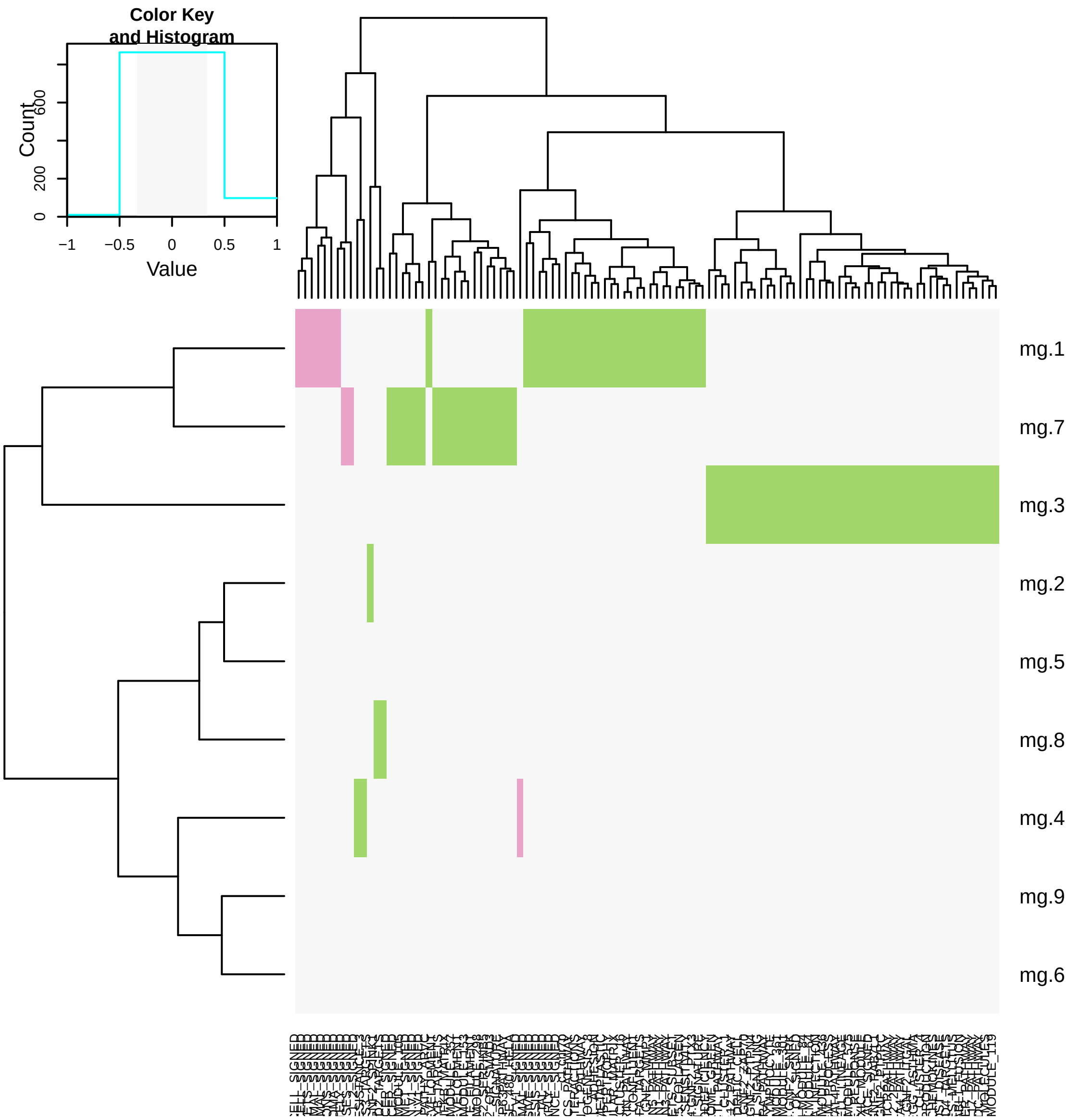
```
##      surv.recr_dsd.p.BY surv.recr_dsd.p.Holm
## mg.1            0.21636              0.46357
## mg.2            1.00000              1.00000
## mg.3            0.01955              0.03018
## mg.4            0.75474              1.00000
## mg.5            1.00000              1.00000
## mg.6            1.00000              1.00000
## mg.7            0.07744              0.15460
## mg.8            1.00000              1.00000
## mg.9            0.08543              0.17736
```

4.4 MSigDB score correlation thresholding

```
temp.sel_cols = apply(abs(xlin.scaled.sel.nmf.msigdb.corr) >= sig.corr.threshold,
    2, any)
heatmap.2(xlin.scaled.sel.nmf.msigdb.corr[, temp.sel_cols], trace = "none",
    scale = "none", useRaster = TRUE, col = brewer.pal(11, "PiYG"), symbreaks = TRUE)
```




```
heatmap.2(xlin.scaled.sel.nmf.msigdb.corr[, temp.sel_cols], trace = "none",
    scale = "none", useRaster = TRUE, col = brewer.pal(3, "PiYG"), breaks = c(-1,
        -sig.corr.threshold, sig.corr.threshold, 1))
```




```r
temp.sig_id = colnames(xlin.scaled.sel.nmf.msigdb.corr)
temp.sig_class = gsub("\\..*", "", temp.sig_id)
temp.nsigs = length(temp.sig_id)
temp.nmeta = nrow(xlin.scaled.sel.nmf.msigdb.corr)
tables = lapply(1:temp.nmeta, function(metagene_i) {
    tapply(1:temp.nsigs, temp.sig_class, function(sig_class_is) {
        all_cors = xlin.scaled.sel.nmf.msigdb.corr[, sig_class_is]
        this_cors = all_cors[metagene_i, ]
        this_ids = temp.sig_id[sig_class_is]

        all_sig_cors = abs(all_cors) >= sig.corr.threshold
        this_sig_cors = all_sig_cors[metagene_i, ]

        sigs_to_report = which(this_sig_cors)

        if (length(sigs_to_report) == 0) {
            table = data.frame(GeneSet = c(), Correlation = c(), Metagenes = c())
```

```
        } else {
            table = data.frame(GeneSet = this_ids[sigs_to_report], Correlation = this_cors[sigs_to_repor
                Metagenes = apply(all_cors[, sigs_to_report, drop = FALSE],
                  2, function(cors) {
                    sel = abs(cors) >= sig.corr.threshold
                    # A positive number implies that positive GSVA signal is associated with
                    # worse prognosis
                    paste(which(sel) * sign(cors[which(sel)]) * sign(xlin.scaled.sel.nmf.cpv.pvals$d.sur
                      collapse = ",")
                  }))
            table = table[order(-(table$Correlation)), ]
            rownames(table) <- NULL
        }
        table
    }, simplify = FALSE)
})

## Error in sign(xlin.scaled.sel.nmf.cpv.pvals$d.surv[metagene_i]):  non-numeric argument to
mathematical function

tables

## Error in eval(expr, envir, enclos):  object 'tables' not found
```

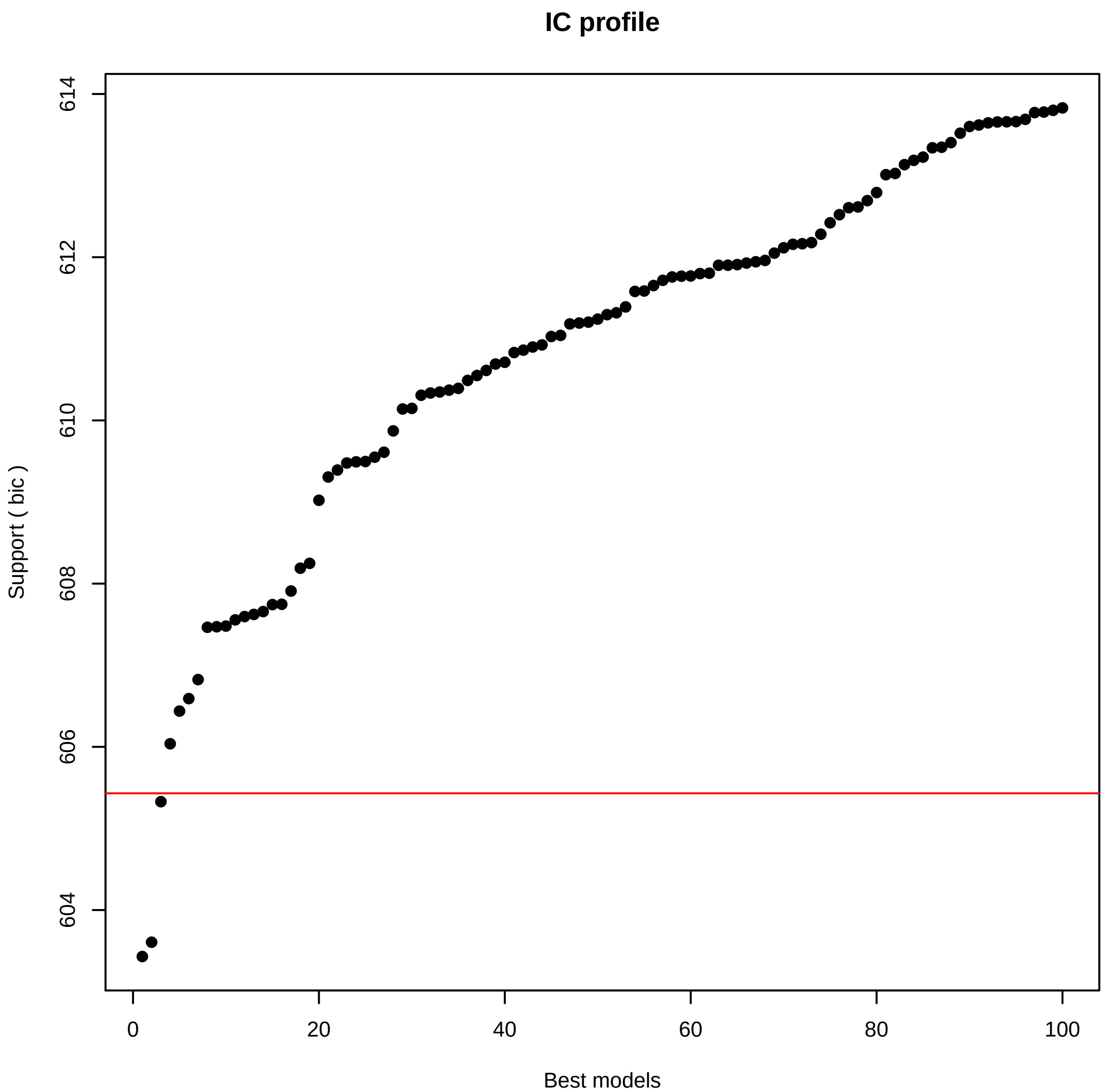
4.4.1 Outcome: Diagnosis to recurrence

```
print(diag_rec.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 100 models:
## Best IC: 603.430717350224
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.5 + mg.7"
## Evidence weight: 0.161659782892363
## Worst IC: 613.829805794669
## 3 models within 2 IC units.
## 65 models to reach 95% of evidence weight.

coef(diag_rec.asreg.result)

##      Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.3  0.03173          0.07202        24    0.09766           0.5324
## mg.8 -0.06076          0.03959        24    0.09852           0.3947
## mg.6 -0.21030          0.60038        25    0.10459           1.5371
## mg.4  0.25643          0.32238        30    0.13979           1.1264
## mg.9 -0.87892          3.12601        31    0.17569           3.5075
## mg.1  1.01497          2.65917        47    0.30088           3.2350
## mg.2  2.57059          8.48483        53    0.52724           5.7785
## mg.5  4.72734          7.02533        64    0.83155           5.2581
## mg.7  6.32776          3.07979        94    0.99017           3.4814
```

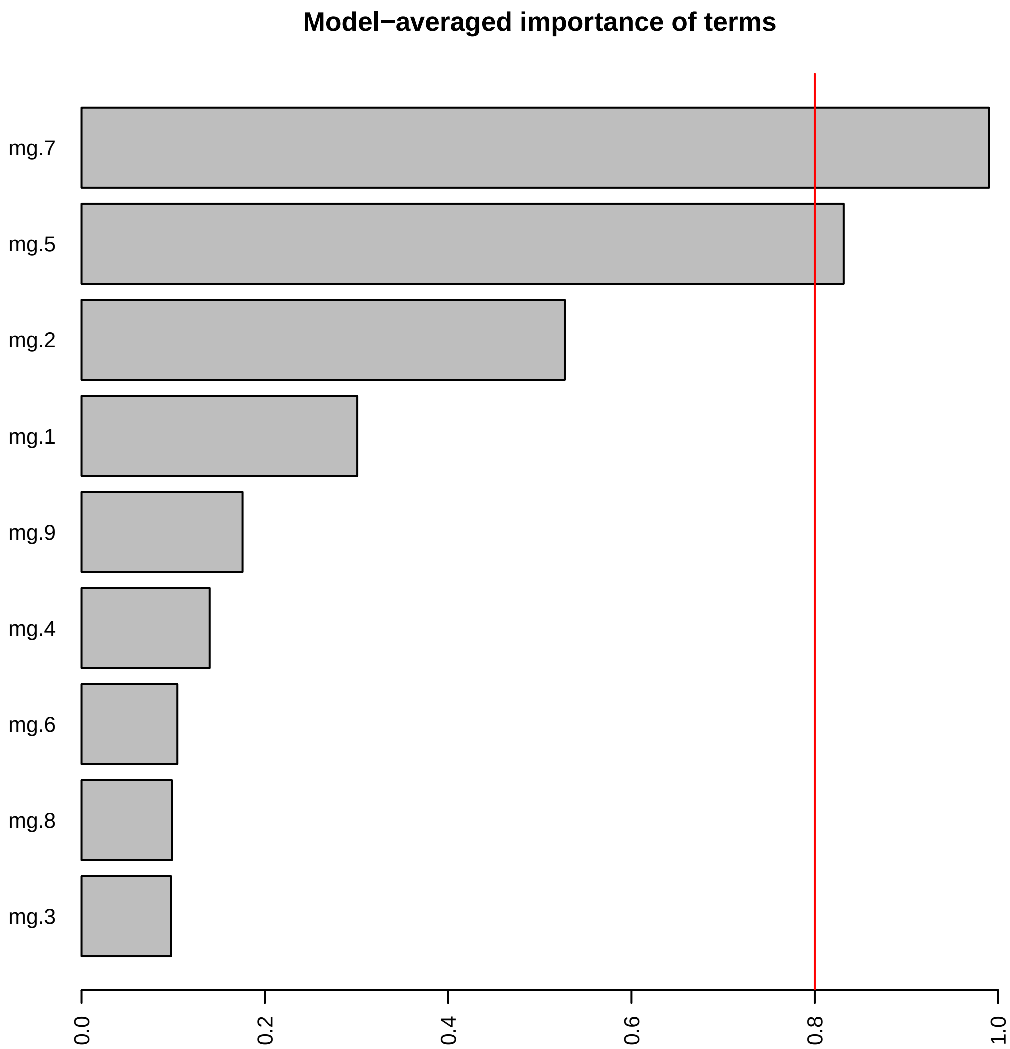
```
summary(diag_rec.asreg.result@objects[[1]])

## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 104, number of events= 77
##
##         coef exp(coef) se(coef)    z Pr(>|z|)
## mg.5   5.76    317.17     1.90 3.04   0.0024
## mg.7   6.09    441.53     1.52 4.02  5.9e-05
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.5       317    0.00315      7.71     13049
## mg.7       442    0.00226     22.60      8625
##
## Concordance= 0.676  (se = 0.036 )
## Rsquare= 0.185   (max possible= 0.997 )
## Likelihood ratio test= 21.3  on 2 df,   p=2.33e-05
## Wald test            = 24.6  on 2 df,   p=4.59e-06
## Score (logrank) test = 25.5  on 2 df,   p=2.86e-06
```

```
plot(diag_rec.asreg.result, type = "p")
```

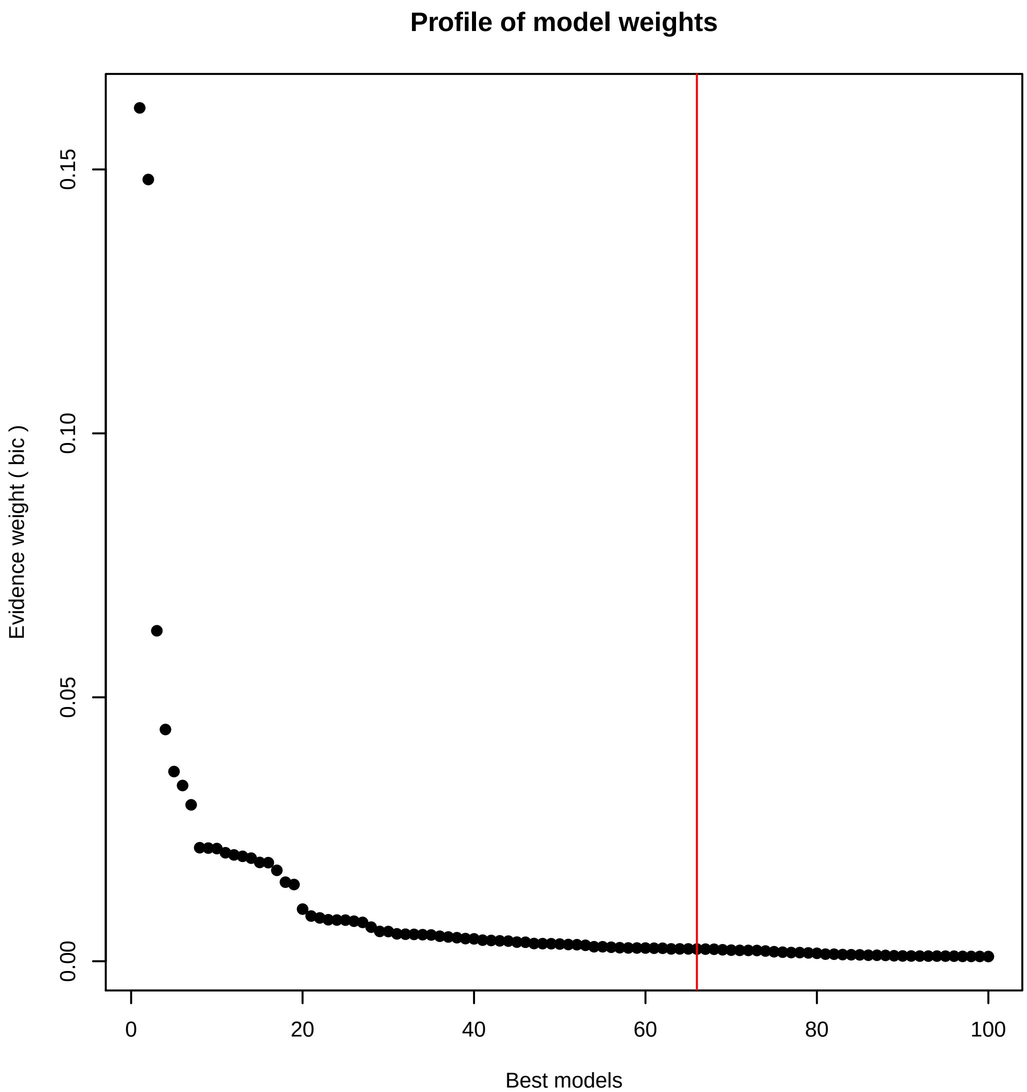
All-subsets regression




```
plot(diag_rec.asreg.result, type = "s")
```




```
plot(diag_rec.asreg.result, type = "w")
```




```
diag_rec.glmnet.coef.1se
```

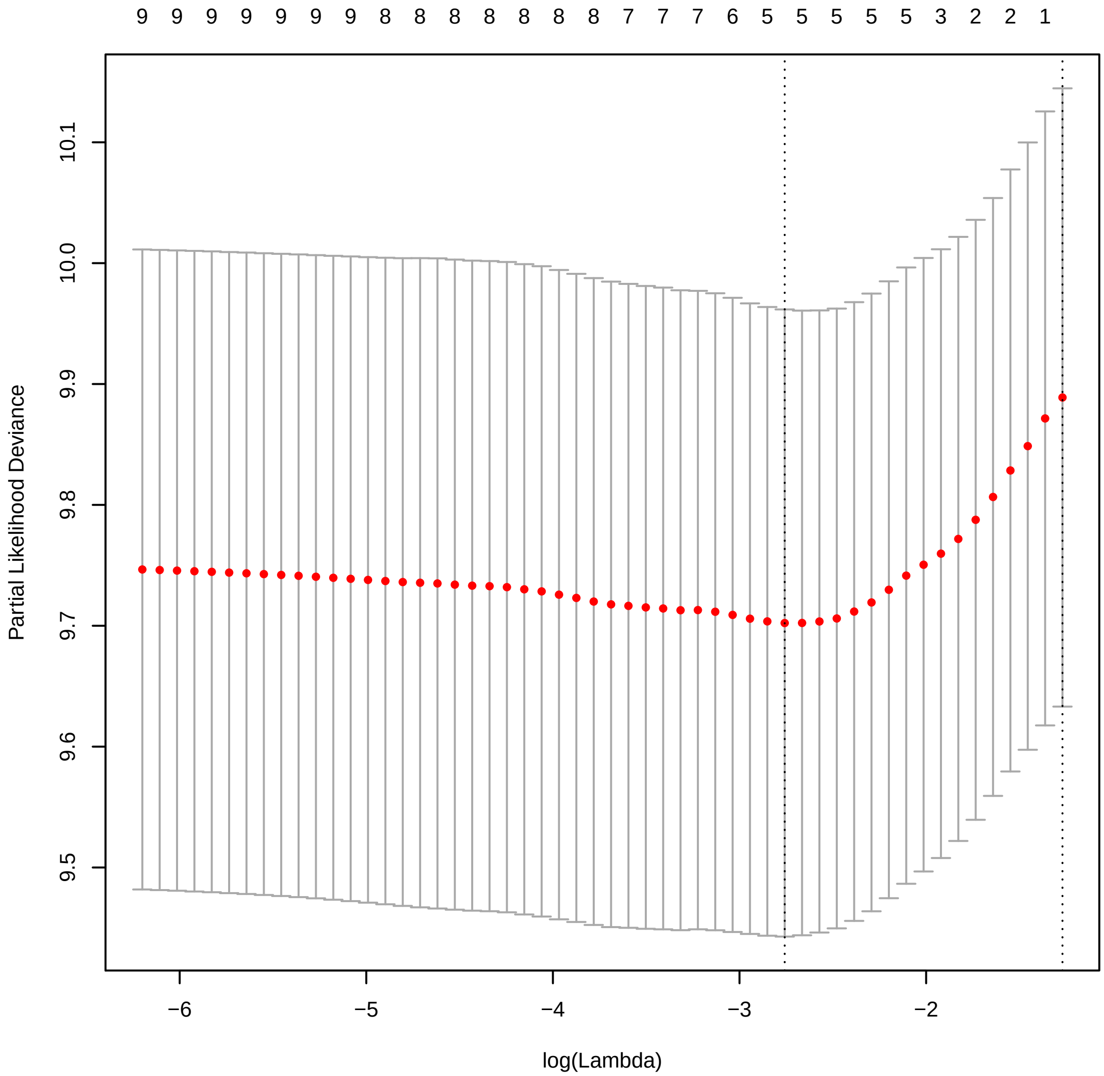
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##      1
## mg.1 .
## mg.2 .
## mg.3 .
## mg.4 .
## mg.5 .
## mg.6 .
## mg.7 .
## mg.8 .
## mg.9 .
```

```
diag_rec.glmnet.coef.min
```

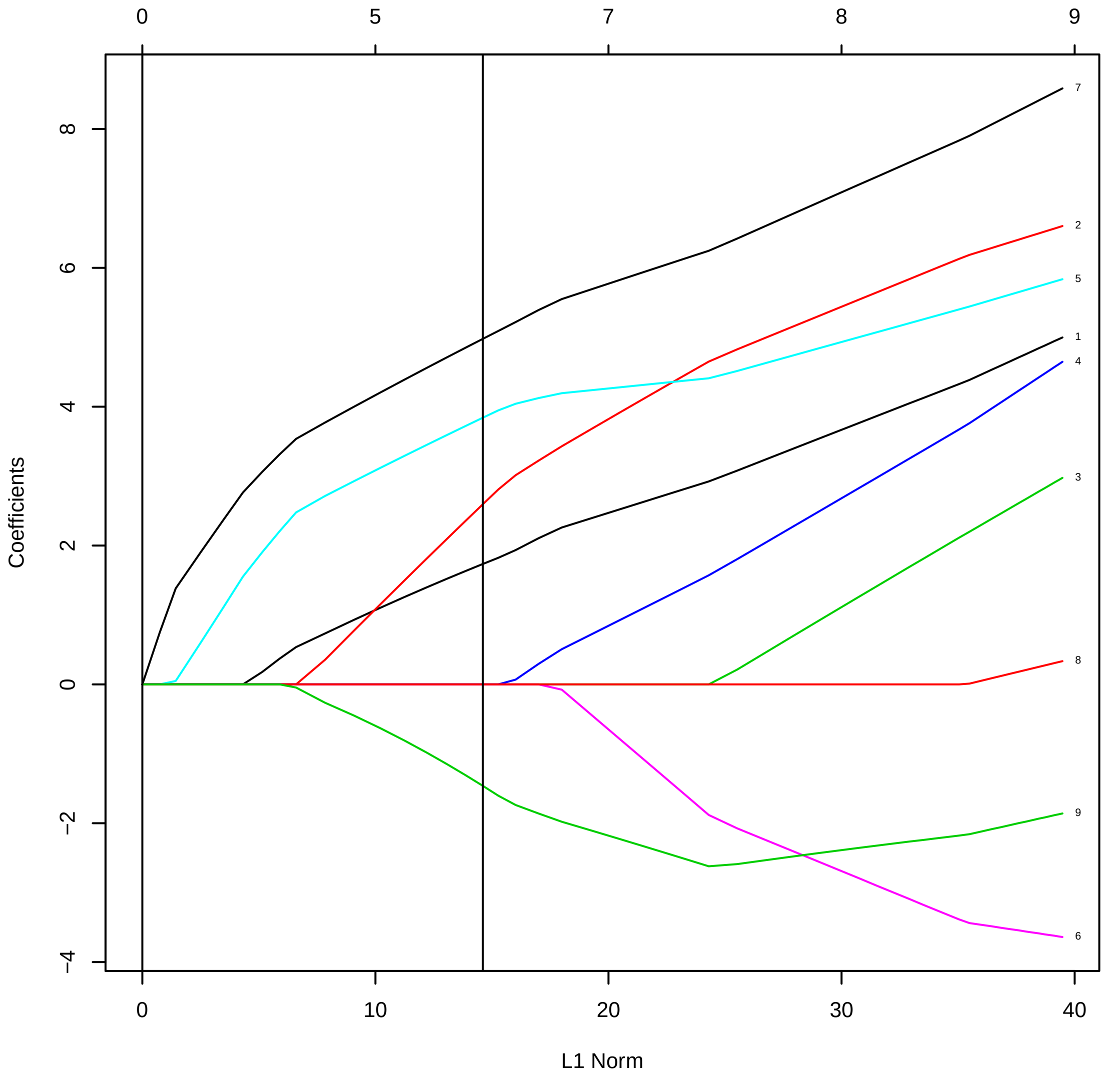
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##           1
## mg.1  1.734
## mg.2  2.593
## mg.3  .
## mg.4  .
## mg.5  3.840
## mg.6  .
## mg.7  4.976
## mg.8  .
## mg.9 -1.460
```

```
plot(diag_rec.glmnet.fit.cv)
```

LASSO




```
plot(diag_rec.glmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_rec.glmnet.coef.1se)))
abline(v = sum(abs(diag_rec.glmnet.coef.min)))
```



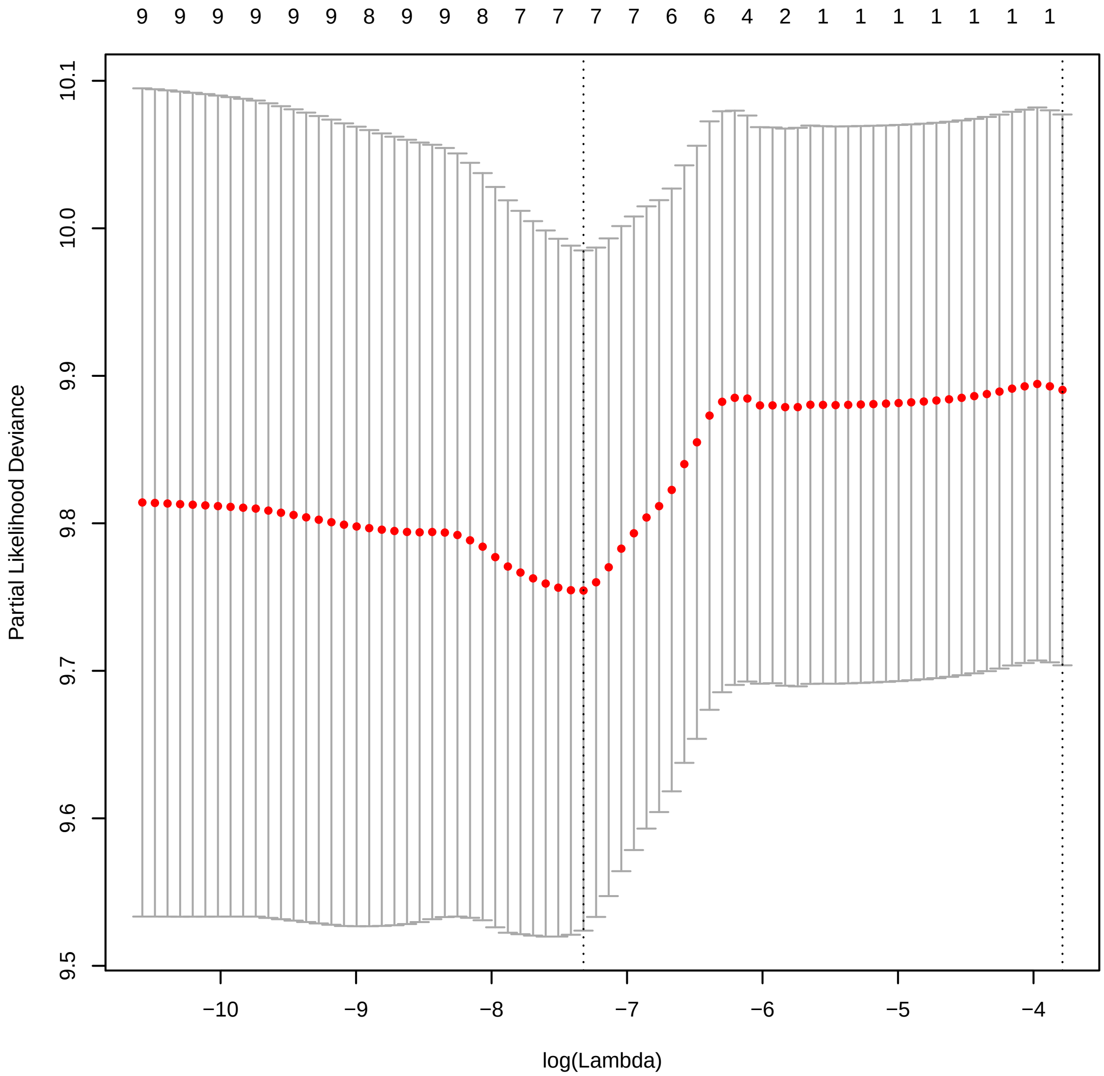

```
diag_rec.adaglmnet.coef.1se/diag_rec.adaglmnet.weights

## 9 x 1 sparse Matrix of class "dgCMatrix"
##      1
## mg.1 .
## mg.2 .
## mg.3 .
## mg.4 .
## mg.5 .
## mg.6 .
## mg.7 .
## mg.8 .
## mg.9 .
```

```r
diag_rec.adaglmnet.coef.min/diag_rec.adaglmnet.weights
```

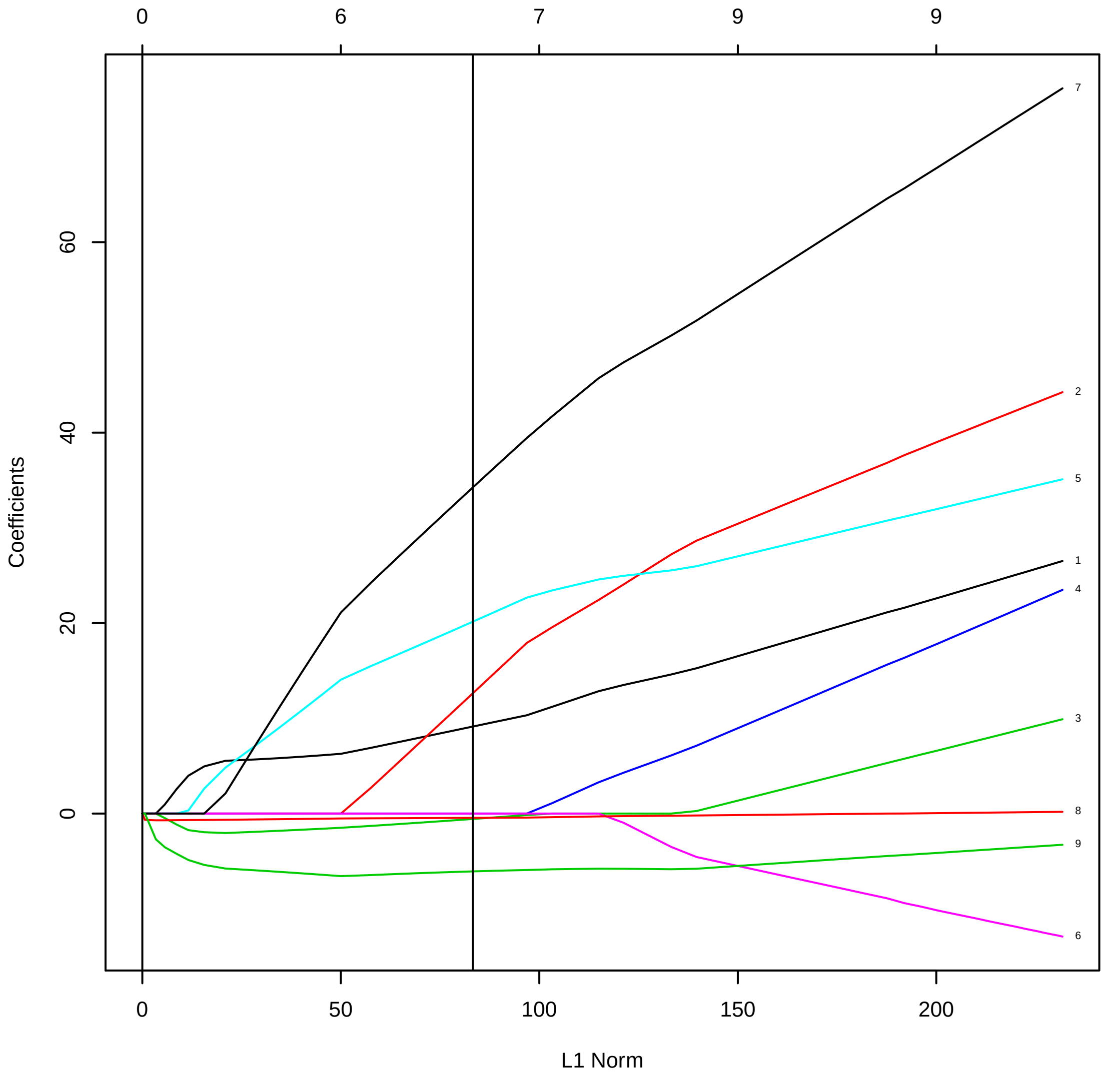
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##              1
## mg.1  48.3081
## mg.2  85.6695
## mg.3  -1.8912
## mg.4   .
## mg.5 121.2622
## mg.6   .
## mg.7 304.4211
## mg.8  -0.2195
## mg.9 -10.5145
```

```r
plot(diag_rec.adaglmnet.fit.cv)
```

Adaptive LASSO




```r
plot(diag_rec.adaglmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_rec.adaglmnet.coef.1se)))
abline(v = sum(abs(diag_rec.adaglmnet.coef.min)))
```



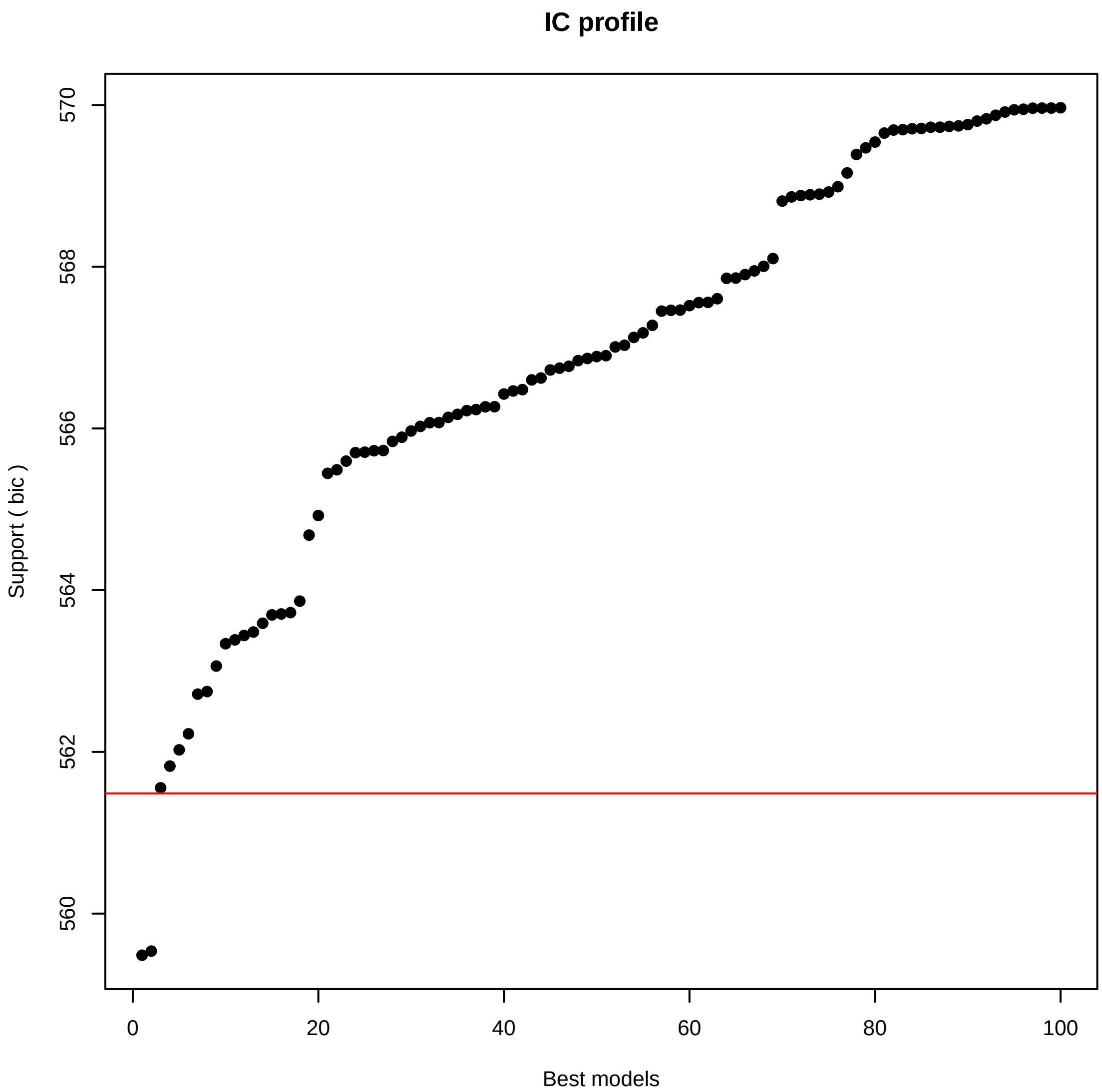
4.4.2 Outcome: Diagnosis to disease-specific death

```
print(diag_dsd.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 100 models:
## Best IC: 559.485656664501
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.1 + mg.2 + mg.7"
## Evidence weight: 0.149970961065674
## Worst IC: 569.966412047098
## 2 models within 2 IC units.
## 60 models to reach 95% of evidence weight.
```

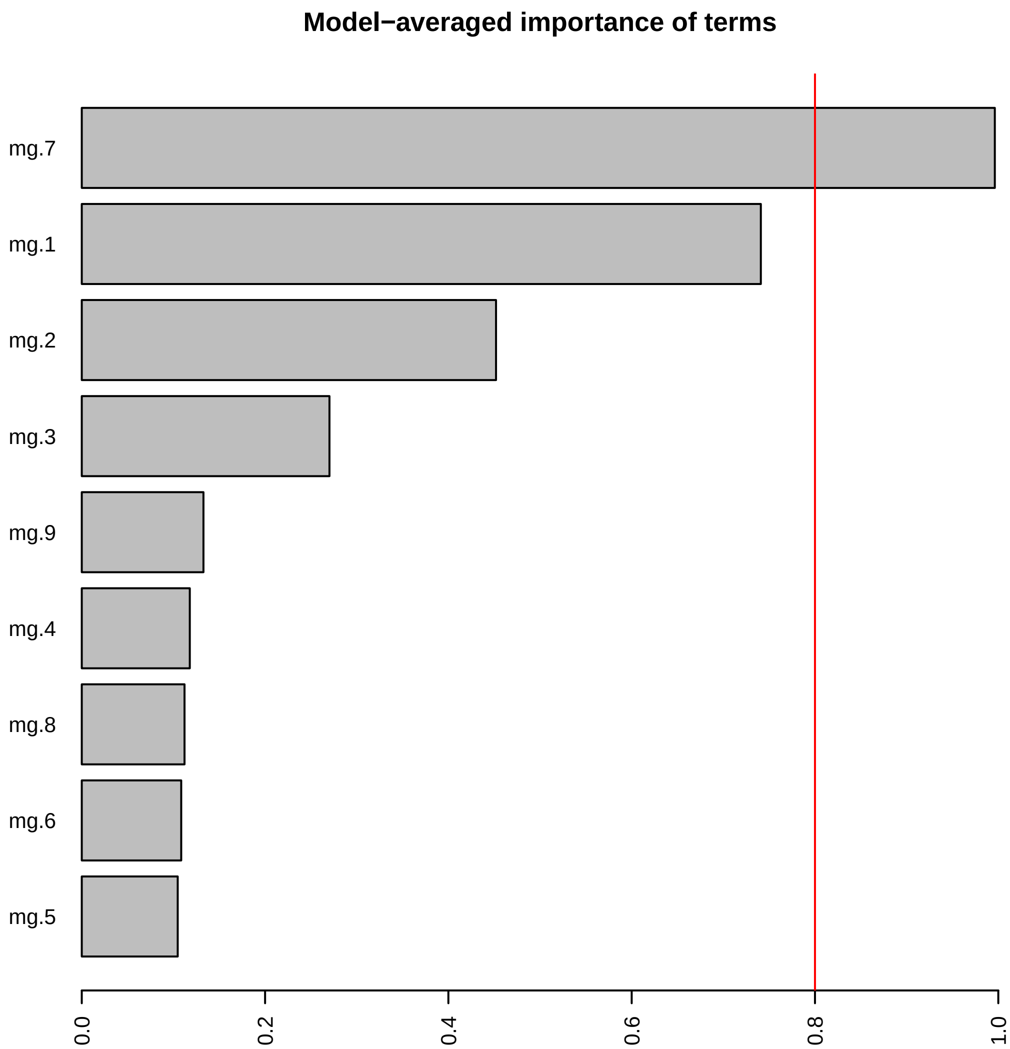
```r
coef(diag_dsd.asreg.result)

##      Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.5  0.09902           0.1082        25     0.1047           0.6521
## mg.6 -0.19265           0.6773        29     0.1085           1.6315
## mg.8  0.09325           0.0672        28     0.1121           0.5139
## mg.4 -0.02173           0.1059        30     0.1179           0.6451
## mg.9  0.45378           1.1512        28     0.1328           2.1270
## mg.3 -0.99781           2.9189        43     0.2703           3.3869
## mg.2  2.13079           7.5358        42     0.4520           5.4420
## mg.1  3.53330           6.4554        59     0.7409           5.0367
## mg.7  7.29291           3.2266        97     0.9962           3.5609

summary(diag_dsd.asreg.result@objects[[1]])

## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 110, number of events= 70
##
##         coef exp(coef) se(coef)    z Pr(>|z|)
## mg.1    5.10    164.76     1.73 2.95   0.0032
## mg.2    4.84    125.94     2.31 2.10   0.0361
## mg.7    7.78   2381.19     1.68 4.63  3.6e-06
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.1       165    0.00607      5.56      4886
## mg.2       126    0.00794      1.37     11591
## mg.7      2381    0.00042     88.65     63958
##
## Concordance= 0.699  (se = 0.038 )
## Rsquare= 0.24   (max possible= 0.995 )
## Likelihood ratio test= 30.2  on 3 df,   p=1.22e-06
## Wald test            = 34.5  on 3 df,   p=1.58e-07
## Score (logrank) test = 36.4  on 3 df,   p=6.14e-08
```

```r
plot(diag_dsd.asreg.result, type = "p")
```

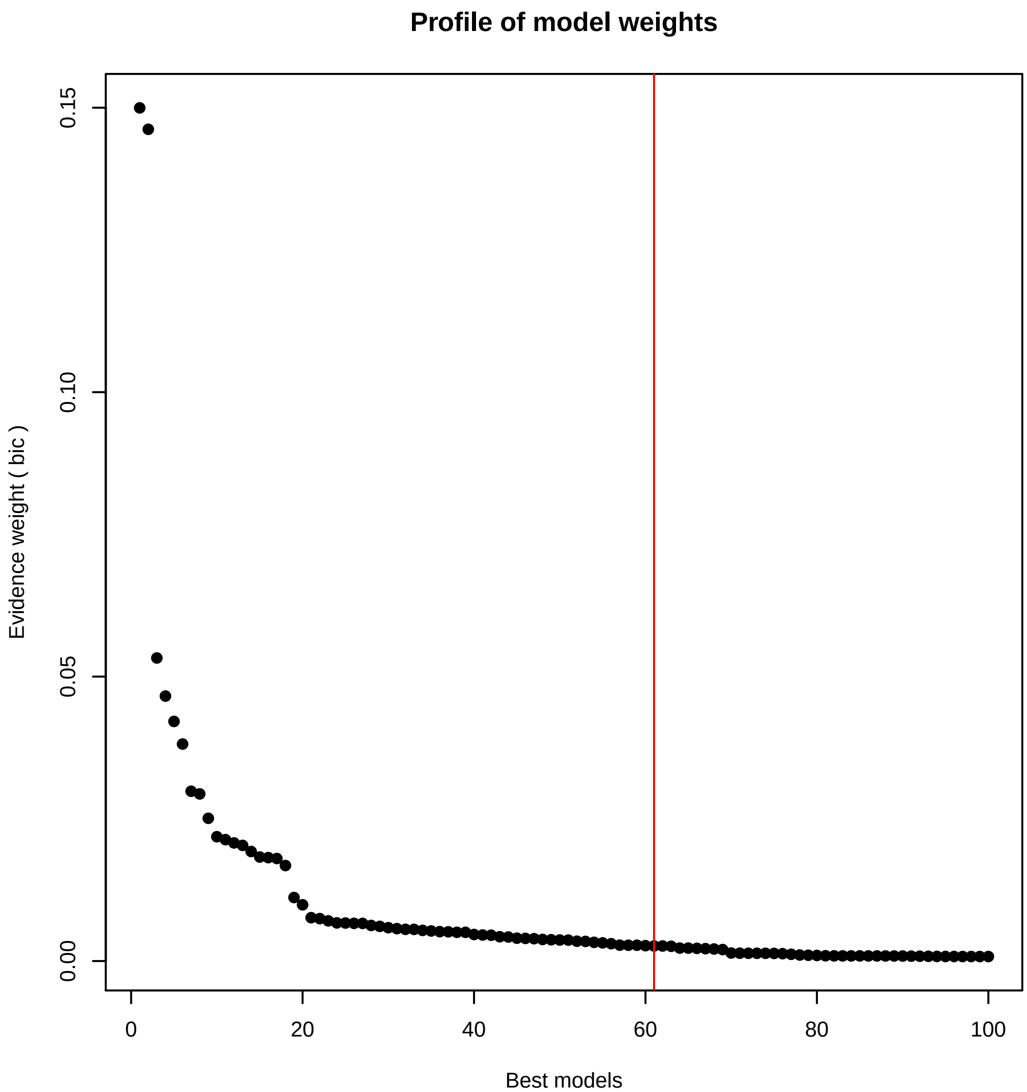
All-subsets regression




```
plot(diag_dsd.asreg.result, type = "s")
```




```r
plot(diag_dsd.asreg.result, type = "w")
```




```
diag_dsd.glmnet.coef.1se
```

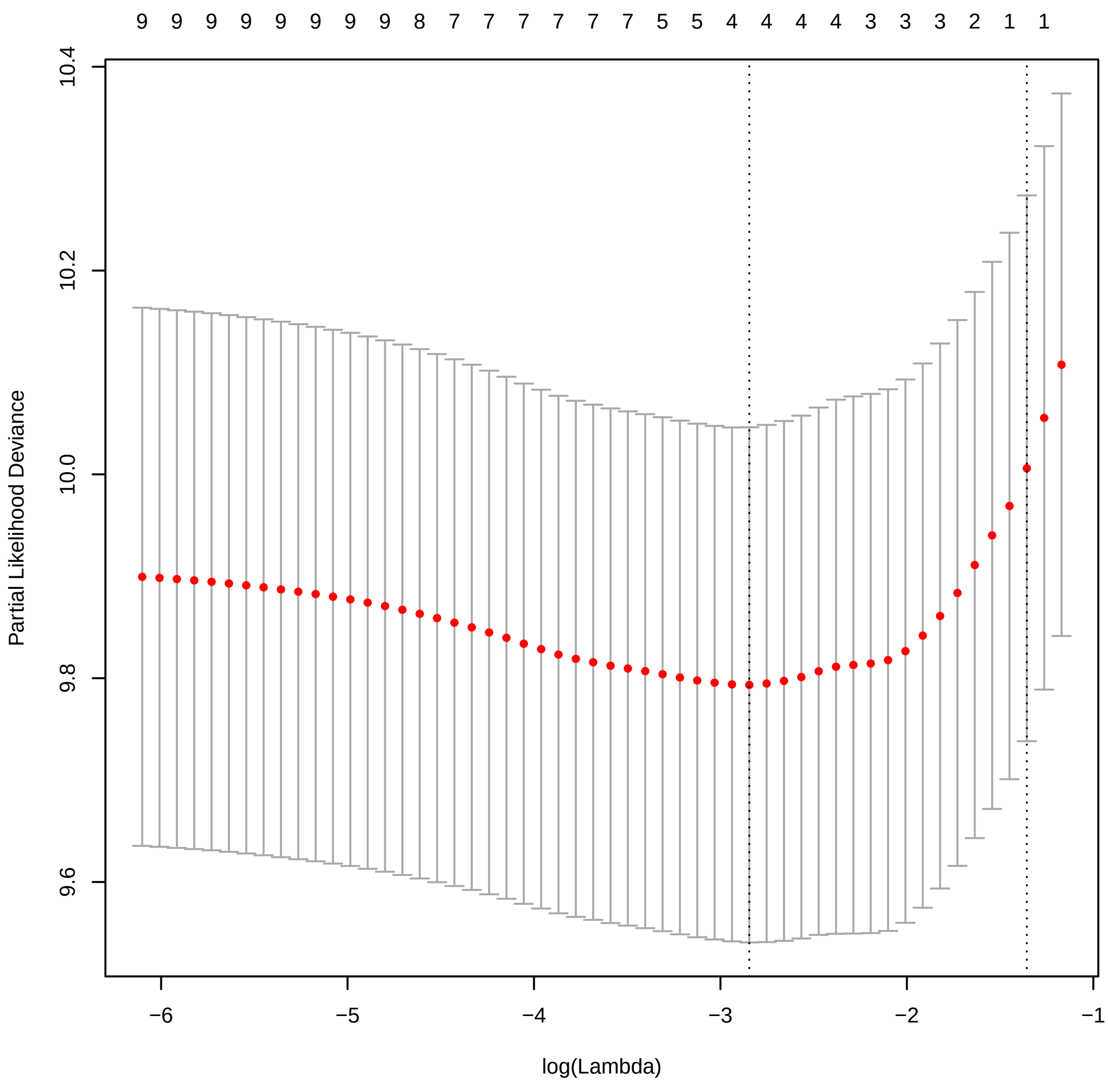
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##          1
## mg.1 .
## mg.2 .
## mg.3 .
## mg.4 .
## mg.5 .
## mg.6 .
## mg.7 1.837
## mg.8 .
## mg.9 .
```

```r
diag_dsd.glmnet.coef.min
```

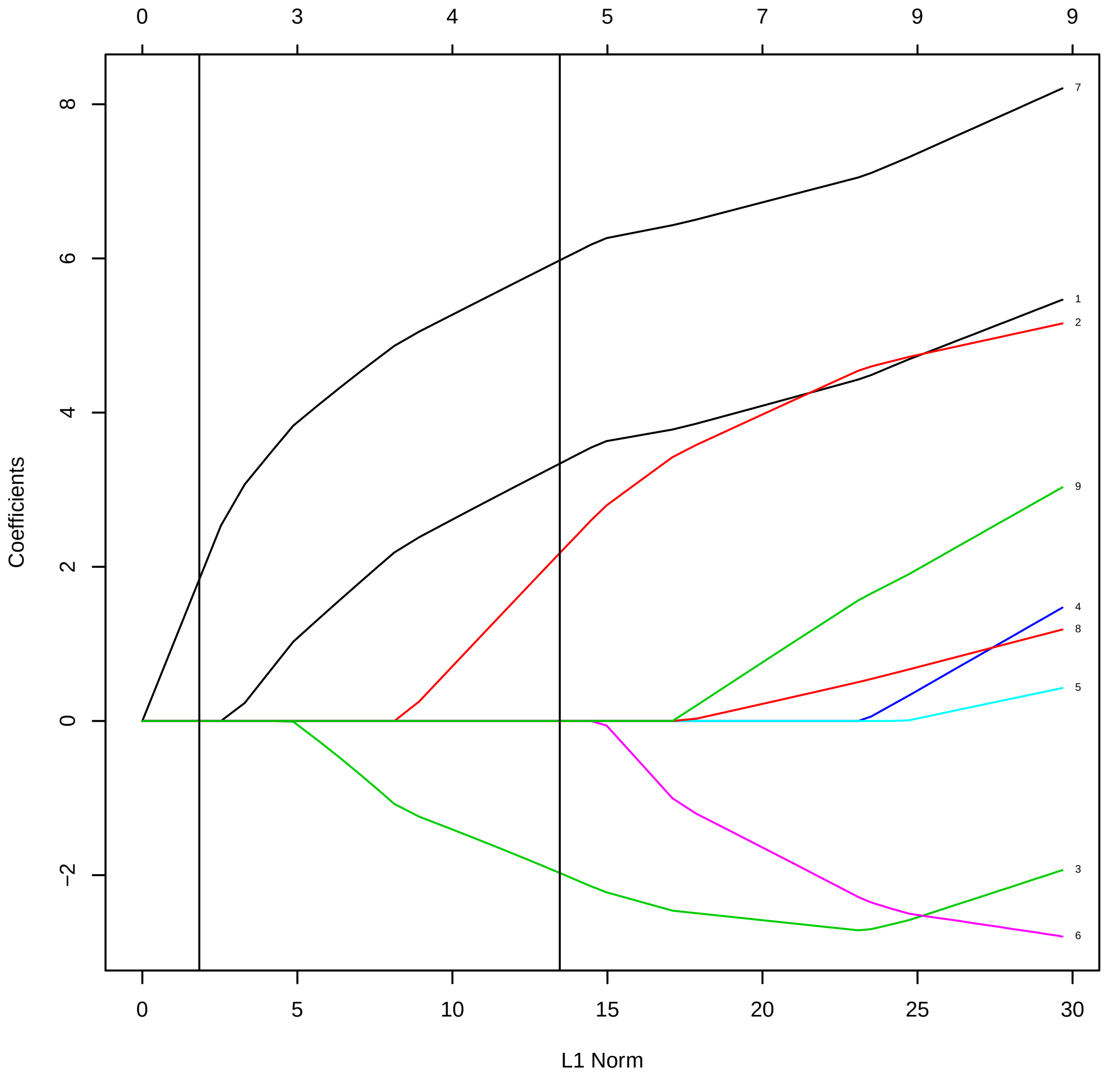
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##            1
## mg.1  3.338
## mg.2  2.178
## mg.3 -1.971
## mg.4  .
## mg.5  .
## mg.6  .
## mg.7  5.975
## mg.8  .
## mg.9  .
```

```r
plot(diag_dsd.glmnet.fit.cv)
```

LASSO




```
plot(diag_dsd.glmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_dsd.glmnet.coef.1se)))
abline(v = sum(abs(diag_dsd.glmnet.coef.min)))
```




```
diag_dsd.adaglmnet.coef.1se/diag_dsd.adaglmnet.weights
```

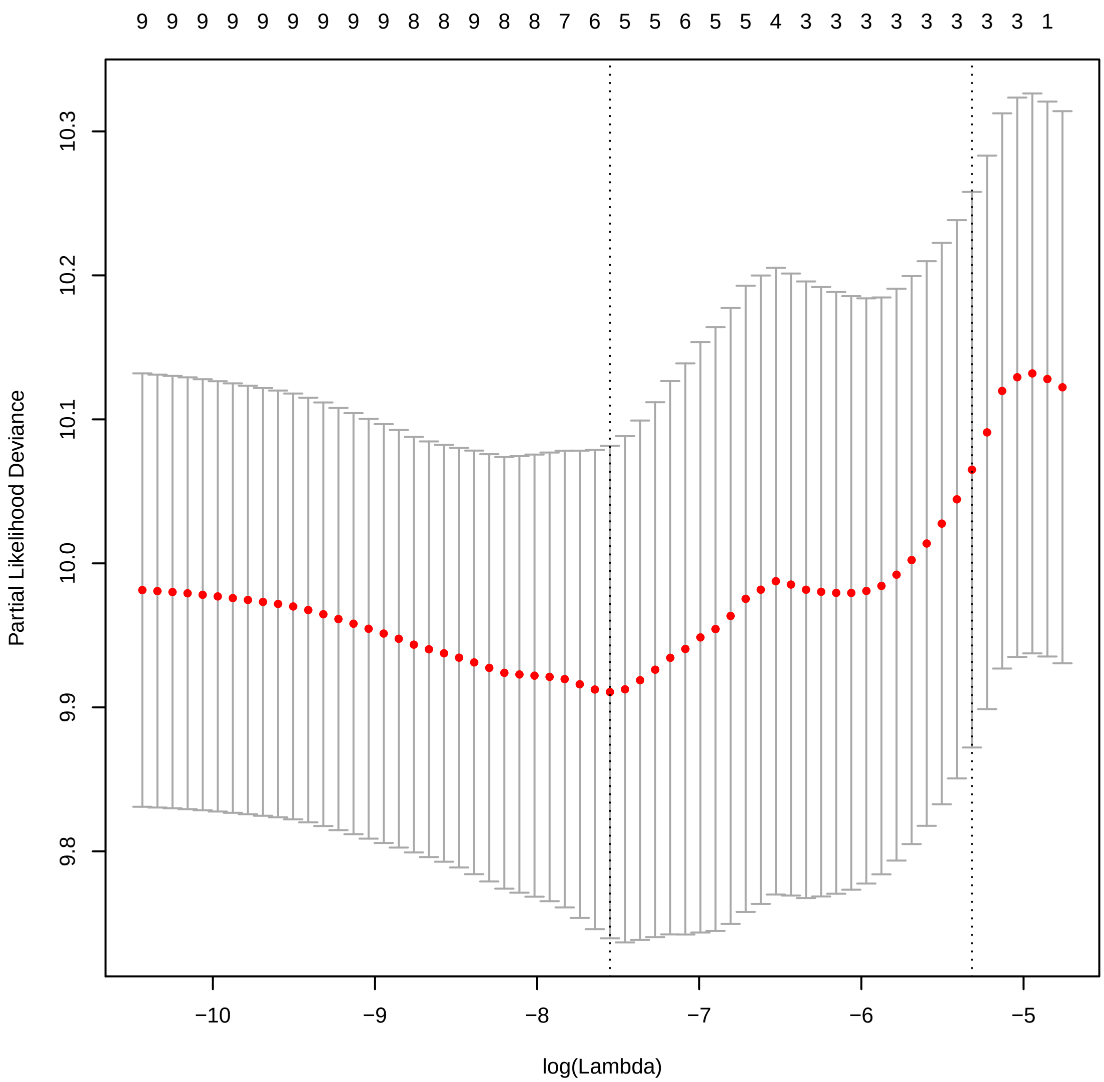
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  .
## mg.2  .
## mg.3 -6.96019
## mg.4 -5.34313
## mg.5  0.02918
## mg.6  .
## mg.7  .
## mg.8  .
## mg.9  .
```

```
diag_dsd.adaglmnet.coef.min/diag_dsd.adaglmnet.weights
```

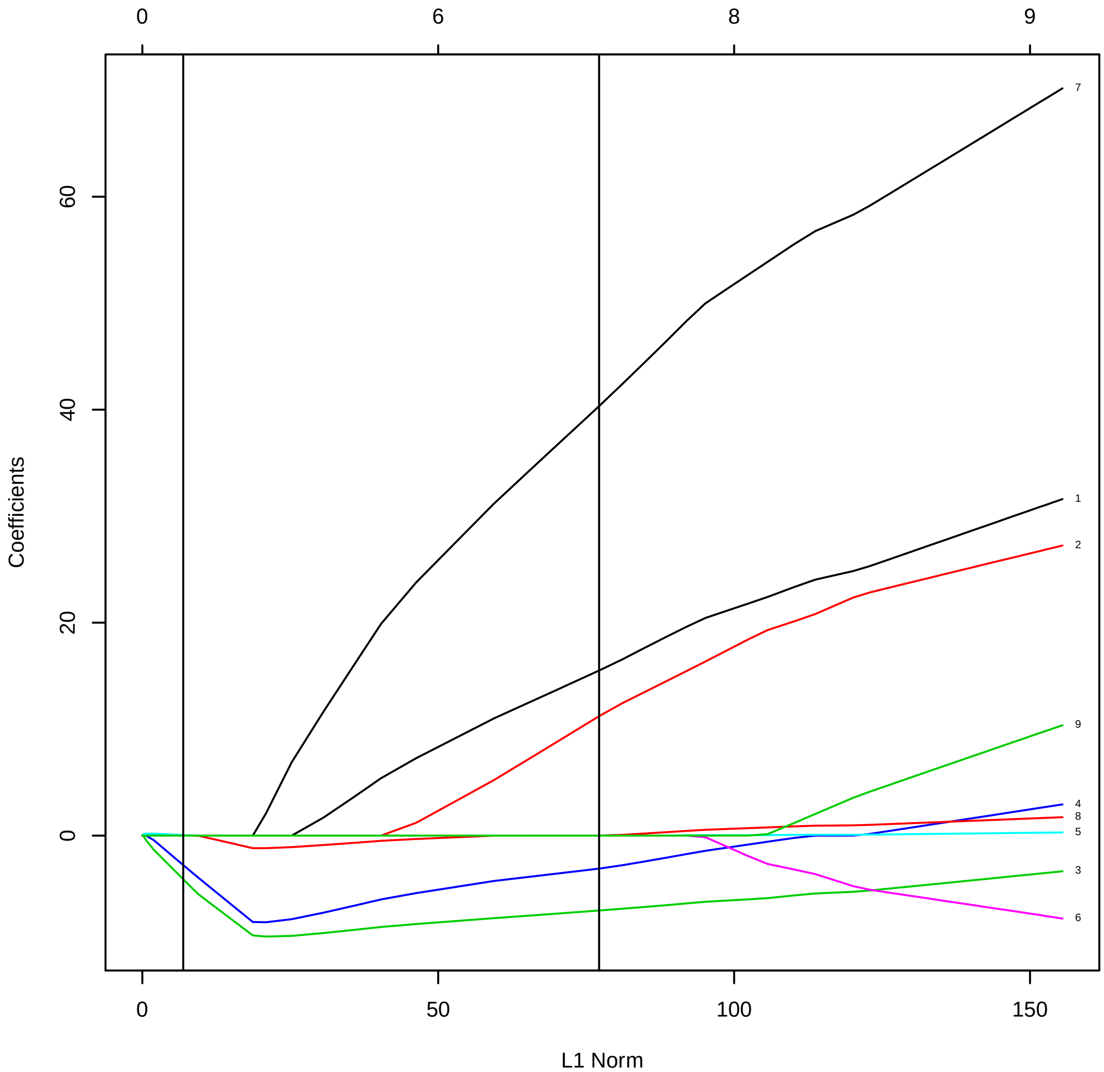
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  89.585
## mg.2  59.853
## mg.3 -11.945
## mg.4  -5.986
## mg.5   .
## mg.6   .
## mg.7 345.093
## mg.8   .
## mg.9   .
```

```
plot(diag_dsd.adaglmnet.fit.cv)
```

Adaptive LASSO




```
plot(diag_dsd.adaglmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_dsd.adaglmnet.coef.1se)))
abline(v = sum(abs(diag_dsd.adaglmnet.coef.min)))
```



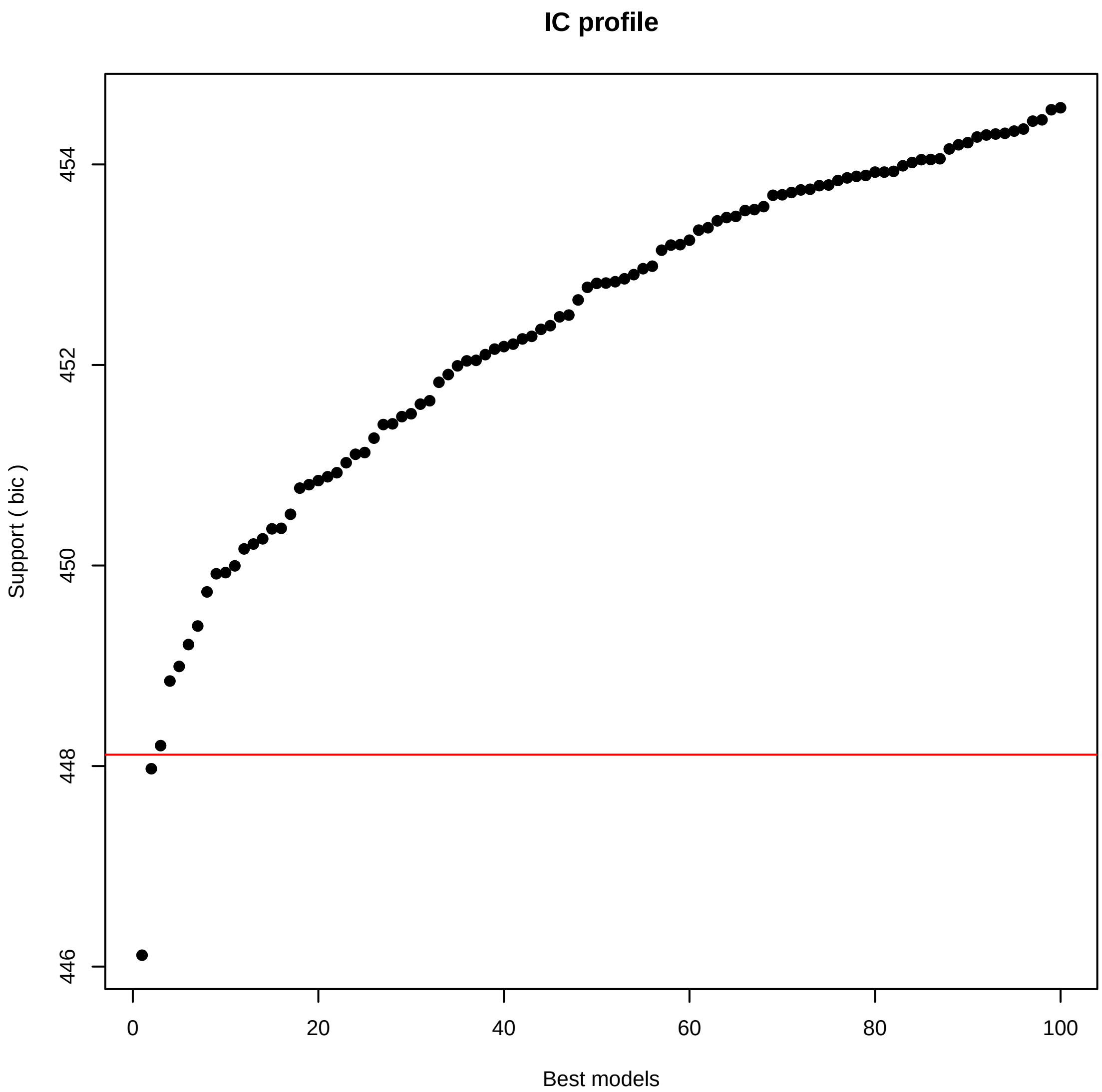
4.4.3 Outcome: Recurrence to disease-specific death

```
print(recr_dsd.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 100 models:
## Best IC: 446.11334502776
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.3 + mg.4 + mg.5"
## Evidence weight: 0.140011204314059
## Worst IC: 454.565555945615
## 2 models within 2 IC units.
## 79 models to reach 95% of evidence weight.
```

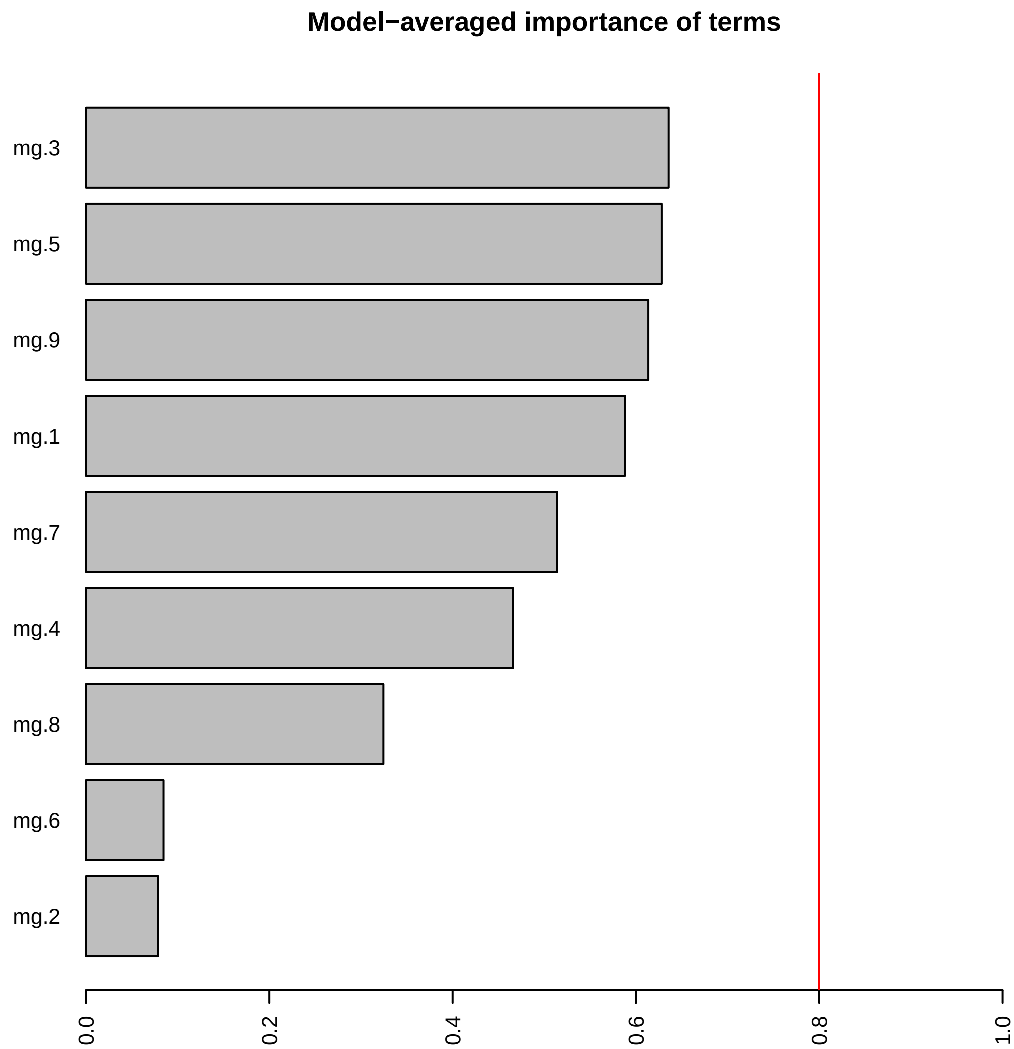
```r
coef(recr_dsd.asreg.result)

##      Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.2 -0.08392          0.08446        18    0.07879           0.5787
## mg.6  0.21108          0.66654        18    0.08456           1.6257
## mg.8  1.07063          2.75008        39    0.32449           3.3021
## mg.4 -1.44030         10.15033        39    0.46586           6.3440
## mg.7  2.75316         10.14446        53    0.51395           6.3421
## mg.1  3.28847         10.89931        62    0.58795           6.5739
## mg.9  7.97119         57.24294        66    0.61349          15.0654
## mg.5 -3.84988         13.63369        63    0.62812           7.3524
## mg.3 -5.61242         26.11881        70    0.63566          10.1765

summary(recr_dsd.asreg.result@objects[[1]])

## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 81, number of events= 64
##
##           coef exp(coef)  se(coef)     z Pr(>|z|)
## mg.3 -1.12e+01  1.36e-05  2.79e+00 -4.02  5.9e-05
## mg.4 -5.99e+00  2.51e-03  2.03e+00 -2.96   0.0031
## mg.5 -7.57e+00  5.13e-04  2.67e+00 -2.84   0.0045
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.3  1.36e-05      73321  5.76e-08   0.00323
## mg.4  2.51e-03        399  4.74e-05   0.13275
## mg.5  5.13e-04       1948  2.77e-06   0.09529
##
## Concordance= 0.682  (se = 0.041 )
## Rsquare= 0.261   (max possible= 0.997 )
## Likelihood ratio test= 24.5  on 3 df,   p=1.99e-05
## Wald test            = 21.1  on 3 df,   p=9.94e-05
## Score (logrank) test = 22.3  on 3 df,   p=5.67e-05
```

```r
plot(recr_dsd.asreg.result, type = "p")
```

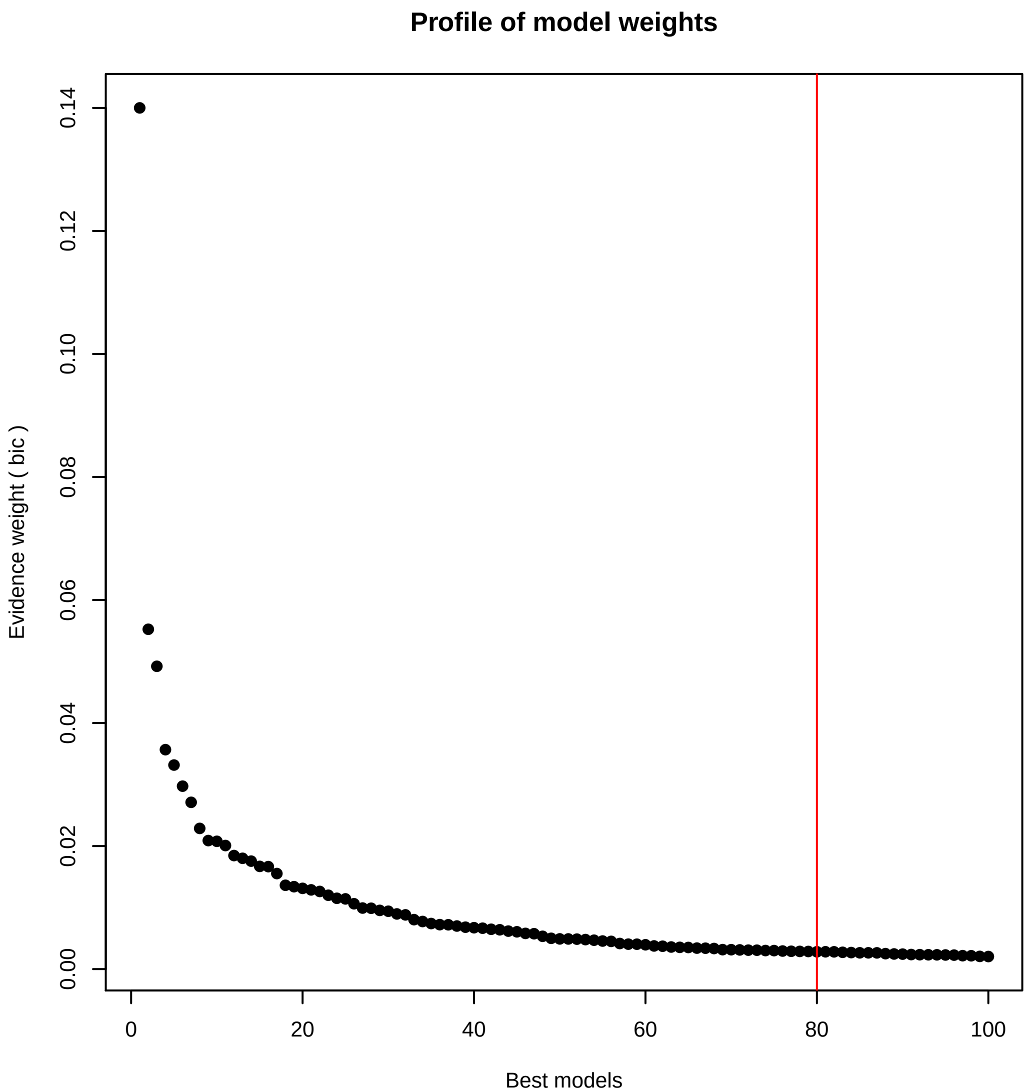
All-subsets regression




```
plot(recr_dsd.asreg.result, type = "s")
```




```
plot(recr_dsd.asreg.result, type = "w")
```




```
recr_dsd.glmnet.coef.1se
```

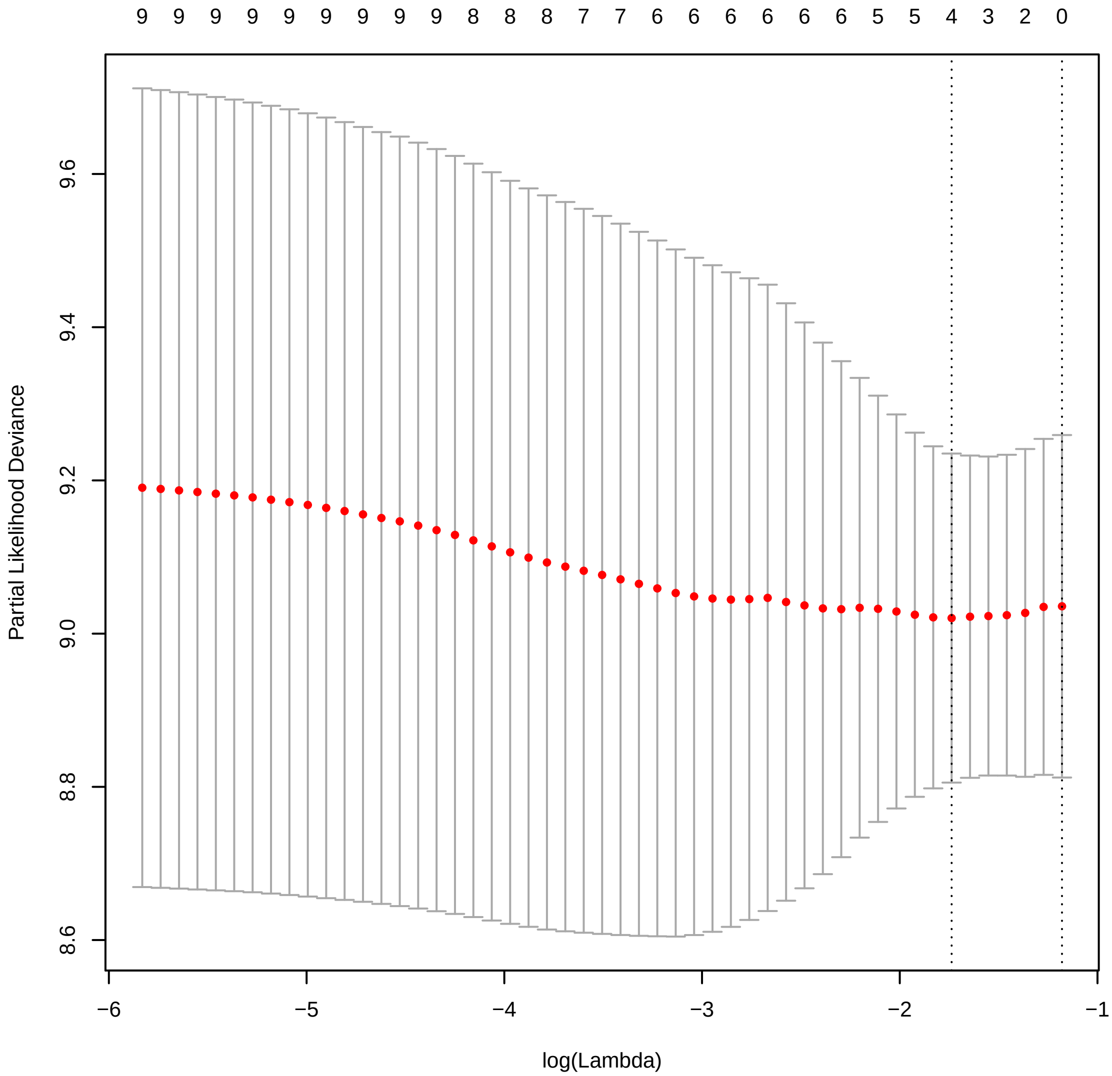
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##      1
## mg.1 .
## mg.2 .
## mg.3 .
## mg.4 .
## mg.5 .
## mg.6 .
## mg.7 .
## mg.8 .
## mg.9 .
```

```r
recr_dsd.glmnet.coef.min
```

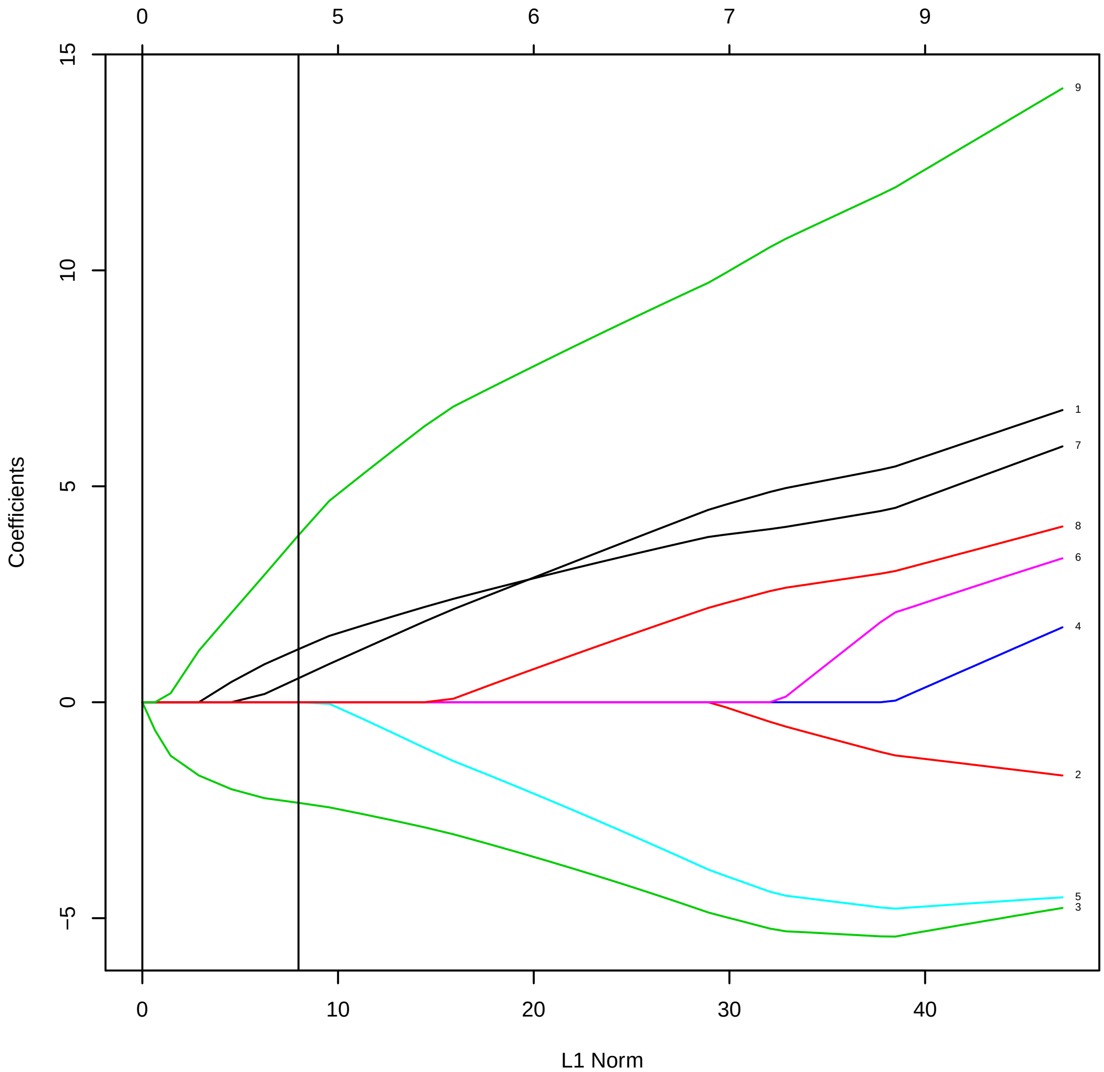
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  0.5558
## mg.2  .
## mg.3 -2.3288
## mg.4  .
## mg.5  .
## mg.6  .
## mg.7  1.2288
## mg.8  .
## mg.9  3.8672
```

```r
plot(recr_dsd.glmnet.fit.cv)
```

LASSO




```
plot(recr_dsd.glmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(recr_dsd.glmnet.coef.1se)))
abline(v = sum(abs(recr_dsd.glmnet.coef.min)))
```




```
recr_dsd.adaglmnet.coef.1se/recr_dsd.adaglmnet.weights
```

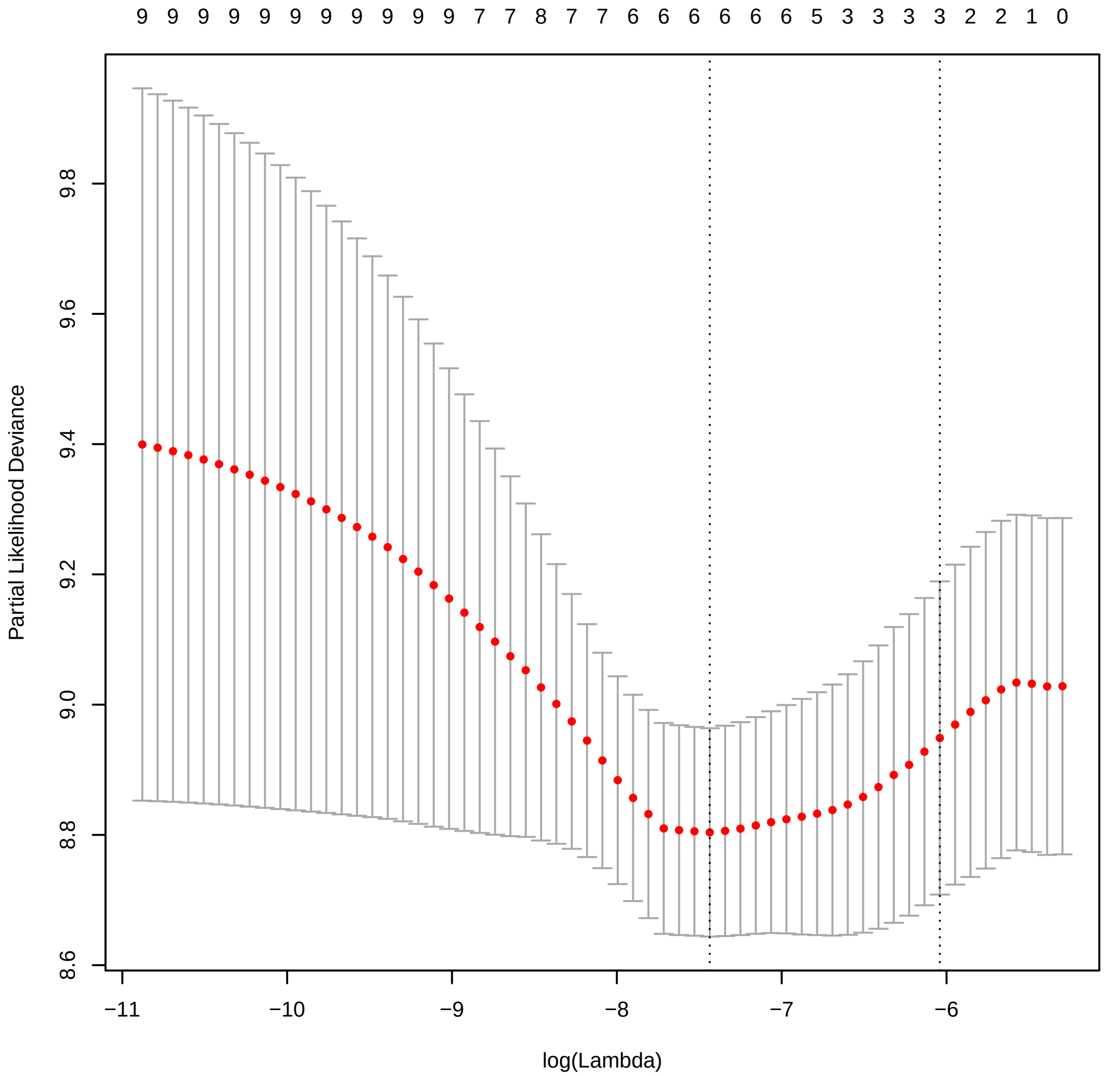
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1    .
## mg.2    .
## mg.3 -67.628
## mg.4  -9.843
## mg.5  -8.898
## mg.6    .
## mg.7    .
## mg.8    .
## mg.9    .
```

```
recr_dsd.adaglmnet.coef.min/recr_dsd.adaglmnet.weights
```

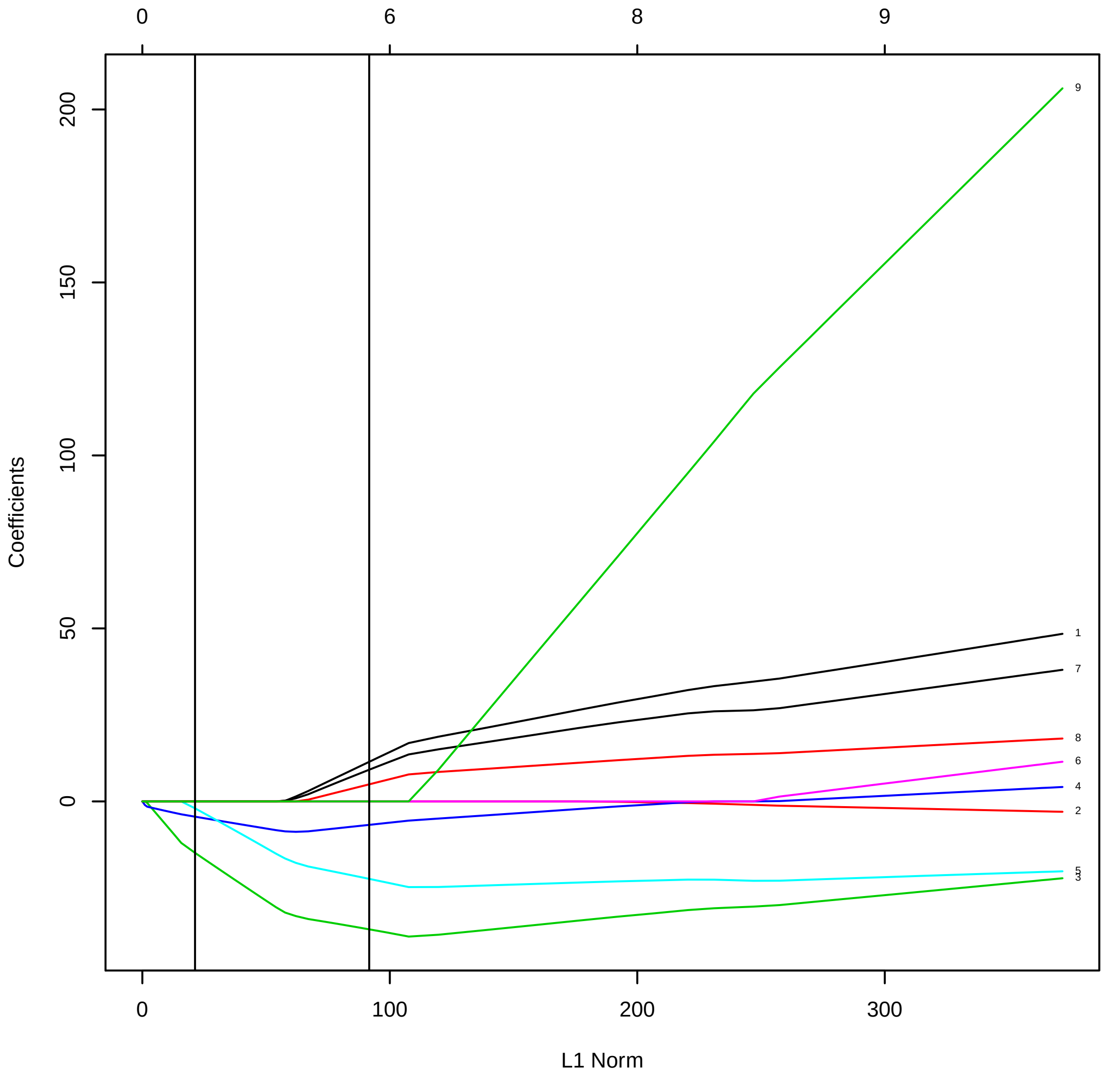
```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1   81.49
## mg.2     .
## mg.3 -168.13
## mg.4  -15.12
## mg.5  -99.44
## mg.6     .
## mg.7   58.09
## mg.8   21.46
## mg.9     .
```

```
plot(recr_dsd.adaglmnet.fit.cv)
```

Adaptive LASSO



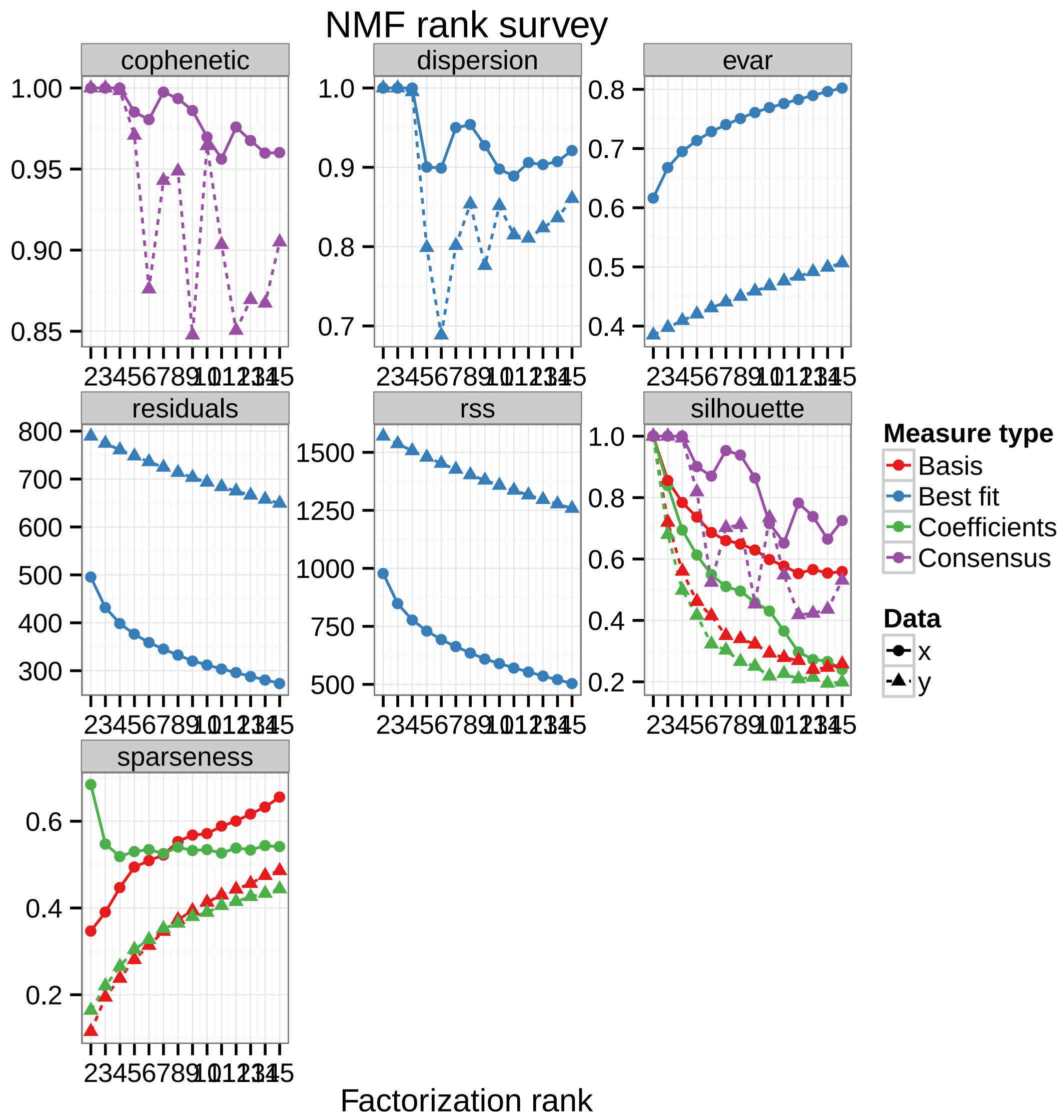

```
plot(recr_dsd.adaglmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(recr_dsd.adaglmnet.coef.1se)))
abline(v = sum(abs(recr_dsd.adaglmnet.coef.min)))
```



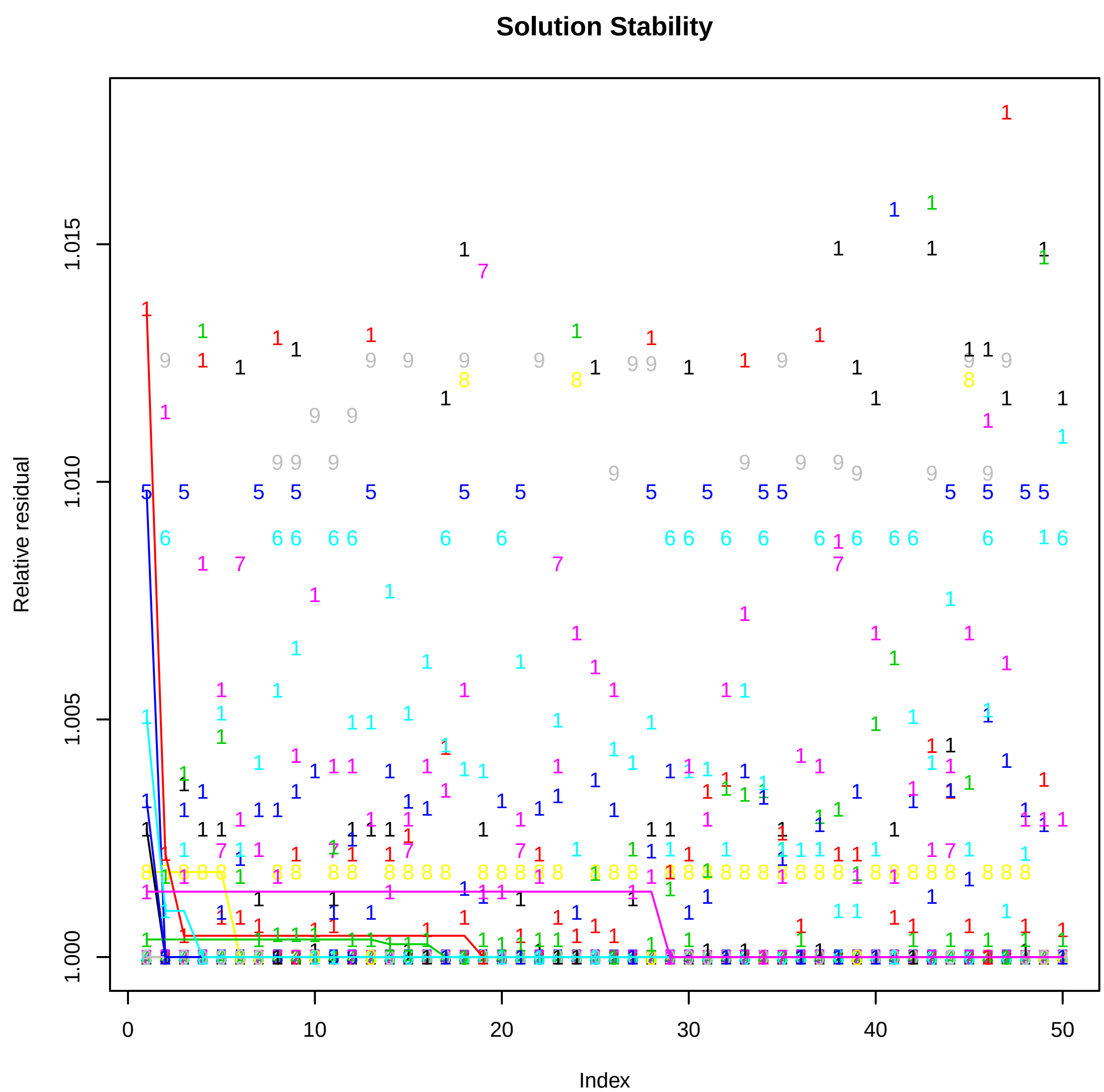

```
load("image-byeye.rda")
nmf.rank.wasauto = FALSE
```

5 Factorization

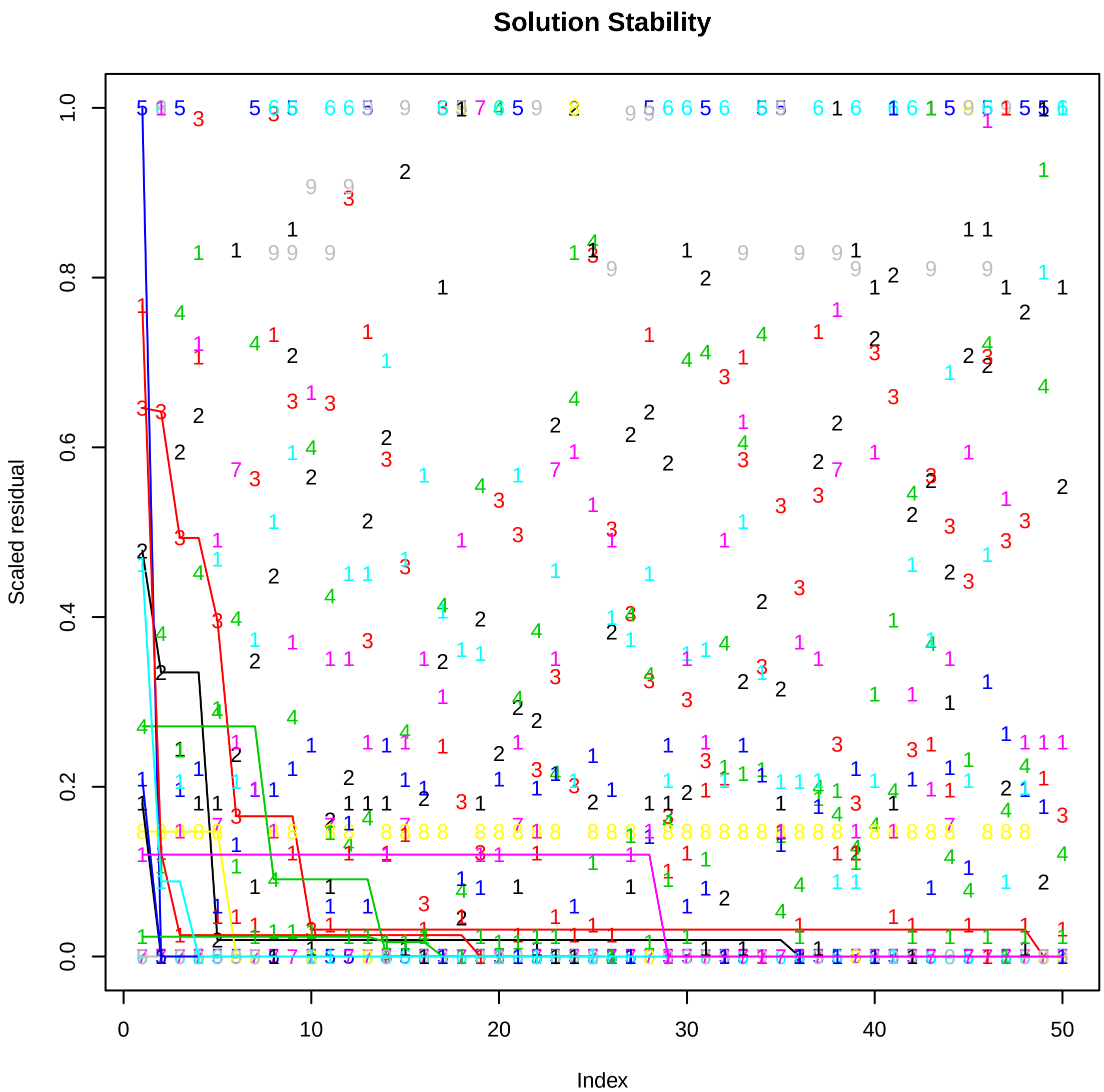
```
plot(temp.nmf.rank, temp.nmf.rank.random[[1]])
```




```
# for (i in 1:length(temp.nmf.rank£fit)) {
# consensusmap(temp.nmf.rank£fit[[i]]) }
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_rel),
    ylab = "Relative residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_rel[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_rel[, i]), col = i)
}
```

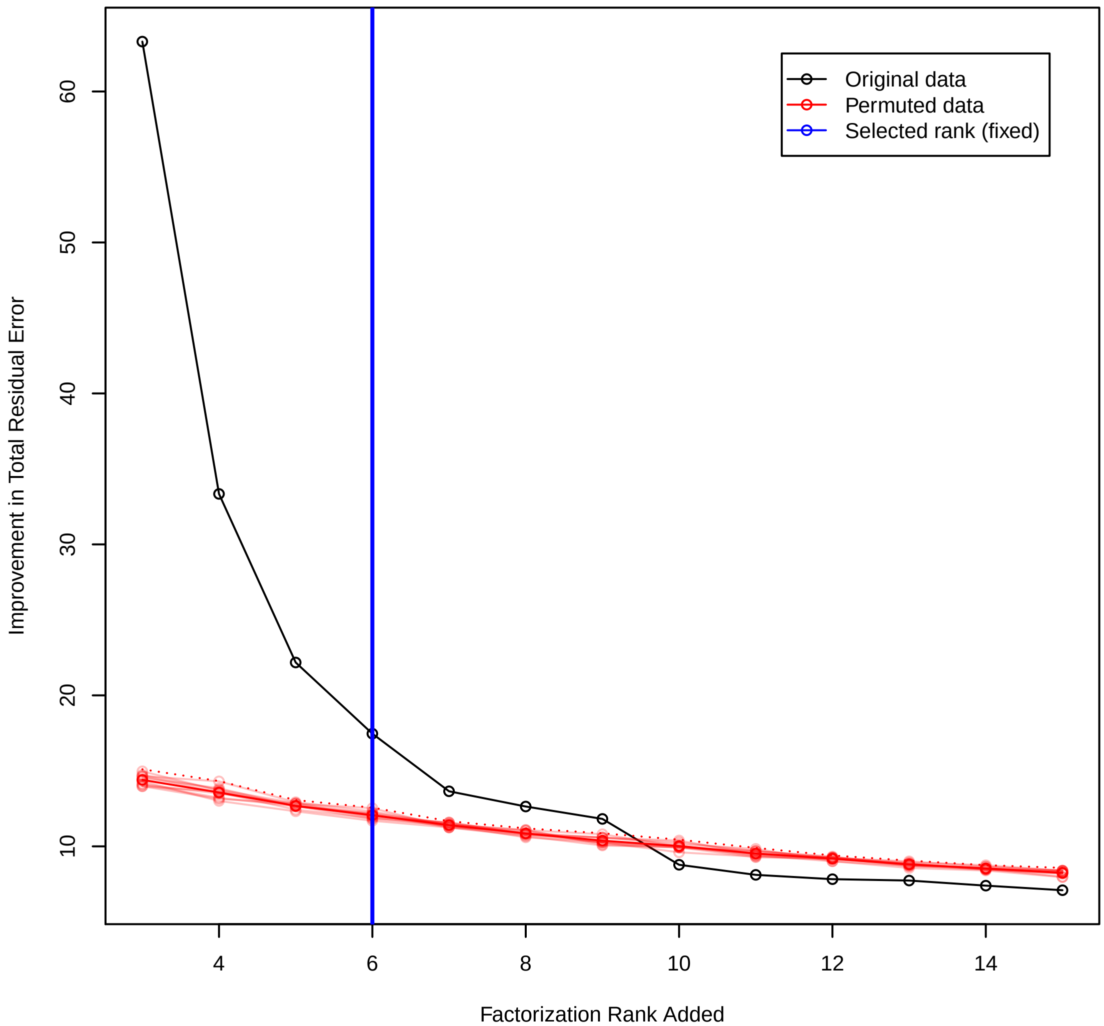



```r
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_scaled),
    ylab = "Scaled residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_scaled[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_scaled[, i]), col = i)
}
```



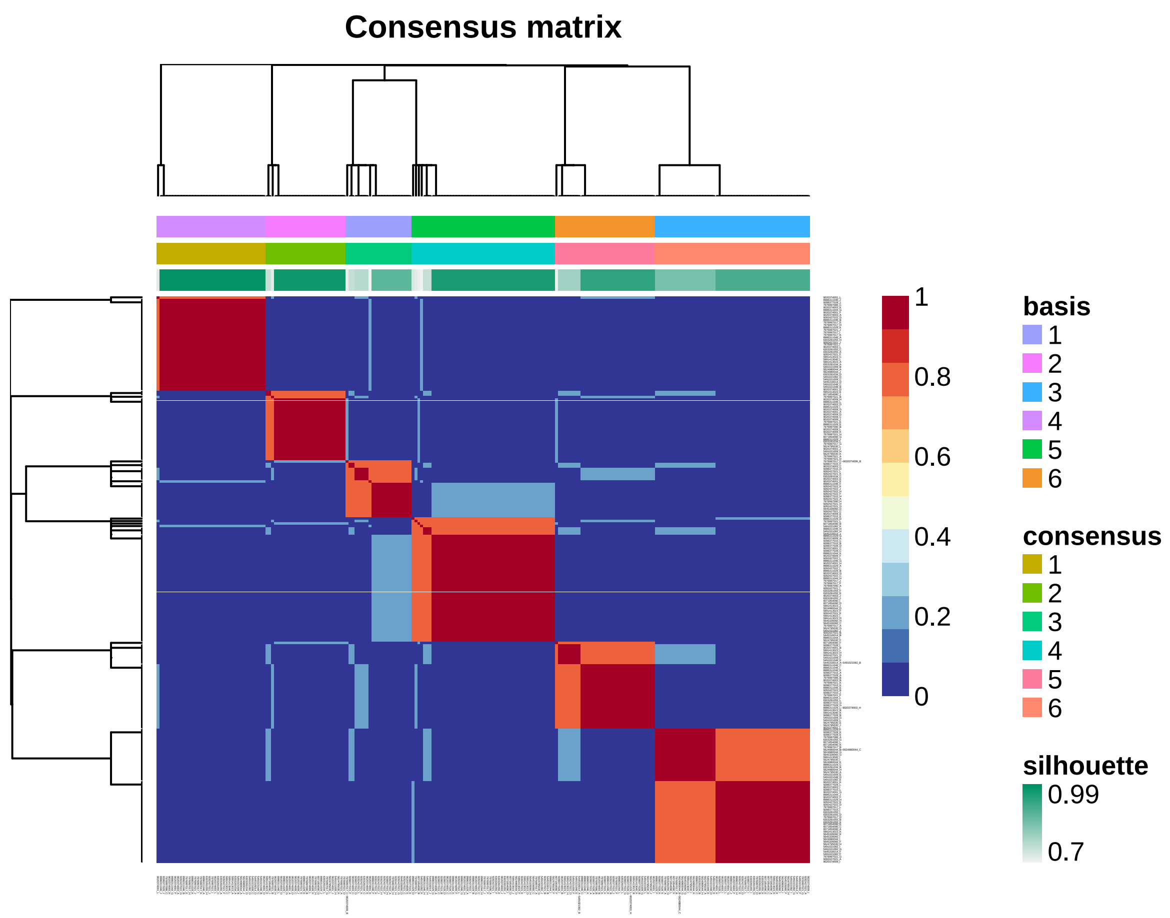

```r
plot(nmf.rankrange[-1], -temp.orig_resids.delta, type = "o", col = "black",
    pch = 21, ylim = range(-c(temp.orig_resids.delta, temp.perm_resids.delta.mean)),
    xlab = "Factorization Rank Added", ylab = "Improvement in Total Residual Error")
lines(nmf.rankrange[-1], -temp.perm_resids.delta.mean, col = "red", type = "o",
    pch = 21, lwd = 1)
for (i in 1:ncol(temp.perm_resids)) {
    lines(nmf.rankrange[-1], -temp.perm_resids.delta[, i], type = "o", col = rgb(1,
        0, 0, 0.25))
}
lines(nmf.rankrange[-1], -temp.perm_resids.delta.threshold, col = "red", lty = "dotted")
if (nmf.rank.wasauto == TRUE) {
    temp.col = "green"
} else {
    temp.col = "blue"
}
abline(v = nmf.rank, col = temp.col, lwd = 2)
legend("topright", legend = c("Original data", "Permuted data", sprintf("Selected rank (%s)",
```

```
ifelse(nmf.rank.wasauto == TRUE, "auto", "fixed"))), col = c("black", "red",
temp.col), lty = "solid", pch = 21, inset = 0.05)
```

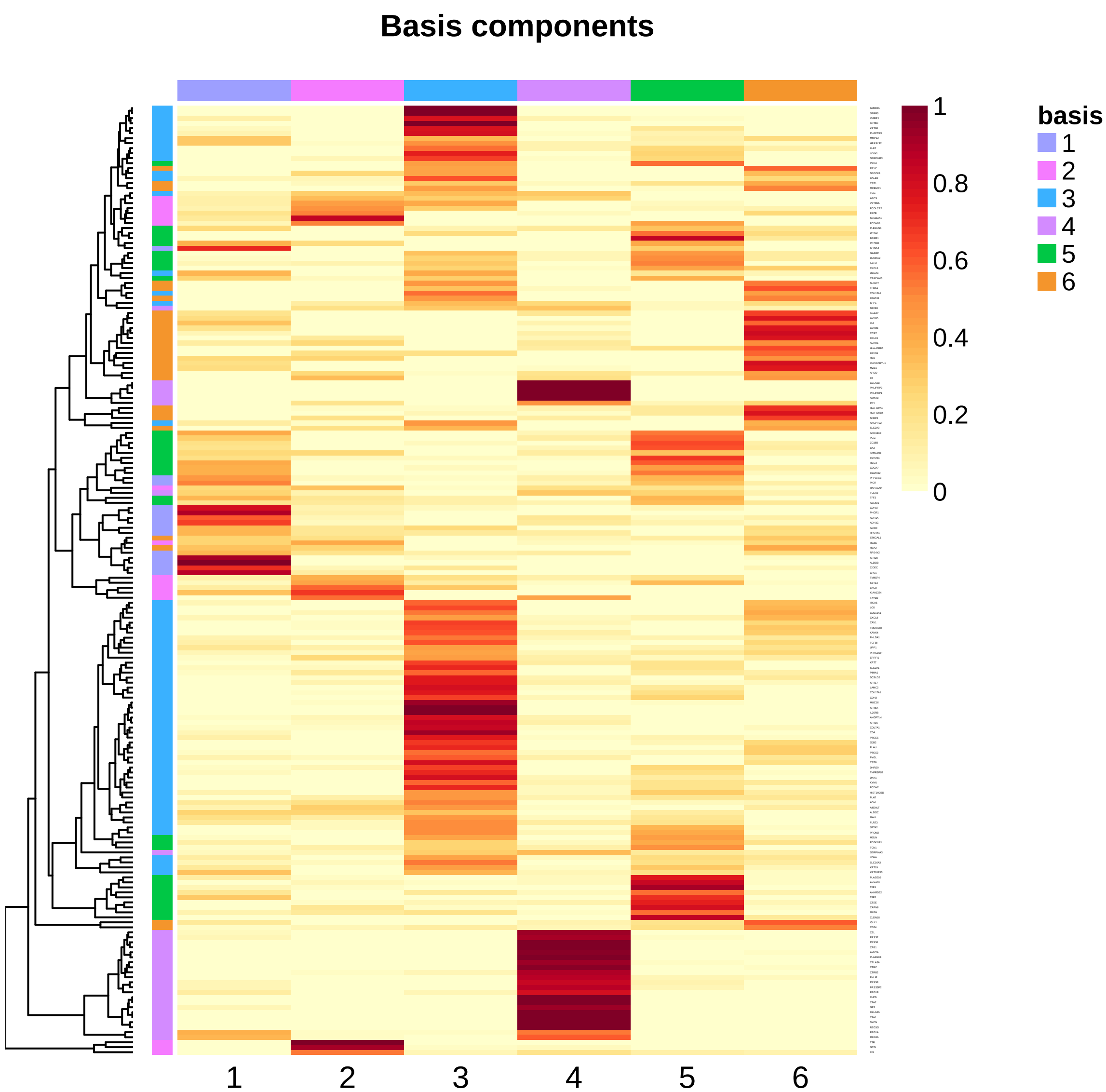


5.1 Fit

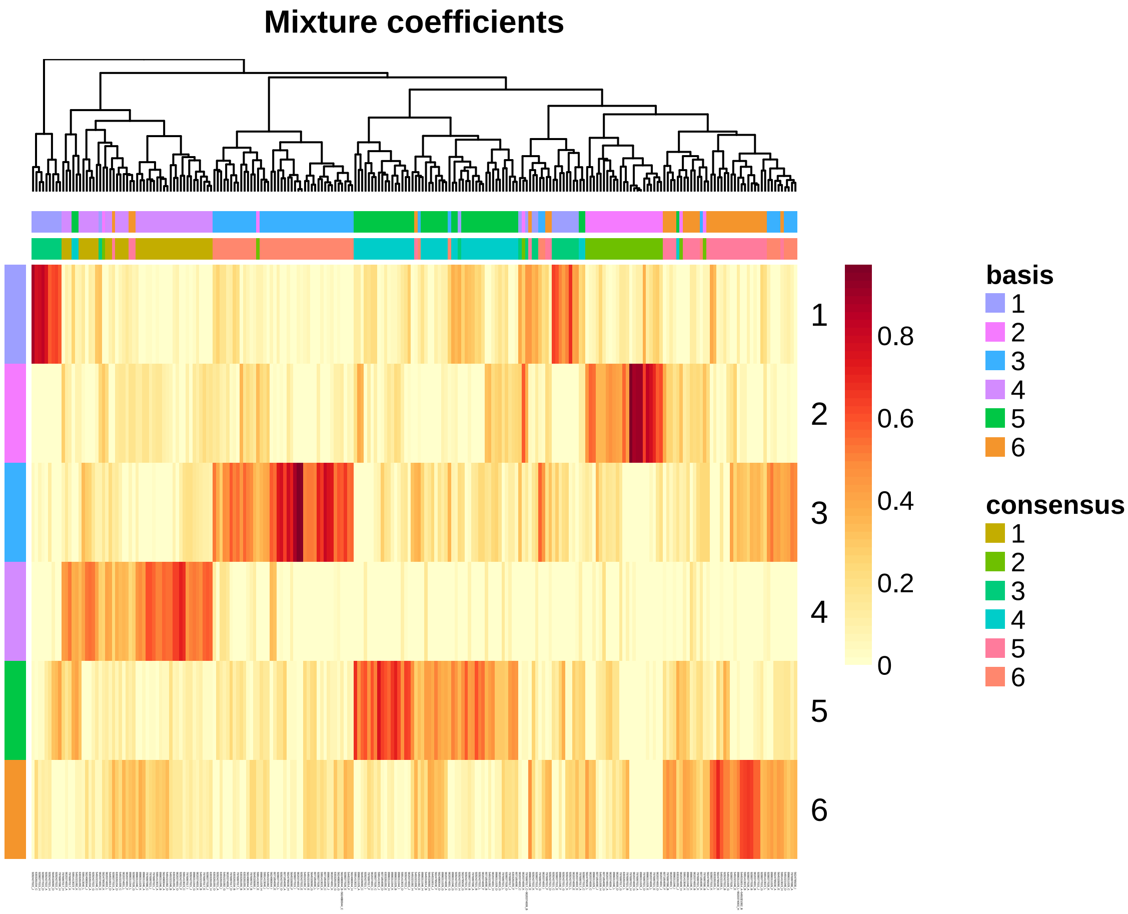
```
consensusmap(xlin.scaled.sel.nmf)
```



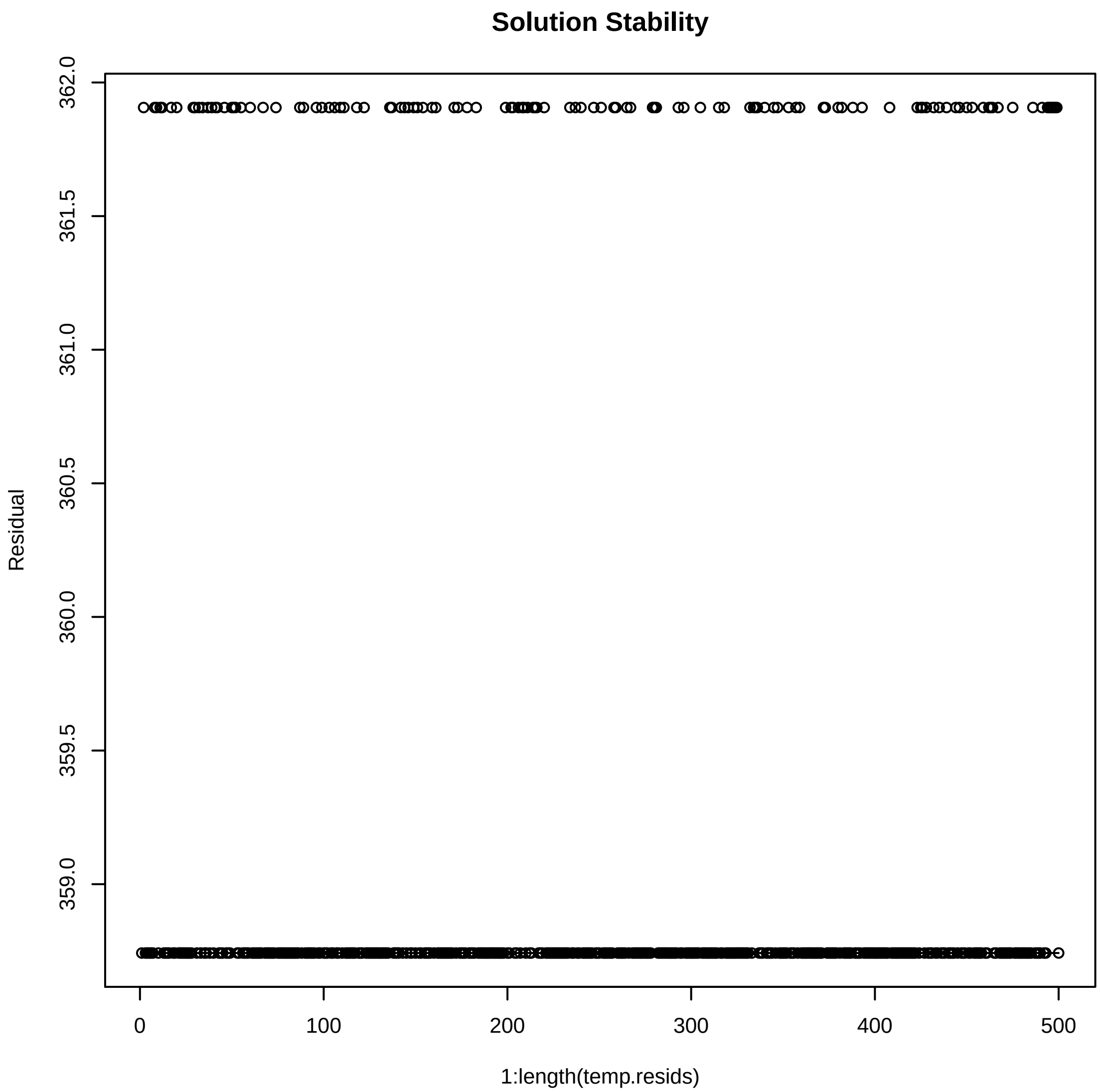

```r
basismap(xlin.scaled.sel.nmf)
```




```
coefmap(xlin.scaled.sel.nmf)
```




```
temp.resids = sapply(xlin.scaled.sel.nmf, residuals)
plot(1:length(temp.resids), temp.resids, ylab = "Residual", main = "Solution Stability")
lines(1:length(temp.resids), cummin(temp.resids))
```



5.2 Component CPV associations

5.2.1 Outcome: Diagnosis to recurrence

```
for (i in 1:ncol(coefs.diag_rec)) {
    print(summary(coxph(y.diag_rec ~ coefs.diag_rec[, i])))
}

## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                           coef exp(coef)  se(coef)    z Pr(>|z|)
## coefs.diag_rec[, i] -7.623743  0.000489  4.015118 -1.9    0.058
##
##                     exp(coef) exp(-coef) lower .95 upper .95
```

```
## coefs.diag_rec[, i]  0.000489       2046  1.87e-07      1.28
##
## Concordance= 0.541  (se = 0.036 )
## Rsquare= 0.038   (max possible= 0.997 )
## Likelihood ratio test= 4.05  on 1 df,   p=0.0441
## Wald test            = 3.61  on 1 df,   p=0.0576
## Score (logrank) test = 3.63  on 1 df,   p=0.0569
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                     coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_rec[, i] 1.63      5.12     2.09 0.78     0.44
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      5.12      0.195    0.0843       310
##
## Concordance= 0.529  (se = 0.036 )
## Rsquare= 0.006   (max possible= 0.997 )
## Likelihood ratio test= 0.59  on 1 df,   p=0.443
## Wald test            = 0.61  on 1 df,   p=0.436
## Score (logrank) test = 0.61  on 1 df,   p=0.436
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                        coef exp(coef) se(coef)   z Pr(>|z|)
## coefs.diag_rec[, i]    6.91   1001.36     1.47 4.7  2.6e-06
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      1001   0.000999      56.1     17869
##
## Concordance= 0.669  (se = 0.036 )
## Rsquare= 0.167   (max possible= 0.997 )
## Likelihood ratio test= 18.9  on 1 df,   p=1.34e-05
## Wald test            = 22.1  on 1 df,   p=2.61e-06
## Score (logrank) test = 22.8  on 1 df,   p=1.79e-06
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                       coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_rec[, i] -1.529     0.217    1.251 -1.22     0.22
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.217       4.61    0.0187      2.52
##
```

```
## Concordance= 0.551  (se = 0.035 )
## Rsquare= 0.015   (max possible= 0.997 )
## Likelihood ratio test= 1.61  on 1 df,   p=0.205
## Wald test            = 1.49  on 1 df,   p=0.222
## Score (logrank) test = 1.51  on 1 df,   p=0.22
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_rec[, i] -1.097     0.334    1.515 -0.72     0.47
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.334          3    0.0171       6.5
##
## Concordance= 0.509  (se = 0.036 )
## Rsquare= 0.005   (max possible= 0.997 )
## Likelihood ratio test= 0.54  on 1 df,   p=0.463
## Wald test            = 0.52  on 1 df,   p=0.469
## Score (logrank) test = 0.53  on 1 df,   p=0.468
##
## Call:
## coxph(formula = y.diag_rec ~ coefs.diag_rec[, i])
##
##   n= 104, number of events= 77
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_rec[, i] -0.872     0.418    1.962 -0.44     0.66
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.418       2.39   0.00894      19.6
##
## Concordance= 0.544  (se = 0.036 )
## Rsquare= 0.002   (max possible= 0.997 )
## Likelihood ratio test= 0.2  on 1 df,   p=0.656
## Wald test            = 0.2  on 1 df,   p=0.657
## Score (logrank) test = 0.2  on 1 df,   p=0.657
```

5.2.2 Outcome: Diagnosis to disease-specific death

```r
for (i in 1:ncol(coefs.diag_dsd)) {
    print(summary(coxph(y.diag_dsd ~ coefs.diag_dsd[, i])))
}
```

```
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
```

```
## coefs.diag_dsd[, i] -1.356     0.258    3.803 -0.36     0.72
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     0.258       3.88  0.000149       445
##
## Concordance= 0.489  (se = 0.037 )
## Rsquare= 0.001   (max possible= 0.995 )
## Likelihood ratio test= 0.13  on 1 df,   p=0.719
## Wald test            = 0.13  on 1 df,   p=0.721
## Score (logrank) test = 0.13  on 1 df,   p=0.721
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                      coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_dsd[, i] 0.754     2.126    2.309 0.33     0.74
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      2.13       0.47     0.023       196
##
## Concordance= 0.501  (se = 0.037 )
## Rsquare= 0.001   (max possible= 0.995 )
## Likelihood ratio test= 0.11  on 1 df,   p=0.745
## Wald test            = 0.11  on 1 df,   p=0.744
## Score (logrank) test = 0.11  on 1 df,   p=0.744
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                        coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.diag_dsd[, i]    8.18   3575.90     1.51 5.42  6.1e-08
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      3576    0.00028       185     69097
##
## Concordance= 0.692  (se = 0.037 )
## Rsquare= 0.202   (max possible= 0.995 )
## Likelihood ratio test= 24.8  on 1 df,   p=6.44e-07
## Wald test            = 29.3  on 1 df,   p=6.12e-08
## Score (logrank) test = 30.6  on 1 df,   p=3.14e-08
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                       coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_dsd[, i] -0.764     0.466    1.259 -0.61     0.54
##
```

```
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     0.466       2.15    0.0395      5.49
##
## Concordance= 0.532  (se = 0.036 )
## Rsquare= 0.003   (max possible= 0.995 )
## Likelihood ratio test= 0.38  on 1 df,   p=0.537
## Wald test            = 0.37  on 1 df,   p=0.544
## Score (logrank) test = 0.37  on 1 df,   p=0.544
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                         coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_dsd[, i] -4.0471    0.0175   1.7939 -2.26    0.024
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]    0.0175       57.2  0.000519     0.588
##
## Concordance= 0.583  (se = 0.038 )
## Rsquare= 0.05   (max possible= 0.995 )
## Likelihood ratio test= 5.6  on 1 df,   p=0.018
## Wald test            = 5.09  on 1 df,   p=0.0241
## Score (logrank) test = 5.17  on 1 df,   p=0.023
##
## Call:
## coxph(formula = y.diag_dsd ~ coefs.diag_dsd[, i])
##
##   n= 110, number of events= 70
##
##                         coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.diag_dsd[, i] -3.1598    0.0424   2.1621 -1.46     0.14
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]    0.0424       23.6  0.000613      2.94
##
## Concordance= 0.577  (se = 0.038 )
## Rsquare= 0.02   (max possible= 0.995 )
## Likelihood ratio test= 2.17  on 1 df,   p=0.141
## Wald test            = 2.14  on 1 df,   p=0.144
## Score (logrank) test = 2.15  on 1 df,   p=0.143
```

5.2.3 Outcome: Recurrence to disease-specific death

```
for (i in 1:ncol(coefs.recr_dsd)) {
    print(summary(coxph(y.recr_dsd ~ coefs.recr_dsd[, i])))
}

## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
```

```
##   n= 81, number of events= 64
##
##                          coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.recr_dsd[, i] 1.36e+01  8.08e+05 4.26e+00 3.19   0.0014
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]    808046   1.24e-06       191  3.42e+09
##
## Concordance= 0.606  (se = 0.041 )
## Rsquare= 0.102   (max possible= 0.997 )
## Likelihood ratio test= 8.74  on 1 df,   p=0.00311
## Wald test            = 10.2  on 1 df,   p=0.00141
## Score (logrank) test = 10.6  on 1 df,   p=0.00116
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                        coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.recr_dsd[, i] -0.170     0.844    2.519 -0.07     0.95
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]     0.844       1.18   0.00606       118
##
## Concordance= 0.542  (se = 0.041 )
## Rsquare= 0   (max possible= 0.997 )
## Likelihood ratio test= 0  on 1 df,   p=0.946
## Wald test            = 0  on 1 df,   p=0.946
## Score (logrank) test = 0  on 1 df,   p=0.946
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                        coef exp(coef) se(coef)    z Pr(>|z|)
## coefs.recr_dsd[, i]   5.51    246.36     1.73 3.18   0.0015
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]       246    0.00406      8.27      7340
##
## Concordance= 0.637  (se = 0.041 )
## Rsquare= 0.108   (max possible= 0.997 )
## Likelihood ratio test= 9.24  on 1 df,   p=0.00237
## Wald test            = 10.1  on 1 df,   p=0.00147
## Score (logrank) test = 10.3  on 1 df,   p=0.0013
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                      coef exp(coef) se(coef)    z Pr(>|z|)
```

```
## coefs.recr_dsd[, i] 0.64      1.90     1.33 0.48     0.63
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]       1.9      0.527     0.141      25.5
##
## Concordance= 0.501  (se = 0.04 )
## Rsquare= 0.003   (max possible= 0.997 )
## Likelihood ratio test= 0.23  on 1 df,   p=0.634
## Wald test            = 0.23  on 1 df,   p=0.629
## Score (logrank) test = 0.23  on 1 df,   p=0.629
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                          coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.recr_dsd[, i] -4.63708   0.00969  2.03266 -2.28    0.023
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]   0.00969        103   0.00018      0.52
##
## Concordance= 0.598  (se = 0.041 )
## Rsquare= 0.068   (max possible= 0.997 )
## Likelihood ratio test= 5.73  on 1 df,   p=0.0166
## Wald test            = 5.2  on 1 df,   p=0.0225
## Score (logrank) test = 5.26  on 1 df,   p=0.0219
##
## Call:
## coxph(formula = y.recr_dsd ~ coefs.recr_dsd[, i])
##
##   n= 81, number of events= 64
##
##                          coef exp(coef) se(coef)     z Pr(>|z|)
## coefs.recr_dsd[, i] -6.34448   0.00176  2.51118 -2.53    0.012
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]   0.00176        569  1.28e-05     0.241
##
## Concordance= 0.595  (se = 0.041 )
## Rsquare= 0.077   (max possible= 0.997 )
## Likelihood ratio test= 6.45  on 1 df,   p=0.0111
## Wald test            = 6.38  on 1 df,   p=0.0115
## Score (logrank) test = 6.48  on 1 df,   p=0.0109
```

5.2.4 Purity

```
apply(coefs, 2, function(xc) cor.test(samps$purity_qpure, xc, method = "kendall"))
```

```
## $mg.1
##
##  Kendall's rank correlation tau
```

```
##
## data:  samps$purity_qpure and xc
## z = 3.423, p-value = 0.0006188
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.1938
##
##
## $mg.2
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -1.639, p-value = 0.1012
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.09352
##
##
## $mg.3
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -0.8905, p-value = 0.3732
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.04993
##
##
## $mg.4
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -4.676, p-value = 2.924e-06
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.2752
##
##
## $mg.5
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 1.64, p-value = 0.101
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
```

```
## 0.09249
##
##
## $mg.6
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -7.03, p-value = 2.058e-12
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.3972
```

5.3 MTC P-values

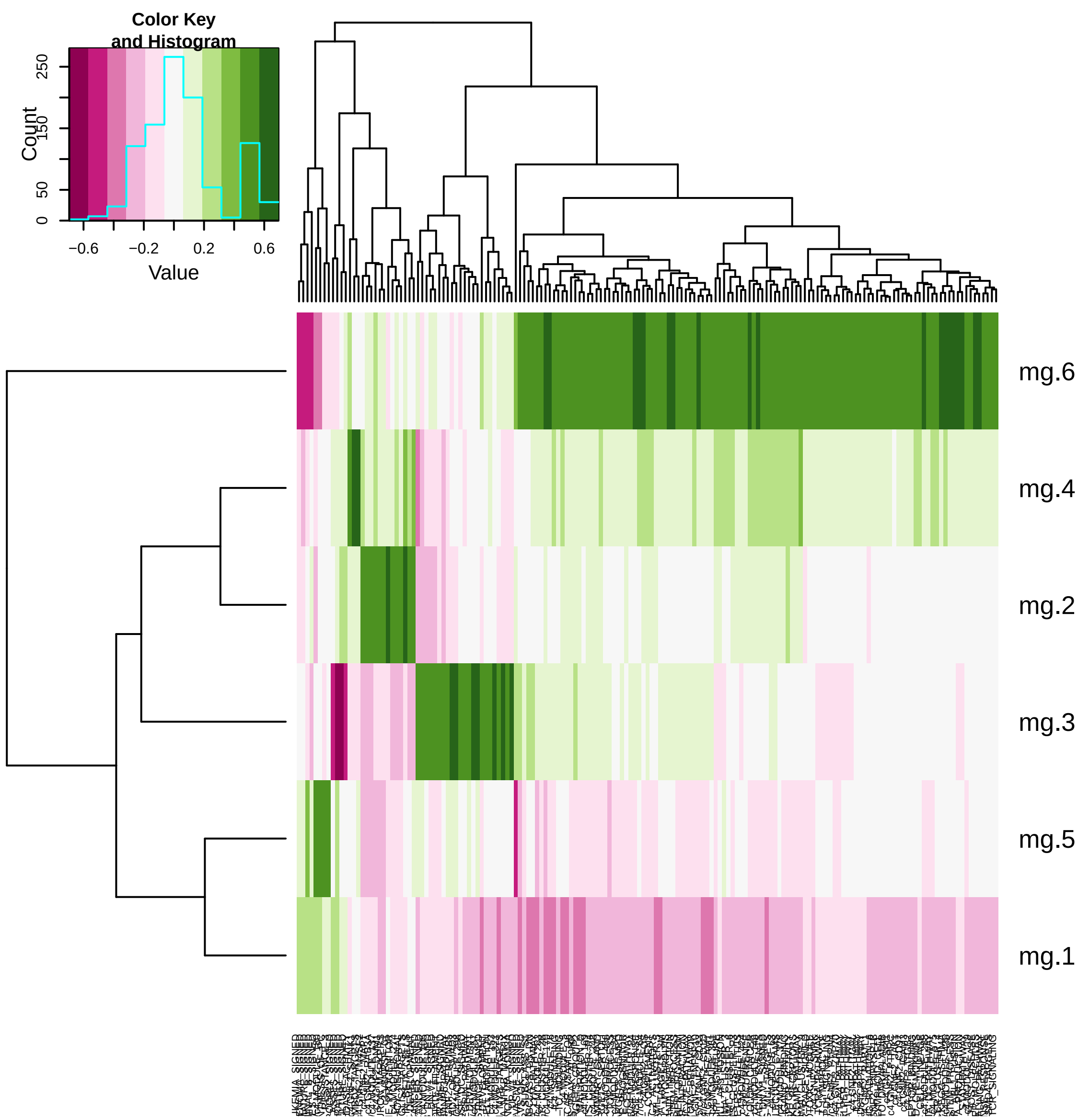
```
xlin.scaled.sel.nmf.cpv.pvals = data.frame(surv.diag_rec.p = apply(coefs.diag_rec,
    2, function(xc) pchisq(2 * diff(coxph(y.diag_rec ~ xc)$loglik), df = 1,
        lower.tail = FALSE)), surv.diag_rec.c = apply(coefs.diag_rec, 2, function(xc) coef(coxph(y.diag_
    xc))), surv.diag_dsd.p = apply(coefs.diag_dsd, 2, function(xc) pchisq(2 *
    diff(coxph(y.diag_dsd ~ xc)$loglik), df = 1, lower.tail = FALSE)), surv.diag_dsd.c = apply(coefs.dia
    2, function(xc) coef(coxph(y.diag_dsd ~ xc))), surv.recr_dsd.p = apply(coefs.recr_dsd,
    2, function(xc) pchisq(2 * diff(coxph(y.recr_dsd ~ xc)$loglik), df = 1,
        lower.tail = FALSE)), surv.recr_dsd.c = apply(coefs.recr_dsd, 2, function(xc) coef(coxph(y.recr_
    xc))), pure.p = apply(coefs, 2, function(xc) cor.test(samps$purity_qpure,
    xc, method = "kendall")$p.value), pure.s = apply(coefs, 2, function(xc) cor.test(samps$purity_qpure,
    xc, method = "kendall")$statistic))
temp.pvals = as.matrix(xlin.scaled.sel.nmf.cpv.pvals[, grepl("\\.p$", colnames(xlin.scaled.sel.nmf.cpv.p
temp.pvals.FWER = matrix(p.adjust(as.vector(temp.pvals), "holm"), nrow = nrow(temp.pvals))
colnames(temp.pvals.FWER) = paste(colnames(temp.pvals), "Holm", sep = ".")
temp.pvals.BY = matrix(p.adjust(as.vector(temp.pvals), "BY"), nrow = nrow(temp.pvals))
colnames(temp.pvals.BY) = paste(colnames(temp.pvals), "BY", sep = ".")
xlin.scaled.sel.nmf.cpv.pvals = cbind(xlin.scaled.sel.nmf.cpv.pvals, temp.pvals.FWER,
    temp.pvals.BY)
xlin.scaled.sel.nmf.cpv.pvals = xlin.scaled.sel.nmf.cpv.pvals[, order(colnames(xlin.scaled.sel.nmf.cpv.p
xlin.scaled.sel.nmf.cpv.pvals

##         pure.p pure.p.BY pure.p.Holm  pure.s surv.diag_dsd.c
## mg.1 6.188e-04 1.122e-02   1.238e-02  3.4232         -1.3558
## mg.2 1.012e-01 7.053e-01   1.000e+00 -1.6392          0.7544
## mg.3 3.732e-01 1.000e+00   1.000e+00 -0.8905          8.1820
## mg.4 2.924e-06 8.832e-05   6.432e-05 -4.6761         -0.7639
## mg.5 1.010e-01 7.053e-01   1.000e+00  1.6399         -4.0471
## mg.6 2.058e-12 1.865e-10   4.940e-11 -7.0305         -3.1598
##      surv.diag_dsd.p surv.diag_dsd.p.BY surv.diag_dsd.p.Holm
## mg.1       7.186e-01          1.0000000            1.000e+00
## mg.2       7.454e-01          1.0000000            1.000e+00
## mg.3       6.445e-07          0.0000292            1.482e-05
## mg.4       5.369e-01          1.0000000            1.000e+00
## mg.5       1.796e-02          0.1627223            2.693e-01
## mg.6       1.407e-01          0.9107806            1.000e+00
##      surv.diag_rec.c surv.diag_rec.p surv.diag_rec.p.BY
```

	pure.p.Holm	pure.s	surv.diag_dsd.c	surv.diag_dsd.p.Holm	surv.diag_rec.c	surv.diag_rec.p.Holm	surv.recr_dsd.c	surv.recr_dsd.p.Holm
mg.1	0.0124	3.4232	-1.3558	1.0000	-7.6237	0.6177	13.6024	0.0559
mg.2	1.0000	-1.6392	0.7544	1.0000	1.6322	1.0000	-0.1697	1.0000
mg.3	1.0000	-0.8905	8.1820	0.0000	6.9091	0.0003	5.5068	0.0450
mg.4	0.0001	-4.6761	-0.7639	1.0000	-1.5286	1.0000	0.6404	1.0000
mg.5	1.0000	1.6399	-4.0471	0.2693	-1.0972	1.0000	-4.6371	0.2663
mg.6	0.0000	-7.0305	-3.1598	1.0000	-0.8718	1.0000	-6.3445	0.1882

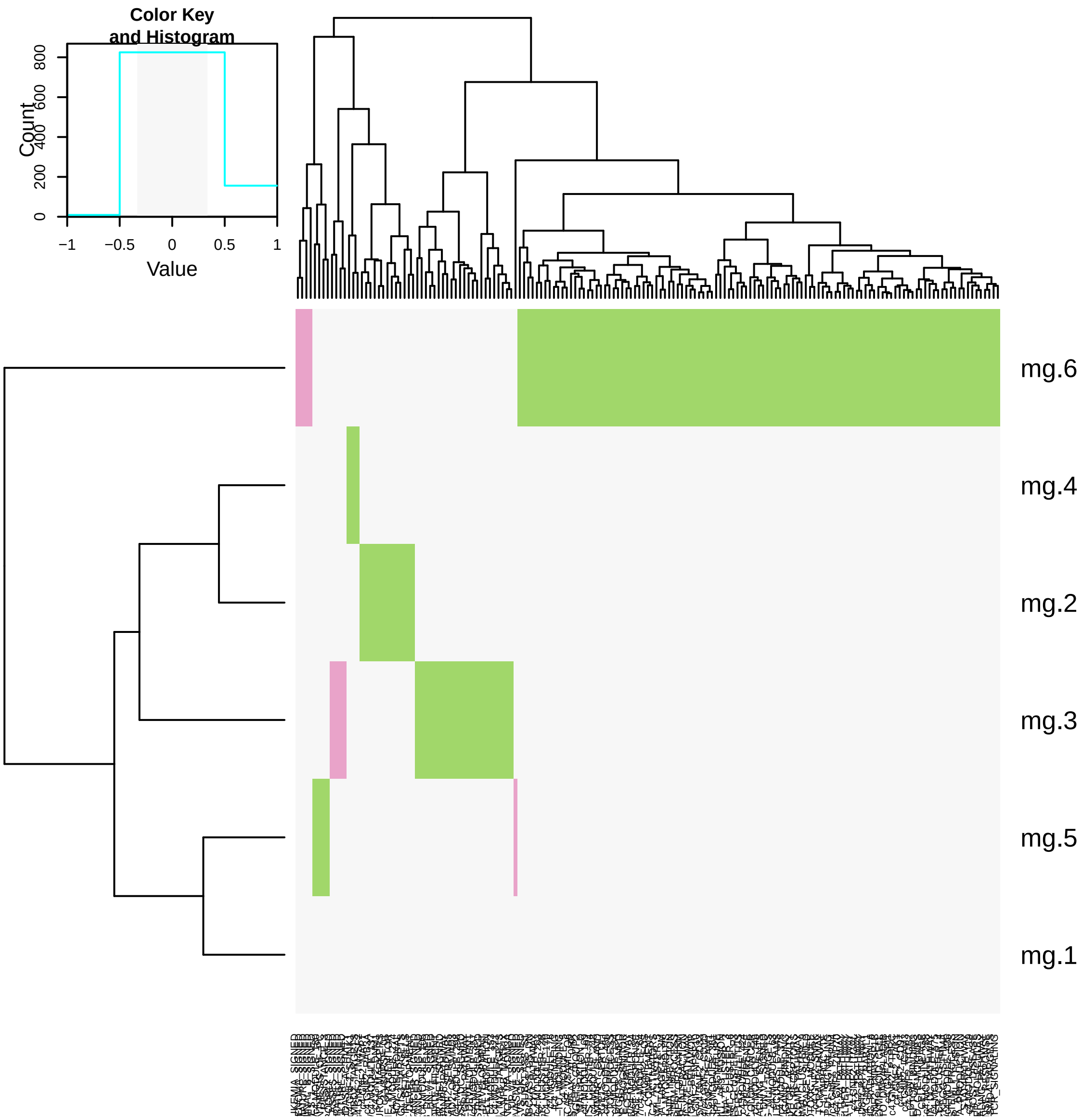
```
## mg.1         -7.6237       4.412e-02           0.363506
## mg.2          1.6322       4.428e-01           1.000000
## mg.3          6.9091       1.342e-05           0.000304
## mg.4         -1.5286       2.051e-01           1.000000
## mg.5         -1.0972       4.629e-01           1.000000
## mg.6         -0.8718       6.560e-01           1.000000
##      surv.diag_rec.p.Holm surv.recr_dsd.c surv.recr_dsd.p
## mg.1            0.6177229         13.6024        0.003108
## mg.2            1.0000000         -0.1697        0.946265
## mg.3            0.0002818          5.5068        0.002369
## mg.4            1.0000000          0.6404        0.634115
## mg.5            1.0000000         -4.6371        0.016645
## mg.6            1.0000000         -6.3445        0.011070
##      surv.recr_dsd.p.BY surv.recr_dsd.p.Holm
## mg.1            0.04024              0.05595
## mg.2            1.00000              1.00000
## mg.3            0.03578              0.04501
## mg.4            1.00000              1.00000
## mg.5            0.16272              0.26632
## mg.6            0.12540              0.18819
```

5.4 MSigDB score correlation thresholding

```
temp.sel_cols = apply(abs(xlin.scaled.sel.nmf.msigdb.corr) >= sig.corr.threshold,
    2, any)
heatmap.2(xlin.scaled.sel.nmf.msigdb.corr[, temp.sel_cols], trace = "none",
    scale = "none", useRaster = TRUE, col = brewer.pal(11, "PiYG"), symbreaks = TRUE)
```




```r
heatmap.2(xlin.scaled.sel.nmf.msigdb.corr[, temp.sel_cols], trace = "none",
    scale = "none", useRaster = TRUE, col = brewer.pal(3, "PiYG"), breaks = c(-1,
        -sig.corr.threshold, sig.corr.threshold, 1))
```




```
temp.sig_id = colnames(xlin.scaled.sel.nmf.msigdb.corr)
temp.sig_class = gsub("\\..*", "", temp.sig_id)
temp.nsigs = length(temp.sig_id)
temp.nmeta = nrow(xlin.scaled.sel.nmf.msigdb.corr)
tables = lapply(1:temp.nmeta, function(metagene_i) {
    tapply(1:temp.nsigs, temp.sig_class, function(sig_class_is) {
        all_cors = xlin.scaled.sel.nmf.msigdb.corr[, sig_class_is]
        this_cors = all_cors[metagene_i, ]
        this_ids = temp.sig_id[sig_class_is]

        all_sig_cors = abs(all_cors) >= sig.corr.threshold
        this_sig_cors = all_sig_cors[metagene_i, ]

        sigs_to_report = which(this_sig_cors)

        if (length(sigs_to_report) == 0) {
            table = data.frame(GeneSet = c(), Correlation = c(), Metagenes = c())
```

```
        } else {
            table = data.frame(GeneSet = this_ids[sigs_to_report], Correlation = this_cors[sigs_to_repor
                Metagenes = apply(all_cors[, sigs_to_report, drop = FALSE],
                  2, function(cors) {
                    sel = abs(cors) >= sig.corr.threshold
                    # A positive number implies that positive GSVA signal is associated with
                    # worse prognosis
                    paste(which(sel) * sign(cors[which(sel)]) * sign(xlin.scaled.sel.nmf.cpv.pvals$d.sur
                      collapse = ",")
                  }))
            table = table[order(-(table$Correlation)), ]
            rownames(table) <- NULL
        }
        table
    }, simplify = FALSE)
})

## Error in sign(xlin.scaled.sel.nmf.cpv.pvals$d.surv[metagene_i]):  non-numeric argument to
mathematical function

tables

## Error in eval(expr, envir, enclos):  object 'tables' not found
```

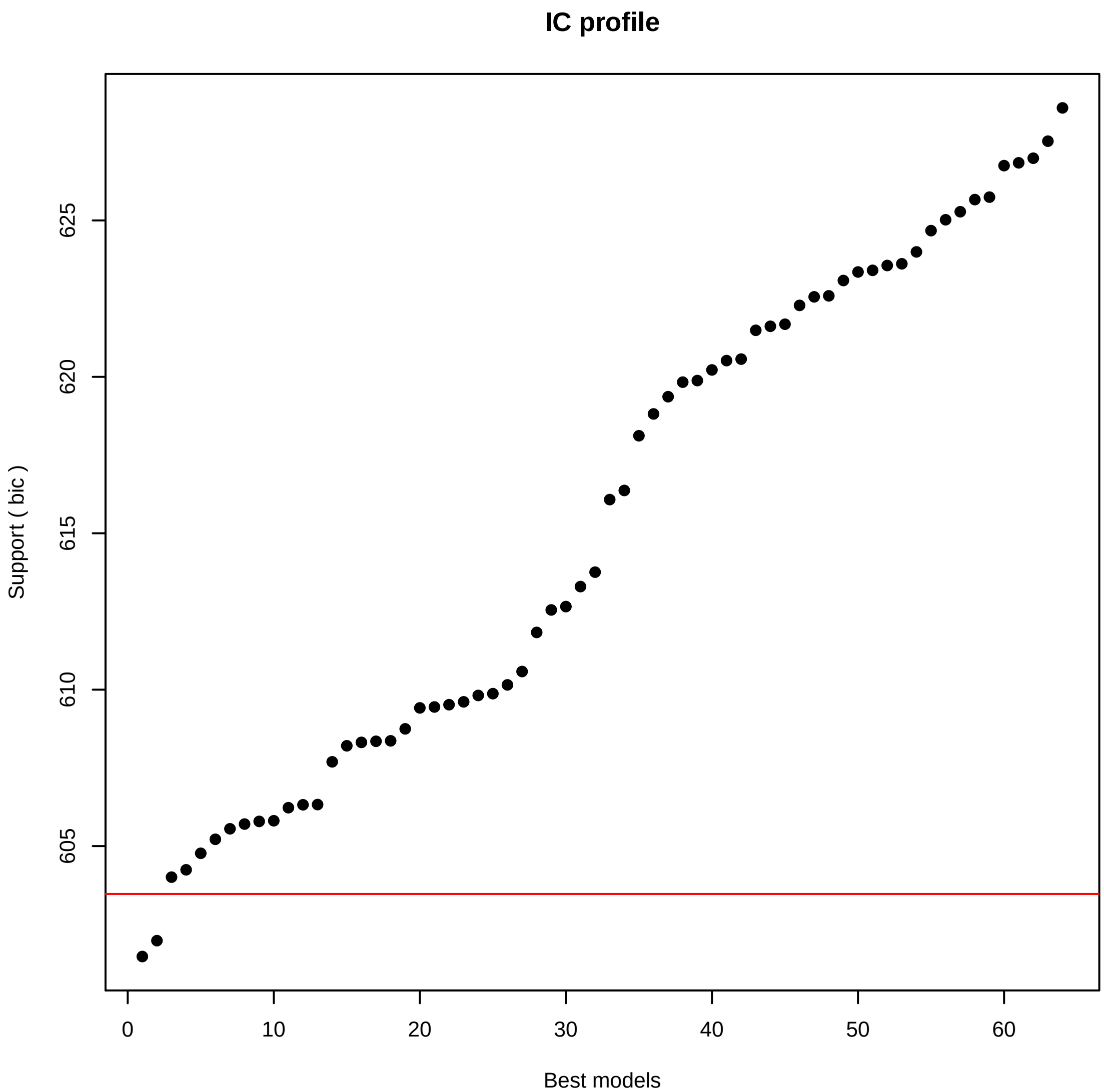
5.4.1 Outcome: Diagnosis to recurrence

```
print(diag_rec.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 64 models:
## Best IC: 601.470175374294
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.3"
## Evidence weight: 0.266632885140608
## Worst IC: 628.597914588117
## 2 models within 2 IC units.
## 17 models to reach 95% of evidence weight.

coef(diag_rec.asreg.result)

##      Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.4   0.0303          0.04091        32     0.1128           0.4012
## mg.6   0.4796          1.02186        32     0.1686           2.0051
## mg.1  -1.0008          3.91030        32     0.1954           3.9223
## mg.5   0.7846          1.85063        32     0.2540           2.6984
## mg.2   1.8517          5.96799        32     0.4332           4.8457
## mg.3   7.7233          3.63008        32     0.9993           3.7792

summary(diag_rec.asreg.result@objects[[1]])
```

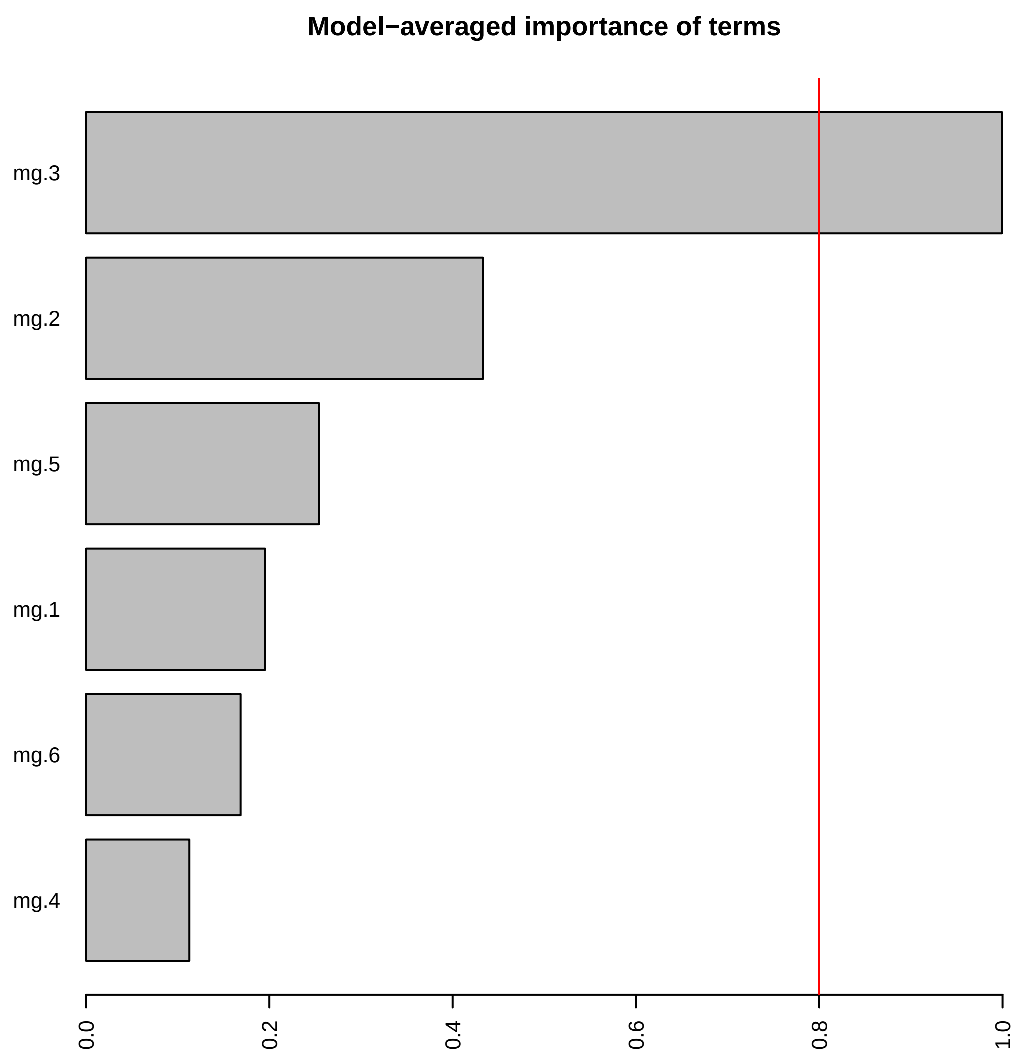
```
## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 104, number of events= 77
##
##         coef exp(coef) se(coef)   z Pr(>|z|)
## mg.3    6.91   1001.36     1.47 4.7  2.6e-06
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.3      1001   0.000999      56.1     17869
##
## Concordance= 0.669  (se = 0.036 )
## Rsquare= 0.167   (max possible= 0.997 )
## Likelihood ratio test= 18.9  on 1 df,   p=1.34e-05
## Wald test            = 22.1  on 1 df,   p=2.61e-06
## Score (logrank) test = 22.8  on 1 df,   p=1.79e-06
```

```
plot(diag_rec.asreg.result, type = "p")
```

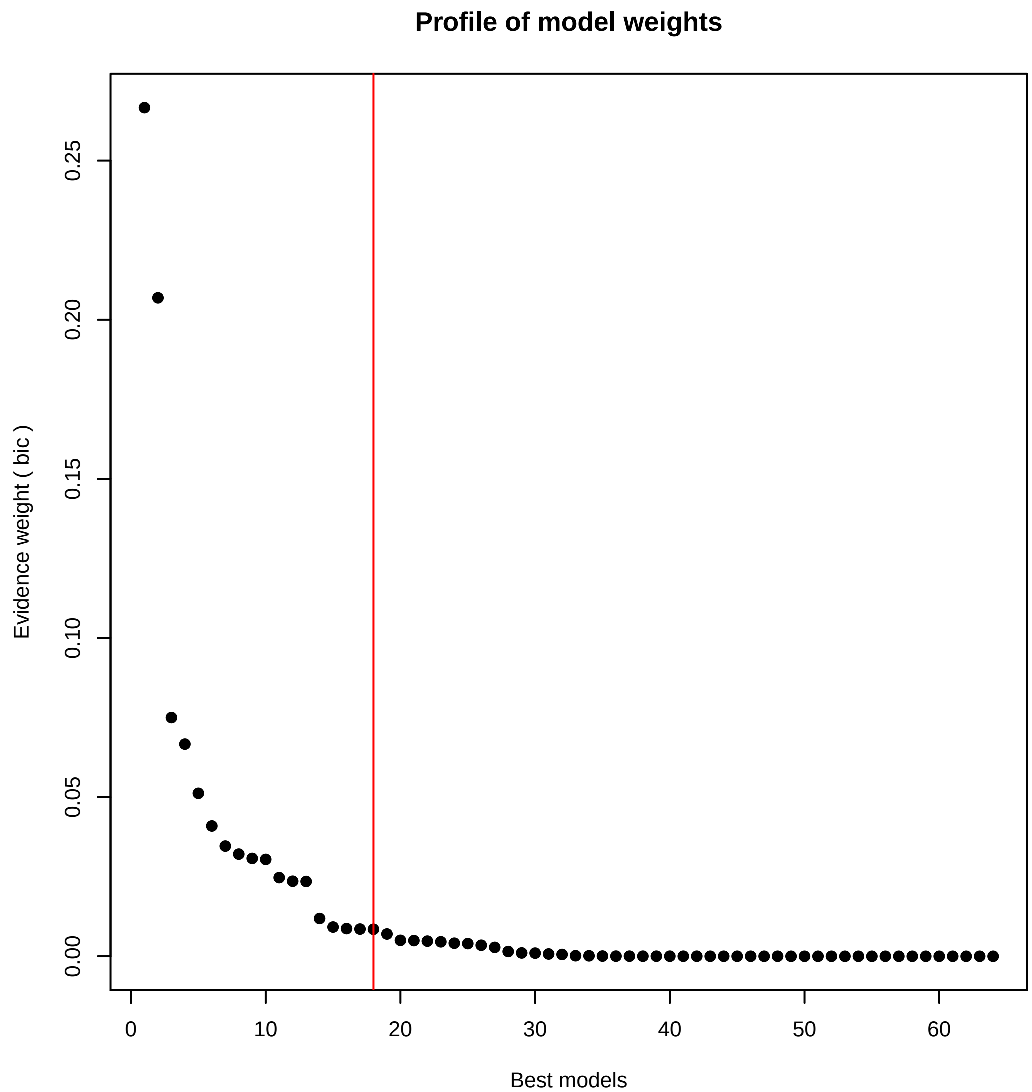
All-subsets regression




```
plot(diag_rec.asreg.result, type = "s")
```




```r
plot(diag_rec.asreg.result, type = "w")
```



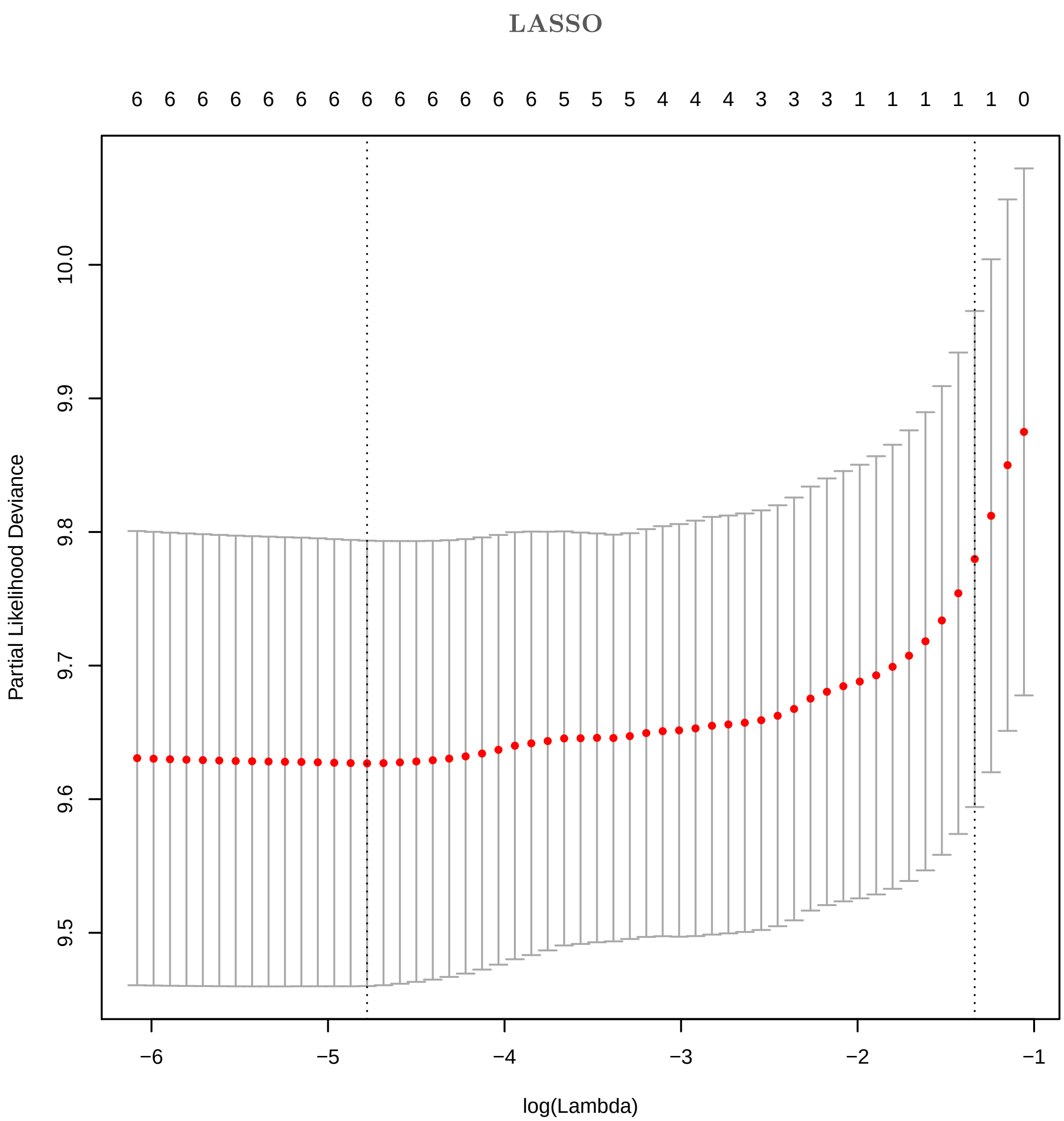

```
diag_rec.glmnet.coef.1se
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##         1
## mg.1 .
## mg.2 .
## mg.3 2.03
## mg.4 .
## mg.5 .
## mg.6 .
```

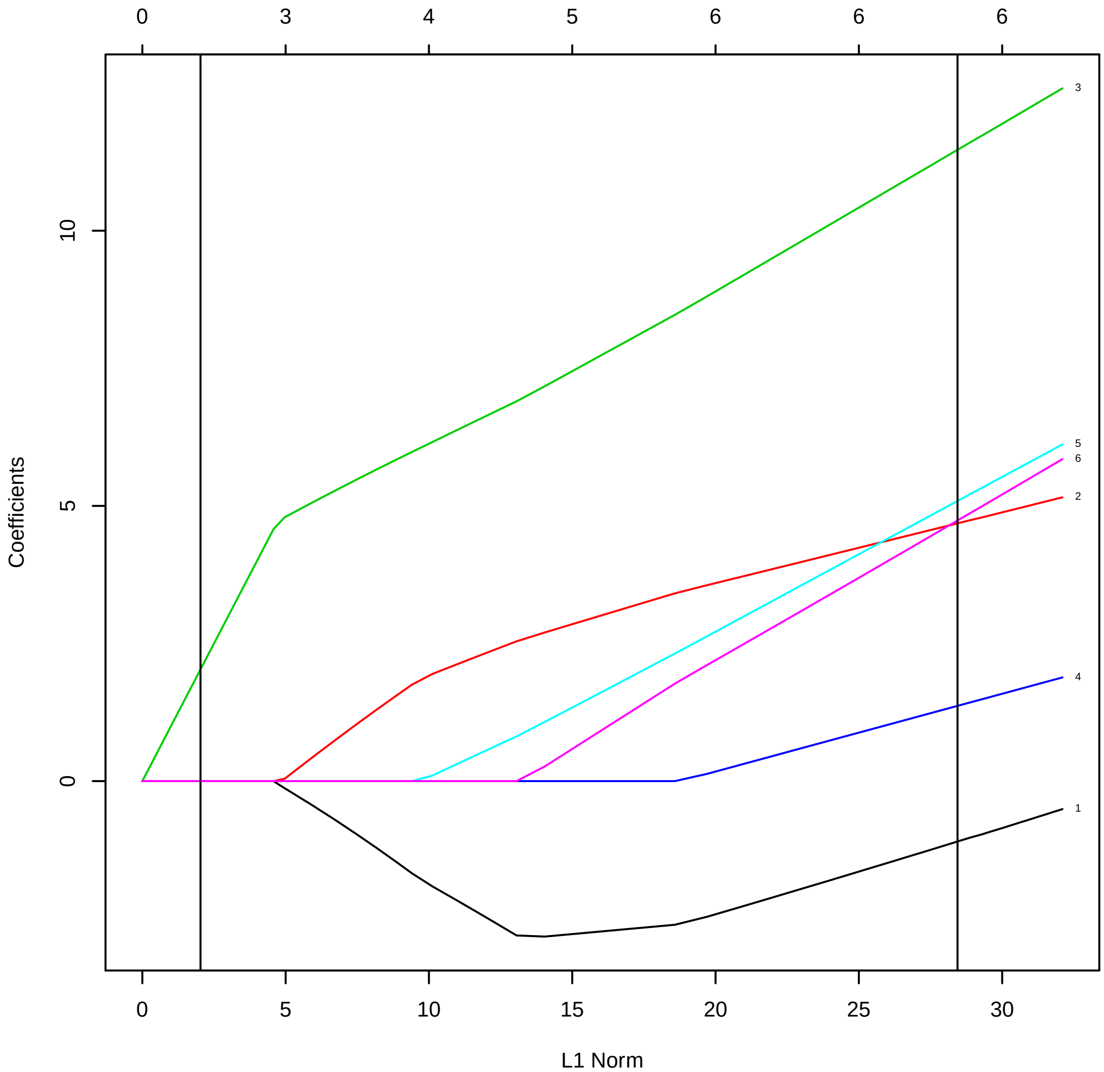
```
diag_rec.glmnet.coef.min
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##            1
## mg.1 -1.097
## mg.2  4.683
## mg.3 11.470
## mg.4  1.367
## mg.5  5.090
## mg.6  4.735
```

```r
plot(diag_rec.glmnet.fit.cv)
```



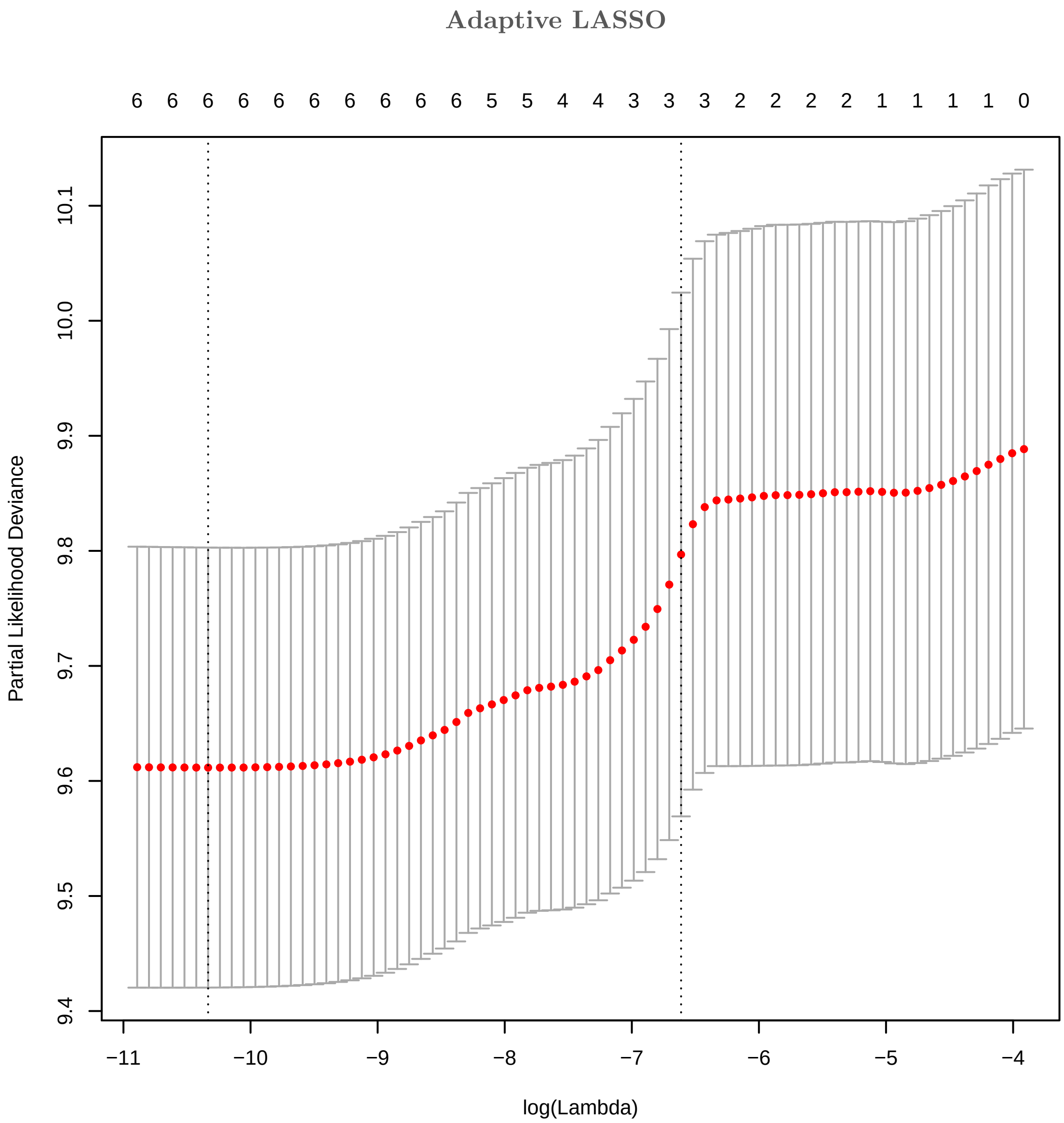

```r
plot(diag_rec.glmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_rec.glmnet.coef.1se)))
abline(v = sum(abs(diag_rec.glmnet.coef.min)))
```



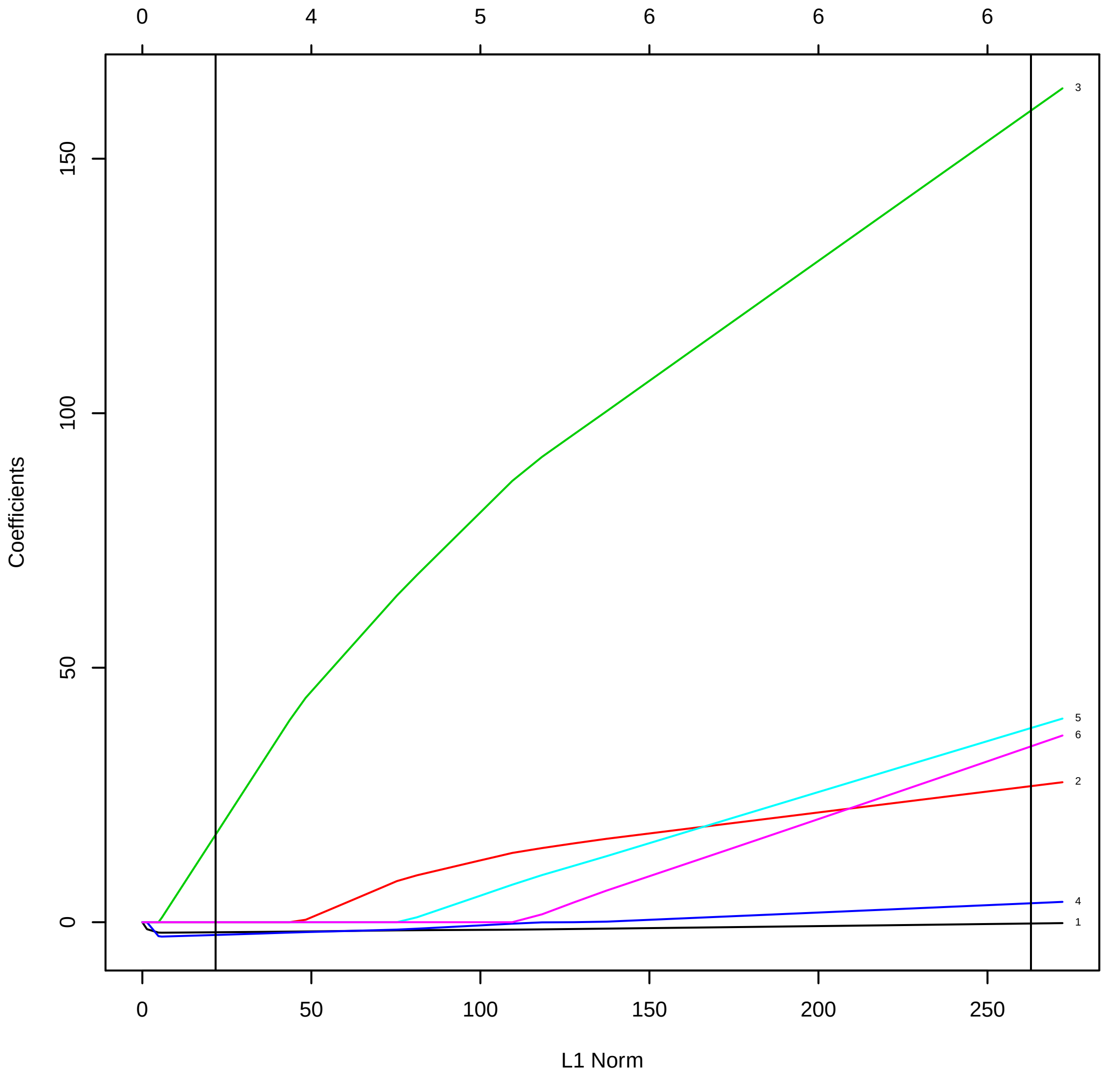

```
diag_rec.adaglmnet.coef.1se/diag_rec.adaglmnet.weights

## 6 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  -0.5362
## mg.2   .
## mg.3 224.1822
## mg.4  -5.2533
## mg.5   .
## mg.6   .

diag_rec.adaglmnet.coef.min/diag_rec.adaglmnet.weights
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                1
## mg.1   -0.06949
## mg.2  142.91935
## mg.3 2079.63653
## mg.4    7.80082
## mg.5  249.14305
## mg.6  217.63954
```

```
plot(diag_rec.adaglmnet.fit.cv)
```




```
plot(diag_rec.adaglmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_rec.adaglmnet.coef.1se)))
abline(v = sum(abs(diag_rec.adaglmnet.coef.min)))
```



5.4.2 Outcome: Diagnosis to disease-specific death

```
print(diag_dsd.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 64 models:
## Best IC: 556.461922765798
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.3"
## Evidence weight: 0.369559134422938
## Worst IC: 590.587347586262
## 2 models within 2 IC units.
## 14 models to reach 95% of evidence weight.
```

```r
coef(diag_dsd.asreg.result)
```

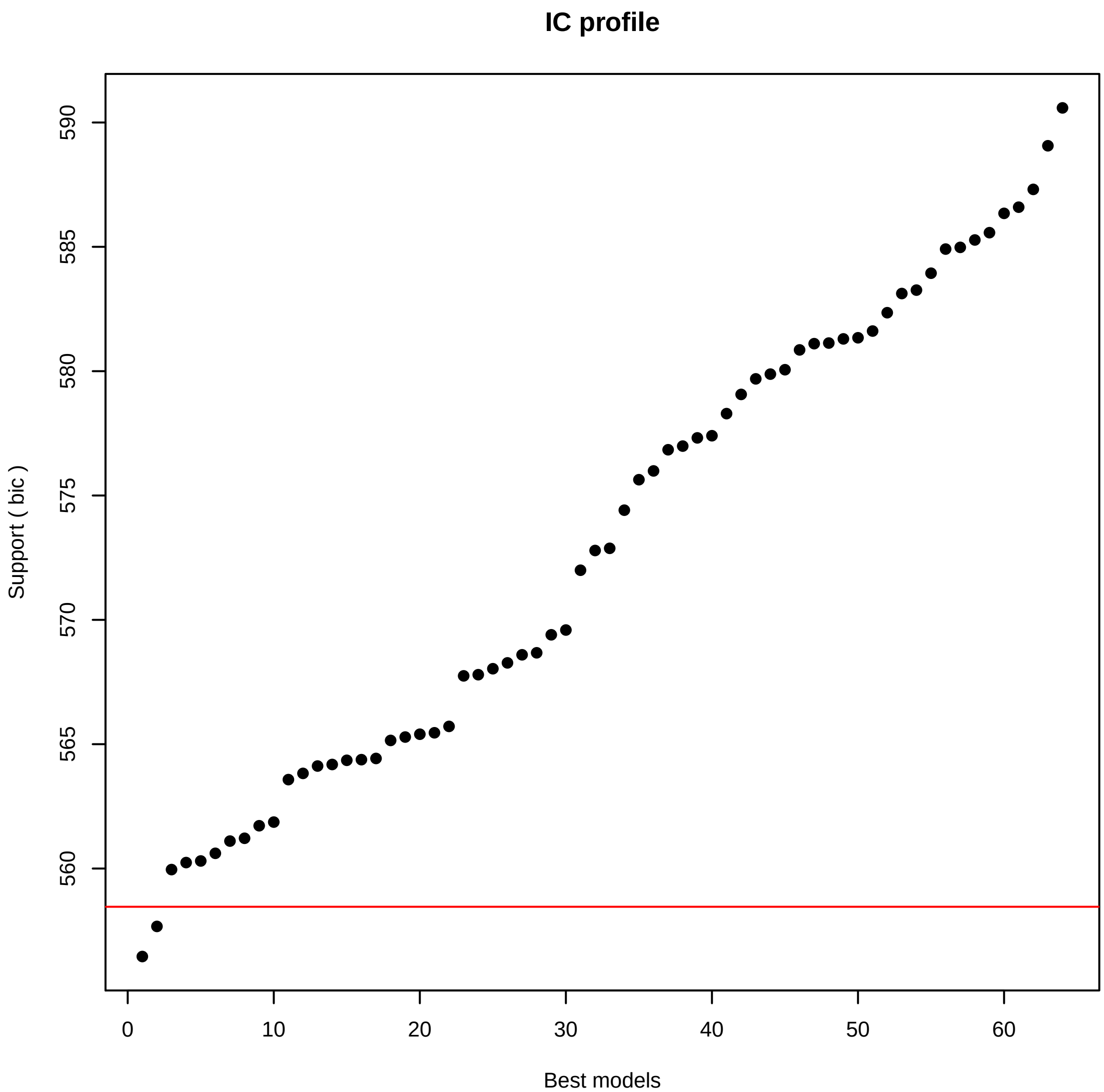
```
##      Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.5 -0.06016          0.08894        32     0.1122           0.5911
## mg.6 -0.16519          0.22964        32     0.1236           0.9499
## mg.1  0.42474          1.08846        32     0.1363           2.0680
## mg.4  0.18944          0.18110        32     0.1507           0.8435
## mg.2  1.47263          4.86163        32     0.3547           4.3705
## mg.3  8.56369          2.85054        32     0.9997           3.3466
```

```r
summary(diag_dsd.asreg.result@objects[[1]])
```

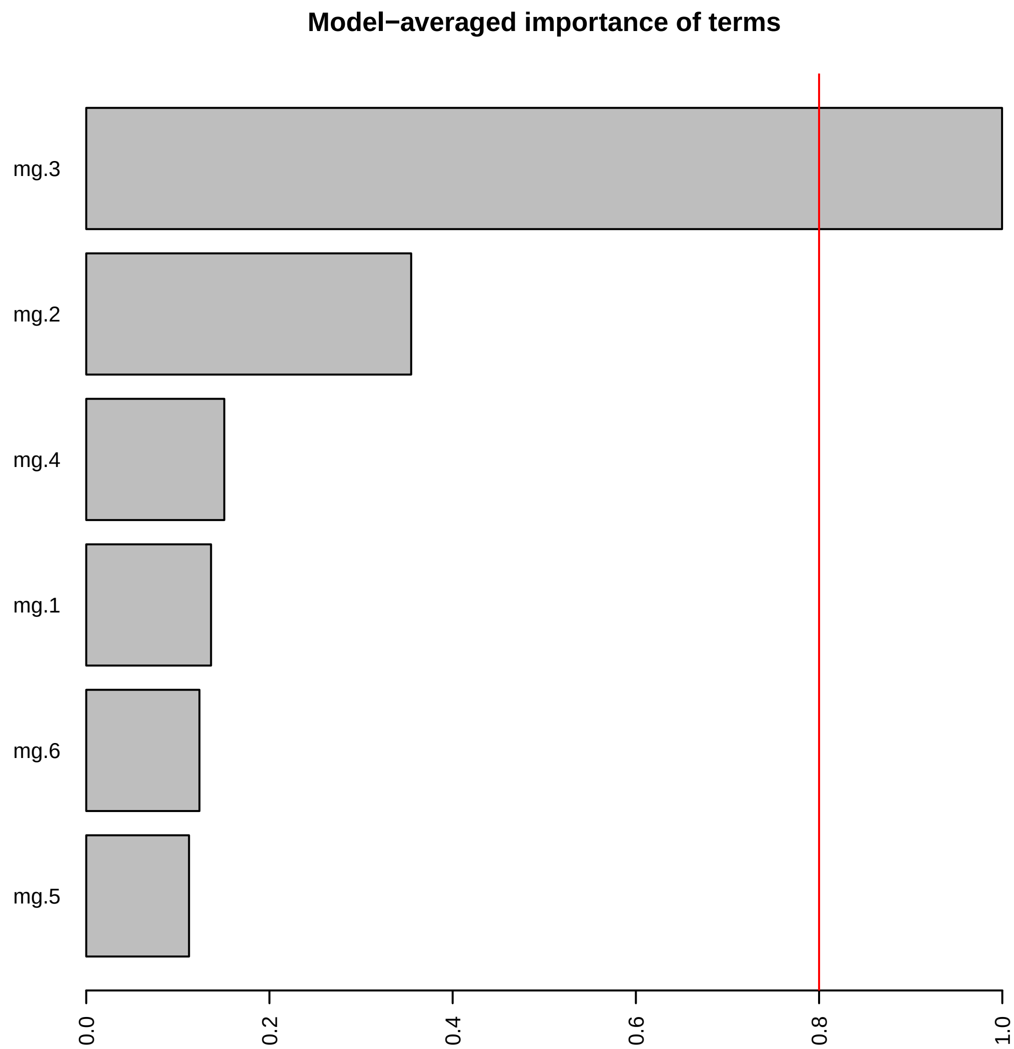
```
## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 110, number of events= 70
##
##         coef exp(coef) se(coef)    z Pr(>|z|)
## mg.3    8.18   3575.90     1.51 5.42  6.1e-08
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.3      3576    0.00028       185     69097
##
## Concordance= 0.692  (se = 0.037 )
## Rsquare= 0.202   (max possible= 0.995 )
## Likelihood ratio test= 24.8  on 1 df,   p=6.44e-07
## Wald test            = 29.3  on 1 df,   p=6.12e-08
## Score (logrank) test = 30.6  on 1 df,   p=3.14e-08
```

```r
plot(diag_dsd.asreg.result, type = "p")
```

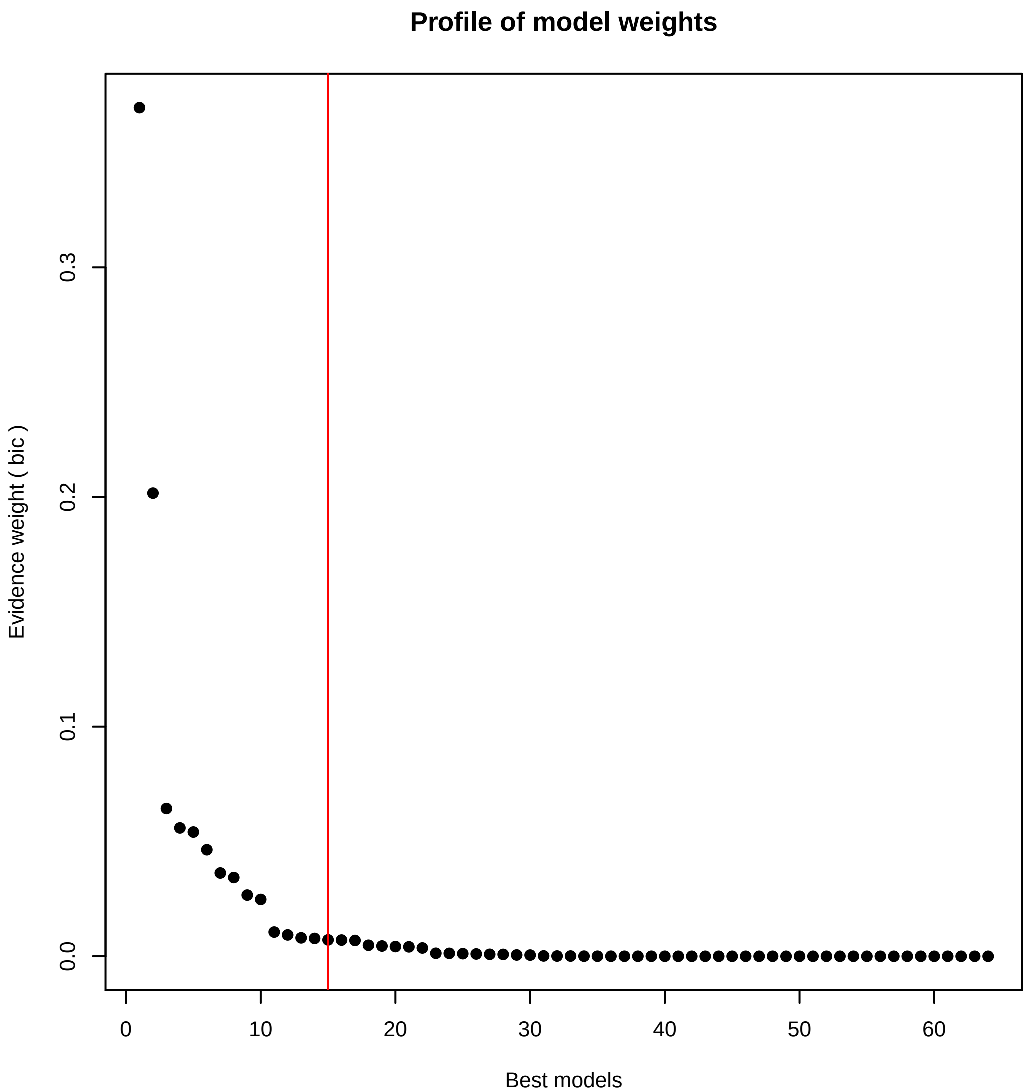
All-subsets regression




```
plot(diag_dsd.asreg.result, type = "s")
```




```
plot(diag_dsd.asreg.result, type = "w")
```



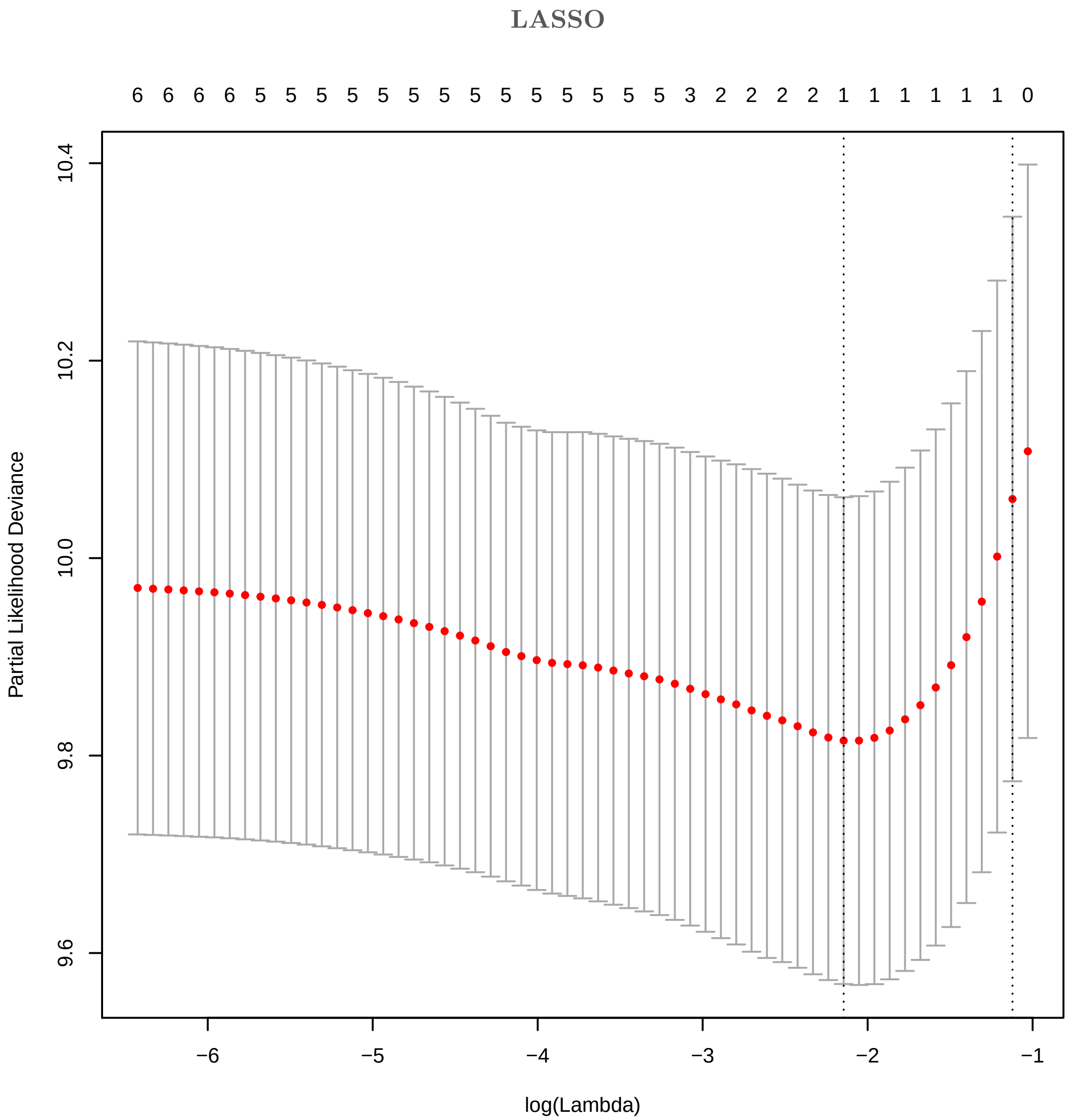

```
diag_dsd.glmnet.coef.1se
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##          1
## mg.1 .
## mg.2 .
## mg.3 0.948
## mg.4 .
## mg.5 .
## mg.6 .
```

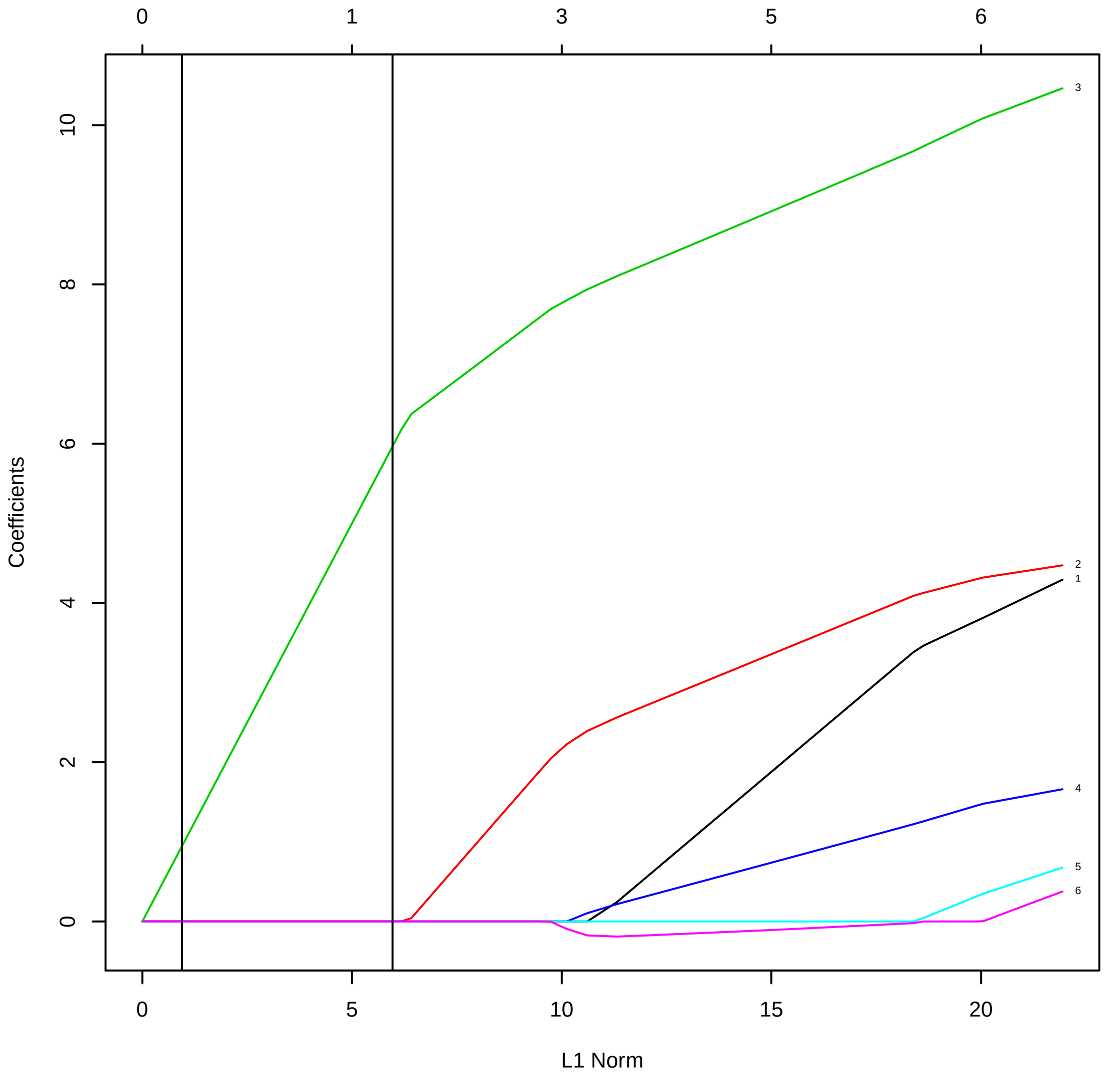
```
diag_dsd.glmnet.coef.min
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##          1
## mg.1 .
## mg.2 .
## mg.3 5.966
## mg.4 .
## mg.5 .
## mg.6 .
```

```r
plot(diag_dsd.glmnet.fit.cv)
```



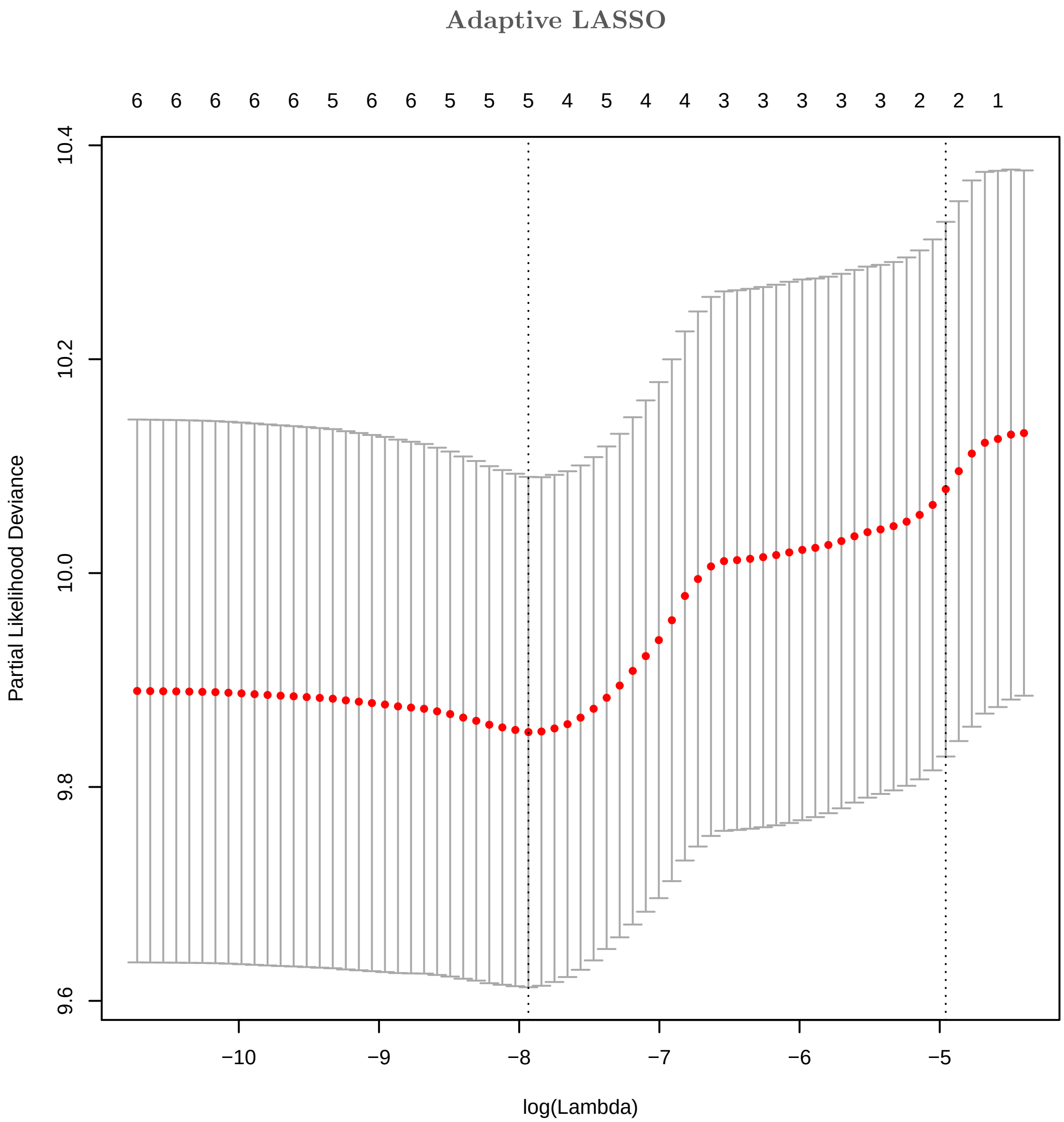

```r
plot(diag_dsd.glmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_dsd.glmnet.coef.1se)))
abline(v = sum(abs(diag_dsd.glmnet.coef.min)))
```



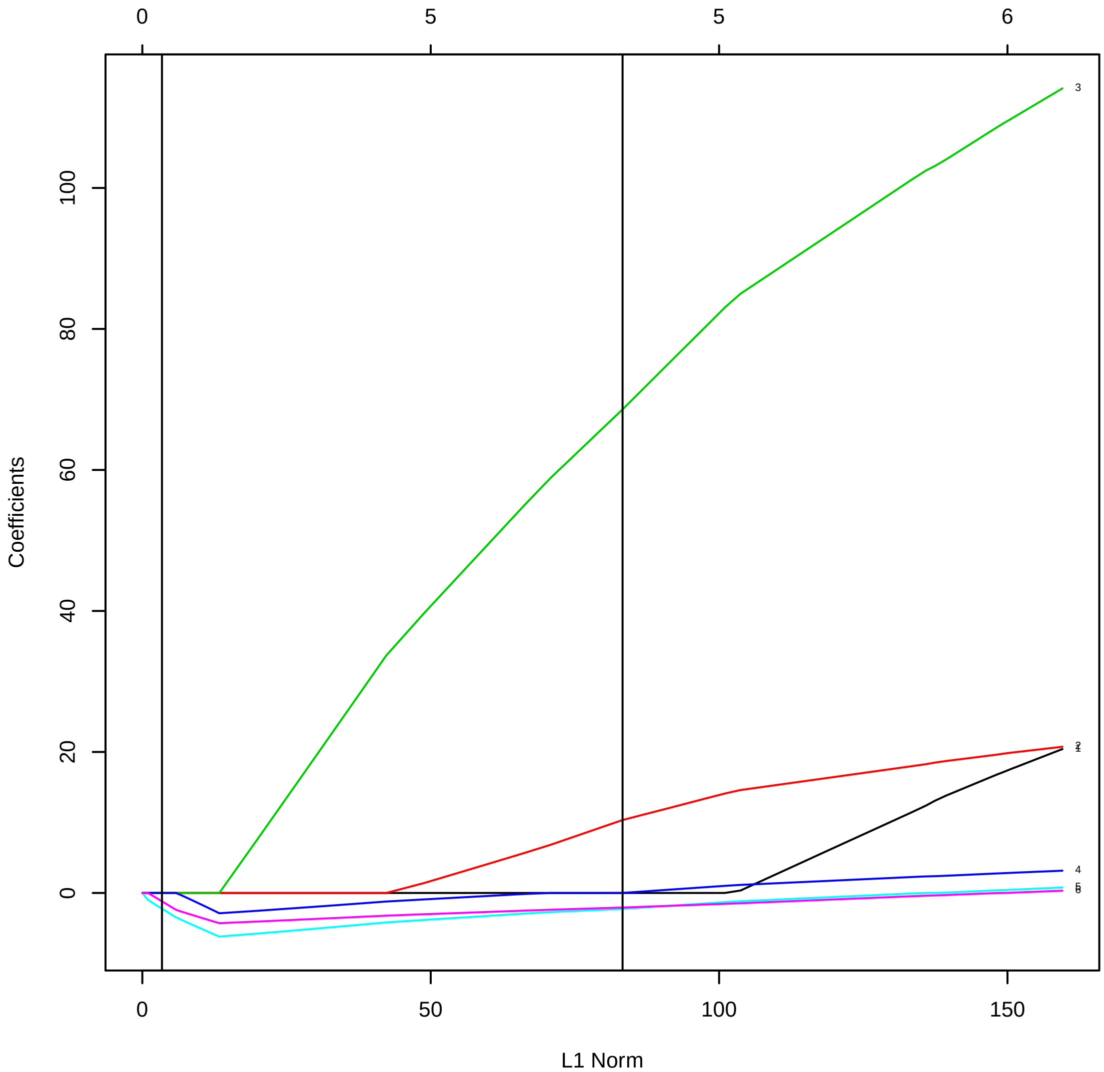

```
diag_dsd.adaglmnet.coef.1se/diag_dsd.adaglmnet.weights

## 6 x 1 sparse Matrix of class "dgCMatrix"
##            1
## mg.1  .
## mg.2  .
## mg.3  .
## mg.4  .
## mg.5 -2.3924
## mg.6 -0.9518

diag_dsd.adaglmnet.coef.min/diag_dsd.adaglmnet.weights
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##               1
## mg.1   .
## mg.2  48.23781
## mg.3 749.19334
## mg.4   0.01407
## mg.5  -2.43240
## mg.6  -1.65951
```

```r
plot(diag_dsd.adaglmnet.fit.cv)
```




```r
plot(diag_dsd.adaglmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(diag_dsd.adaglmnet.coef.1se)))
abline(v = sum(abs(diag_dsd.adaglmnet.coef.min)))
```



5.4.3 Outcome: Recurrence to disease-specific death

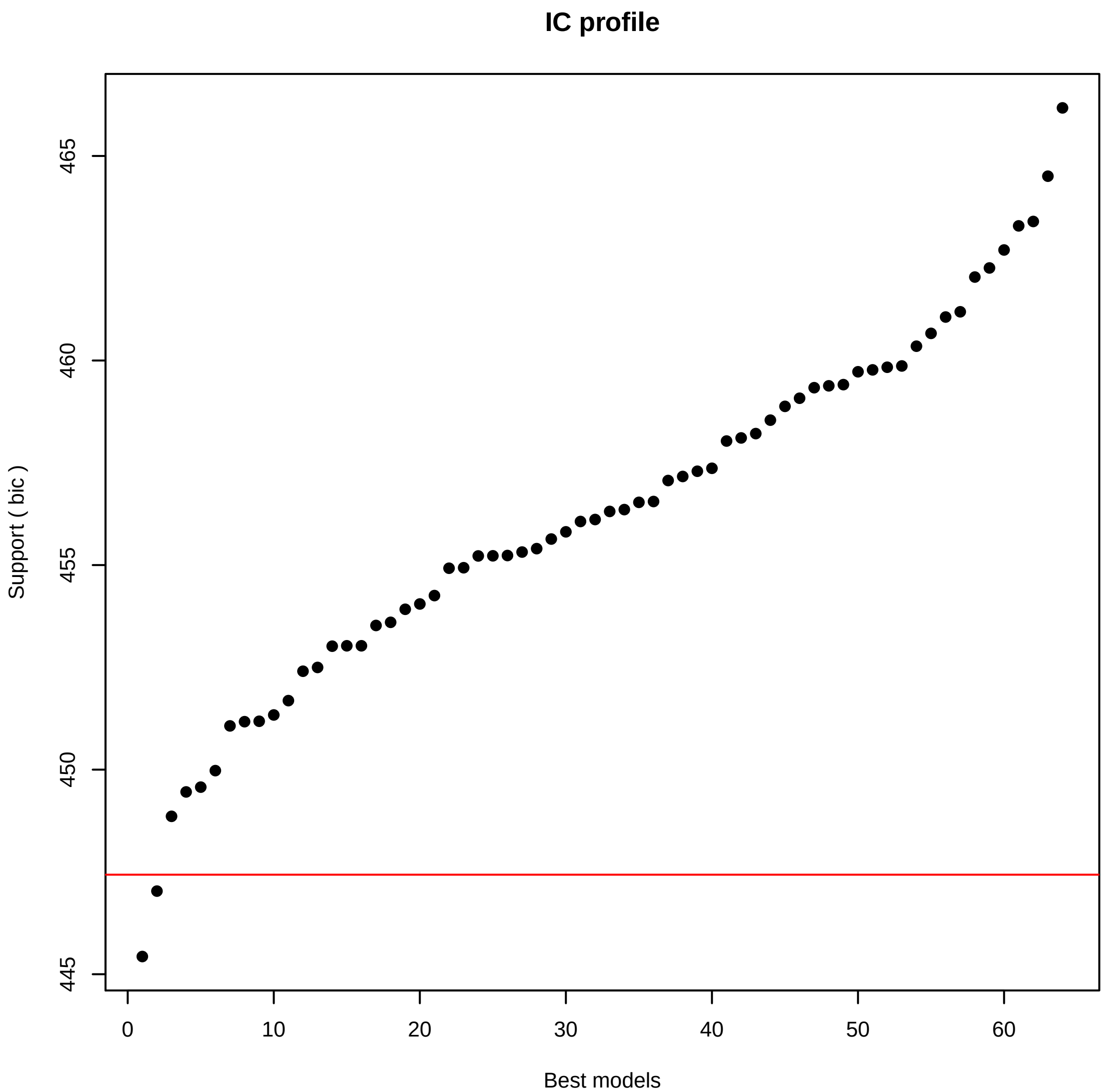
```
print(recr_dsd.asreg.result)
```

```
## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 64 models:
## Best IC: 445.432602899645
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.1 + mg.3"
## Evidence weight: 0.387098104422653
## Worst IC: 466.175045768034
## 2 models within 2 IC units.
## 20 models to reach 95% of evidence weight.
```

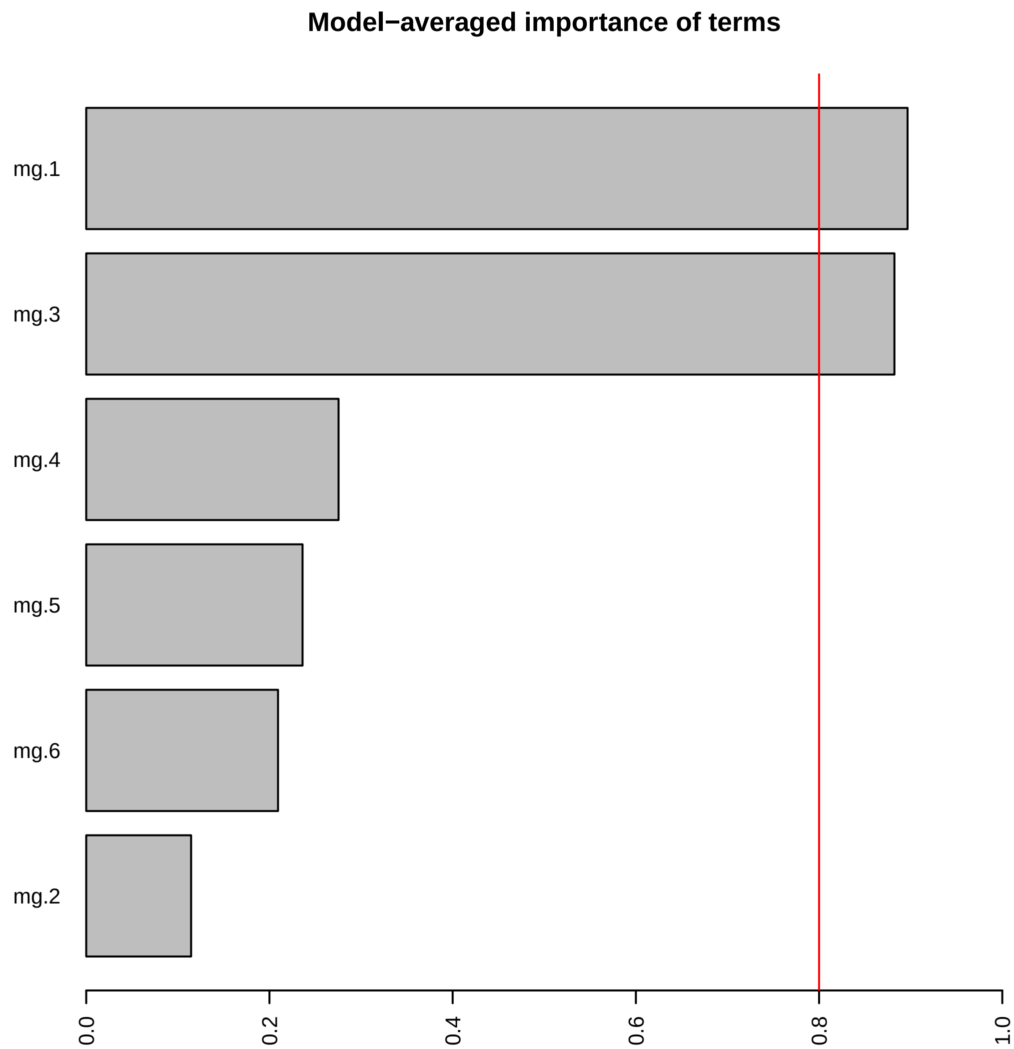
```r
coef(recr_dsd.asreg.result)

##      Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.2  0.01532           0.1068        32     0.1145           0.6507
## mg.6 -0.91314           3.2210        32     0.2094           3.5728
## mg.5 -0.81092           2.3886        32     0.2362           3.0767
## mg.4  0.64503           1.2720        32     0.2755           2.2452
## mg.3  5.81157           6.8939        32     0.8822           5.2269
## mg.1 14.08010          34.8968        32     0.8966          11.7598

summary(recr_dsd.asreg.result@objects[[1]])

## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 81, number of events= 64
##
##          coef exp(coef) se(coef)    z Pr(>|z|)
## mg.1 1.62e+01  1.05e+07 4.32e+00 3.74  0.00018
## mg.3 6.57e+00  7.10e+02 1.80e+00 3.65  0.00026
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.1  10506432   9.52e-08    2213.1  4.99e+10
## mg.3       710   1.41e-03      20.9  2.41e+04
##
## Concordance= 0.682  (se = 0.041 )
## Rsquare= 0.228   (max possible= 0.997 )
## Likelihood ratio test= 21  on 2 df,   p=2.76e-05
## Wald test            = 22.4  on 2 df,   p=1.36e-05
## Score (logrank) test = 23.7  on 2 df,   p=7.07e-06
```

```r
plot(recr_dsd.asreg.result, type = "p")
```

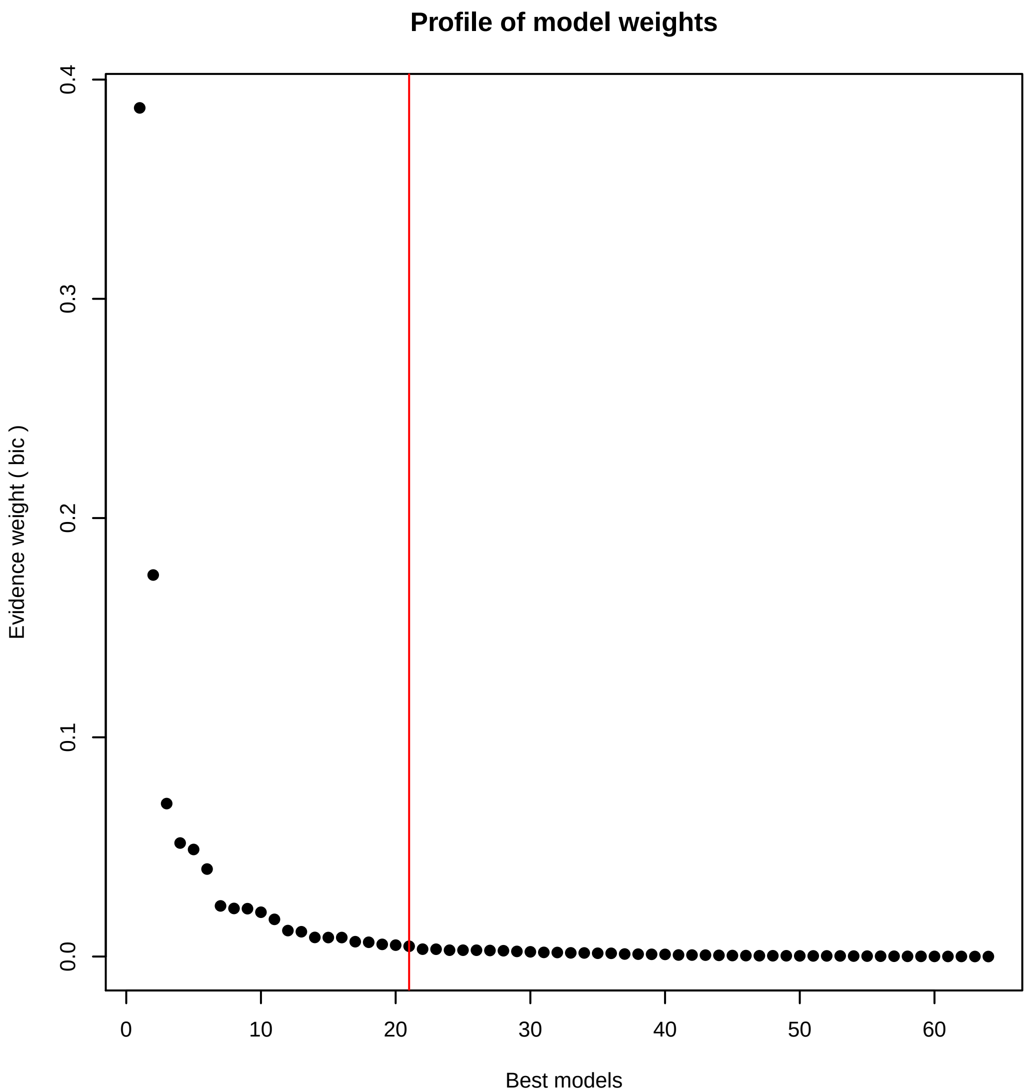
All-subsets regression




```r
plot(recr_dsd.asreg.result, type = "s")
```




```
plot(recr_dsd.asreg.result, type = "w")
```



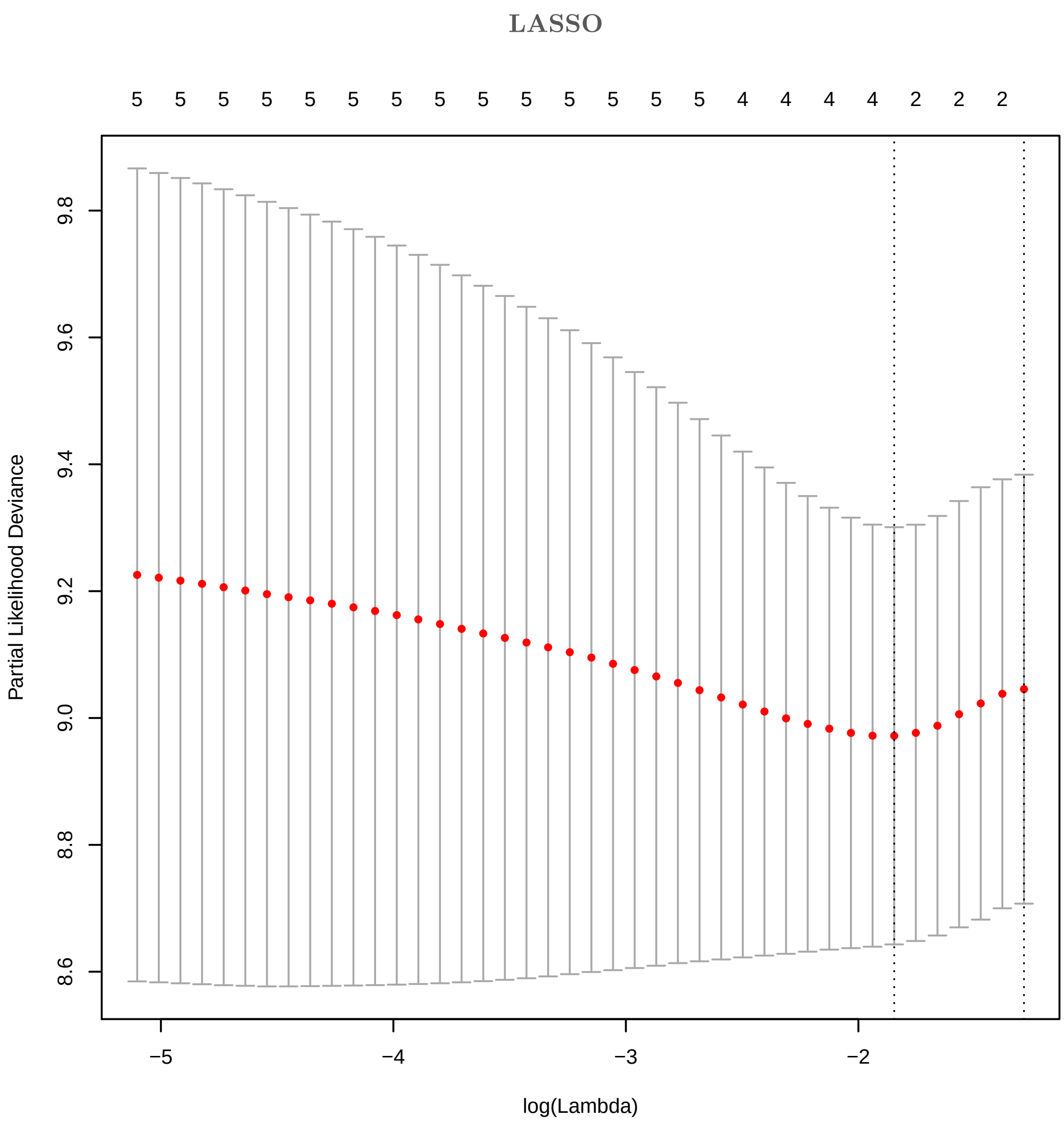

```
recr_dsd.glmnet.coef.1se
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##      1
## mg.1 .
## mg.2 .
## mg.3 .
## mg.4 .
## mg.5 .
## mg.6 .
```

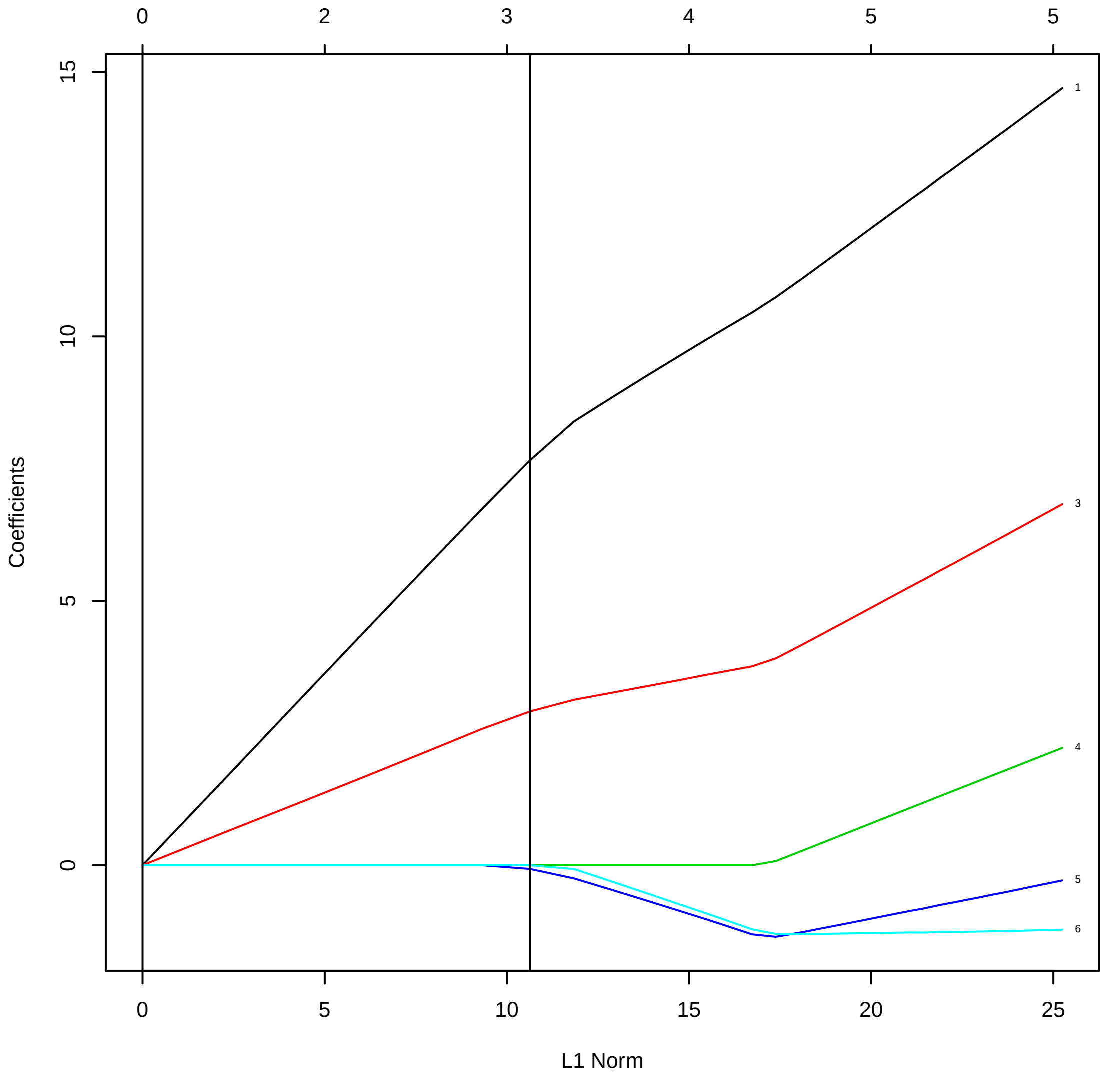
```
recr_dsd.glmnet.coef.min
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##              1
## mg.1  7.65806
## mg.2  .
## mg.3  2.90888
## mg.4  .
## mg.5 -0.06988
## mg.6  .
```

```
plot(recr_dsd.glmnet.fit.cv)
```




```
plot(recr_dsd.glmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(recr_dsd.glmnet.coef.1se)))
abline(v = sum(abs(recr_dsd.glmnet.coef.min)))
```



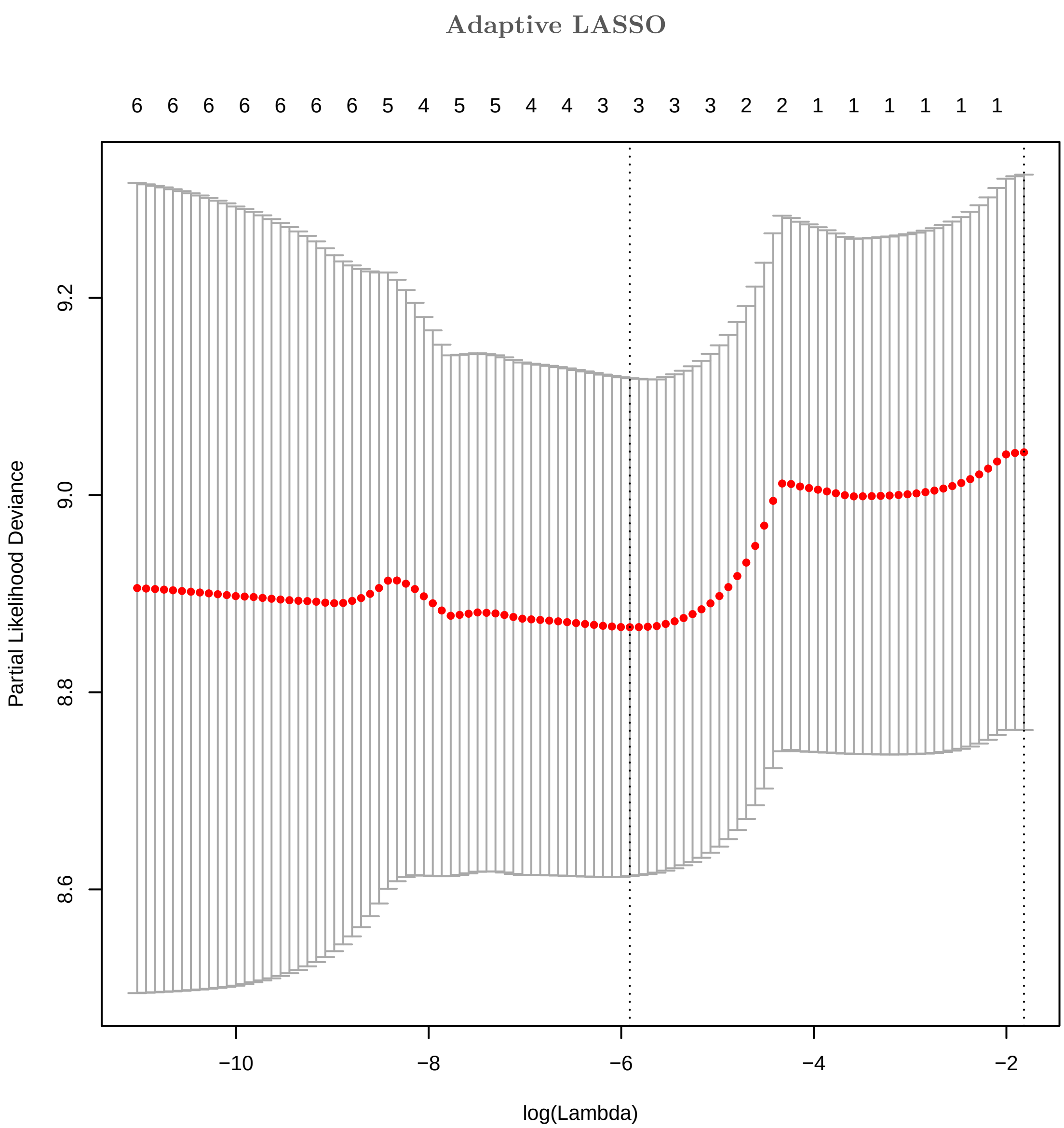

```
recr_dsd.adaglmnet.coef.1se/recr_dsd.adaglmnet.weights
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##      1
## mg.1 .
## mg.2 .
## mg.3 .
## mg.4 .
## mg.5 .
## mg.6 .
```

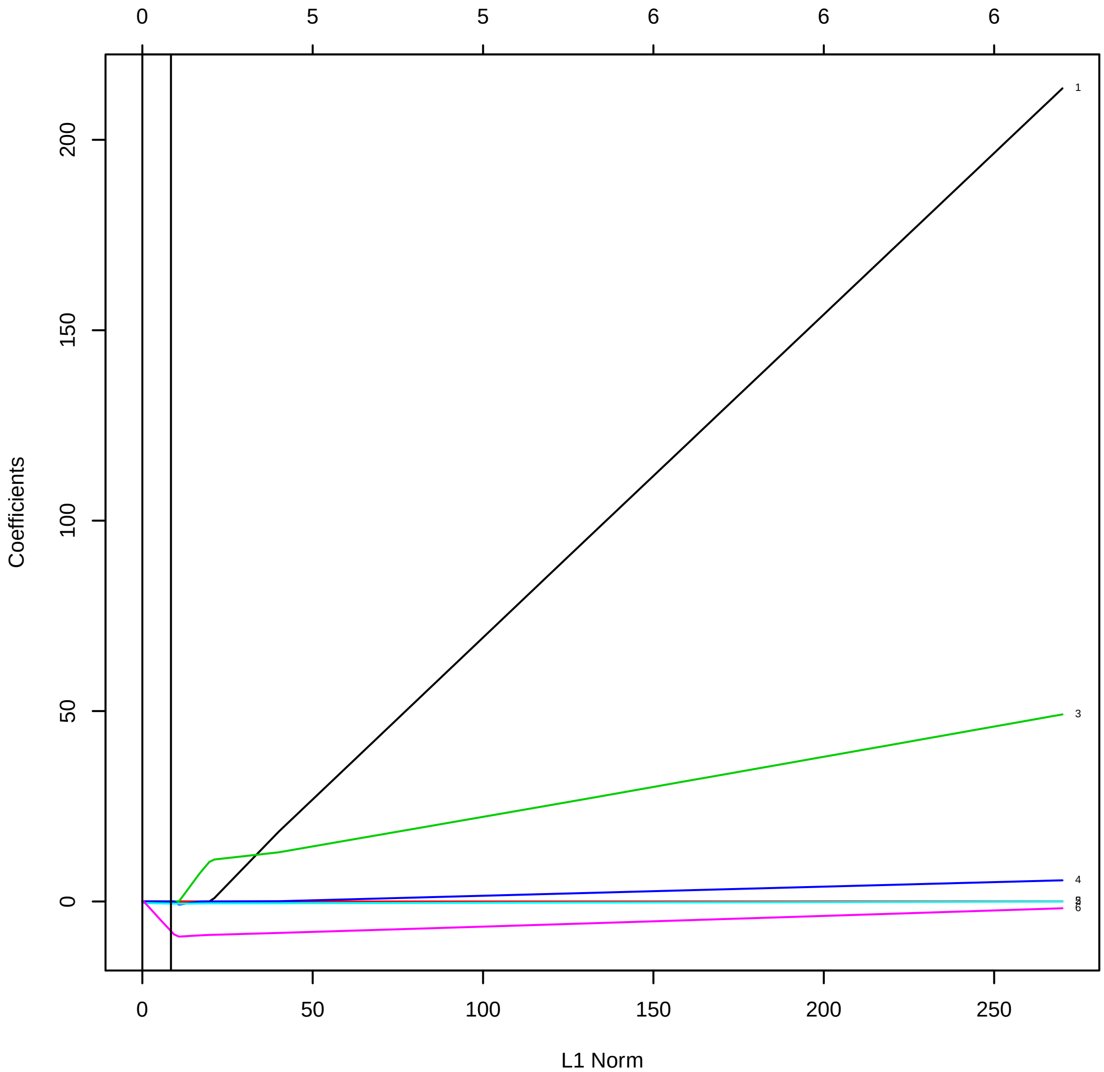
```
recr_dsd.adaglmnet.coef.min/recr_dsd.adaglmnet.weights
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##              1
## mg.1  .
## mg.2 -0.01217
## mg.3  .
## mg.4  .
## mg.5 -0.05456
## mg.6 -8.97073
```

```
plot(recr_dsd.adaglmnet.fit.cv)
```




```
plot(recr_dsd.adaglmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(recr_dsd.adaglmnet.coef.1se)))
abline(v = sum(abs(recr_dsd.adaglmnet.coef.min)))
```



6 Session information

```
session_info
```

```
## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_AU.UTF-8          LC_NUMERIC=C
##  [3] LC_TIME=en_AU.UTF-8           LC_COLLATE=en_AU.UTF-8
##  [5] LC_MONETARY=en_AU.UTF-8       LC_MESSAGES=en_AU.UTF-8
##  [7] LC_PAPER=en_AU.UTF-8          LC_NAME=en_AU.UTF-8
##  [9] LC_ADDRESS=en_AU.UTF-8        LC_TELEPHONE=en_AU.UTF-8
## [11] LC_MEASUREMENT=en_AU.UTF-8    LC_IDENTIFICATION=en_AU.UTF-8
##
## attached base packages:
```

```
## [1] parallel  splines   stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] doParallel_1.0.8    iterators_1.0.7     foreach_1.4.2
##  [4] NMF_0.20.5          Biobase_2.26.0      BiocGenerics_0.12.1
##  [7] cluster_1.15.3      rngtools_1.2.4      pkgmaker_0.22
## [10] registry_0.2        ahaz_1.14           survival_2.37-7
## [13] gplots_2.14.2       RColorBrewer_1.0-5  energy_1.6.2
## [16] glmnet_1.9-8        Matrix_1.1-4        glmulti_1.0.7
## [19] rJava_0.9-6
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6       boot_1.3-13        caTools_1.17.1
##  [4] codetools_0.2-9    colorspace_1.2-4   compiler_3.1.1
##  [7] digest_0.6.4       gdata_2.13.3       ggplot2_1.0.0
## [10] grid_3.1.1         gridBase_0.4-7     gtable_0.1.2
## [13] gtools_3.4.1       KernSmooth_2.23-13 lattice_0.20-29
## [16] MASS_7.3-35        munsell_0.4.2      plyr_1.8.1
## [19] proto_0.3-10       Rcpp_0.11.3        reshape2_1.4
## [22] scales_0.2.4       stringr_0.6.2      tools_3.1.1
## [25] xtable_1.7-4
```

```
sessionInfo()
```

```
## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_AU.UTF-8          LC_NUMERIC=C
##  [3] LC_TIME=en_AU.UTF-8           LC_COLLATE=en_AU.UTF-8
##  [5] LC_MONETARY=en_AU.UTF-8       LC_MESSAGES=en_AU.UTF-8
##  [7] LC_PAPER=en_AU.UTF-8          LC_NAME=en_AU.UTF-8
##  [9] LC_ADDRESS=en_AU.UTF-8        LC_TELEPHONE=en_AU.UTF-8
## [11] LC_MEASUREMENT=en_AU.UTF-8    LC_IDENTIFICATION=en_AU.UTF-8
##
## attached base packages:
## [1] parallel  methods   splines   stats     graphics  grDevices utils
## [8] datasets  base
##
## other attached packages:
##  [1] stargazer_5.1       xtable_1.7-4        gplots_2.14.2
##  [4] RColorBrewer_1.0-5  glmnet_1.9-8        Matrix_1.1-4
##  [7] glmulti_1.0.7       rJava_0.9-6         NMF_0.20.5
## [10] Biobase_2.26.0      BiocGenerics_0.12.1 cluster_1.15.3
## [13] rngtools_1.2.4      pkgmaker_0.22       registry_0.2
## [16] energy_1.6.2        survival_2.37-7     knitr_1.8
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6       boot_1.3-13        caTools_1.17.1
##  [4] codetools_0.2-9    colorspace_1.2-4   digest_0.6.4
##  [7] doParallel_1.0.8   evaluate_0.5.5     foreach_1.4.2
## [10] formatR_1.0        gdata_2.13.3       ggplot2_1.0.0
## [13] grid_3.1.1         gridBase_0.4-7     gtable_0.1.2
```

```
## [16] gtools_3.4.1       highr_0.4          iterators_1.0.7
## [19] KernSmooth_2.23-13 labeling_0.3       lattice_0.20-29
## [22] MASS_7.3-35        munsell_0.4.2      plyr_1.8.1
## [25] proto_0.3-10       Rcpp_0.11.3        reshape2_1.4
## [28] scales_0.2.4       stringr_0.6.2      tools_3.1.1
```