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 [OK] | Cores 63/64

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)

## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
##
## Attaching package:  'gplots'
##
## The following object is masked from 'package:stats':
##
##    lowess
```

```
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

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

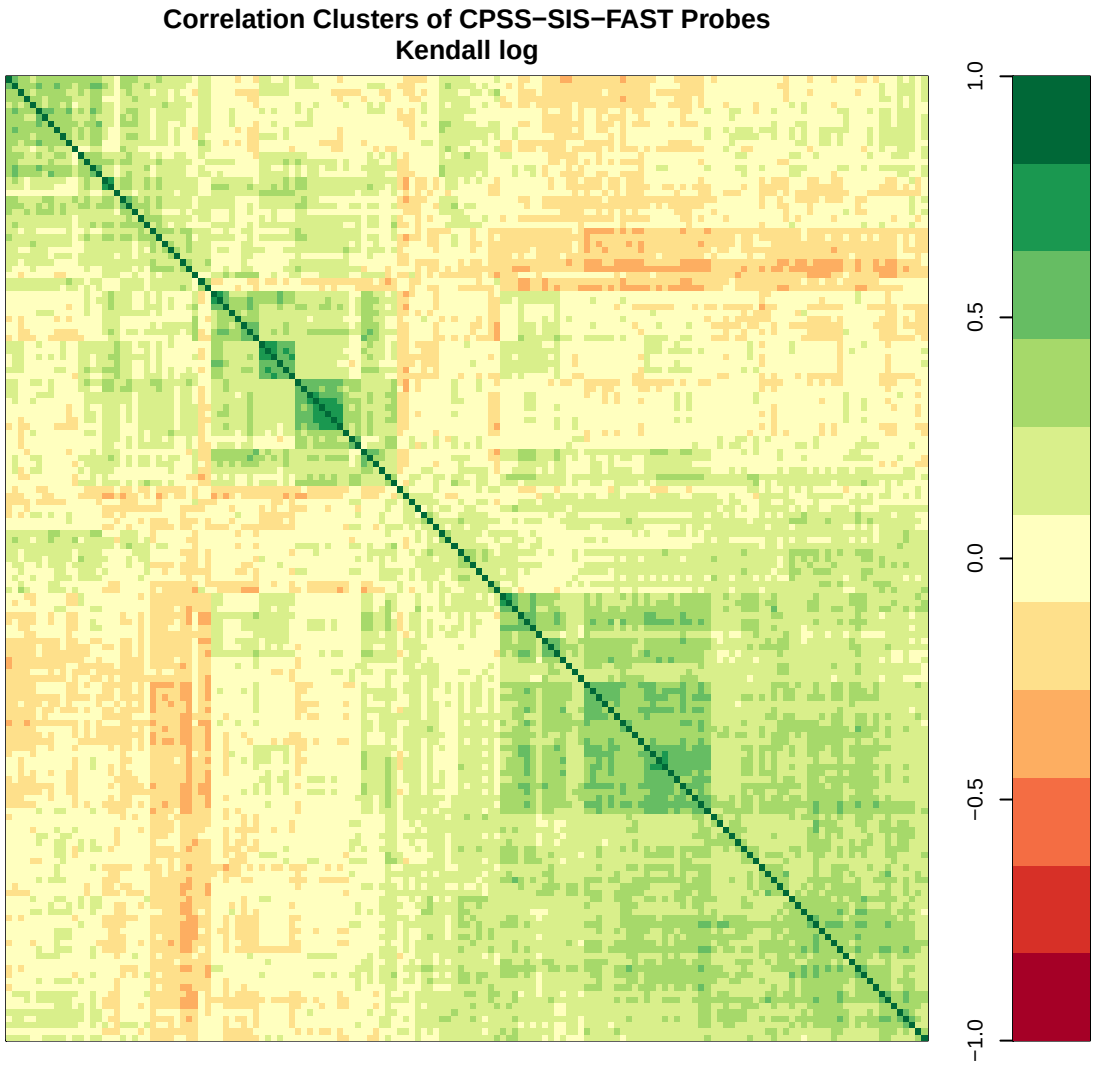
```
##
## FALSE  TRUE
## 12847   153
```

```
mean(cpss.sis$sel)
```

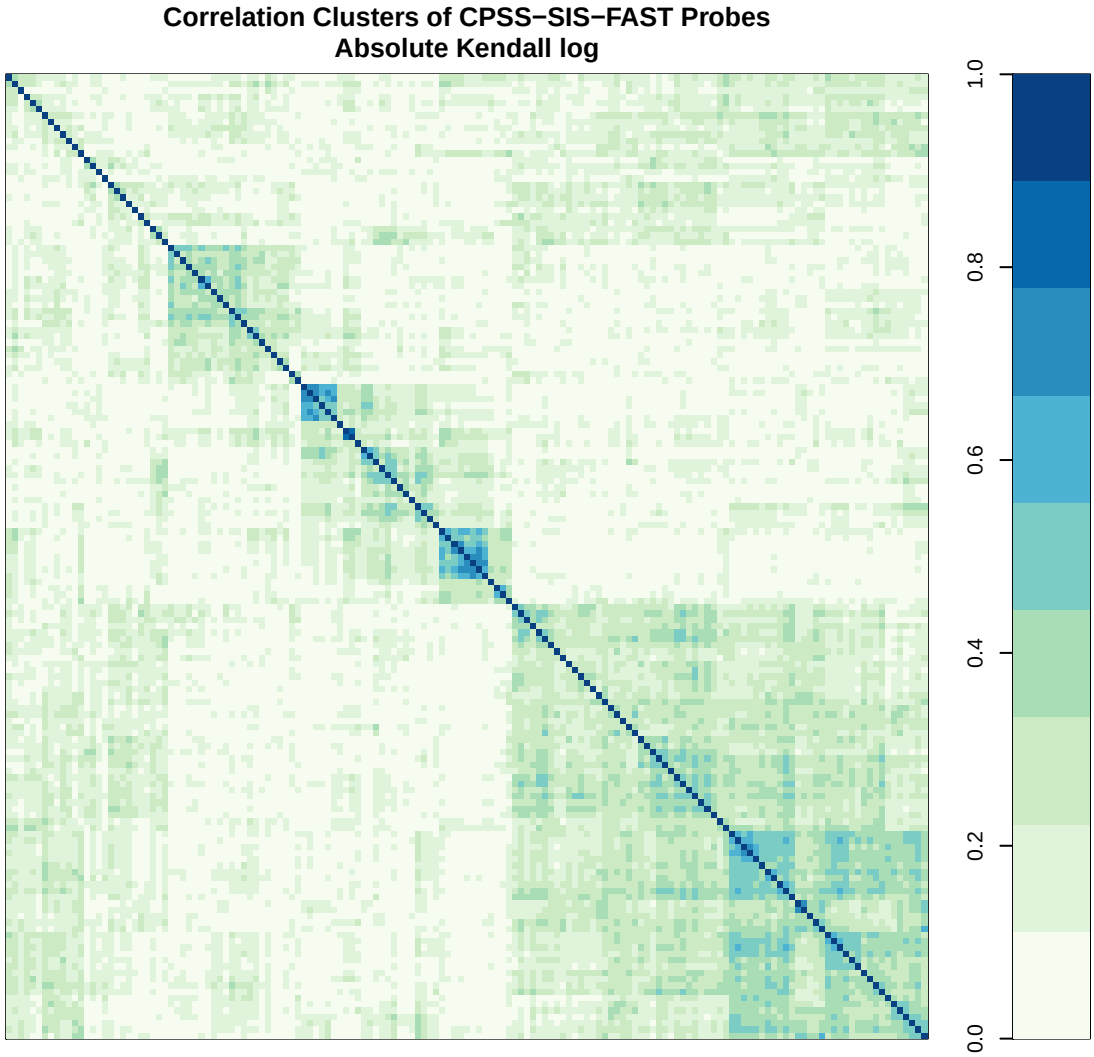
```
## [1] 0.01177
```

3 Expression correlation

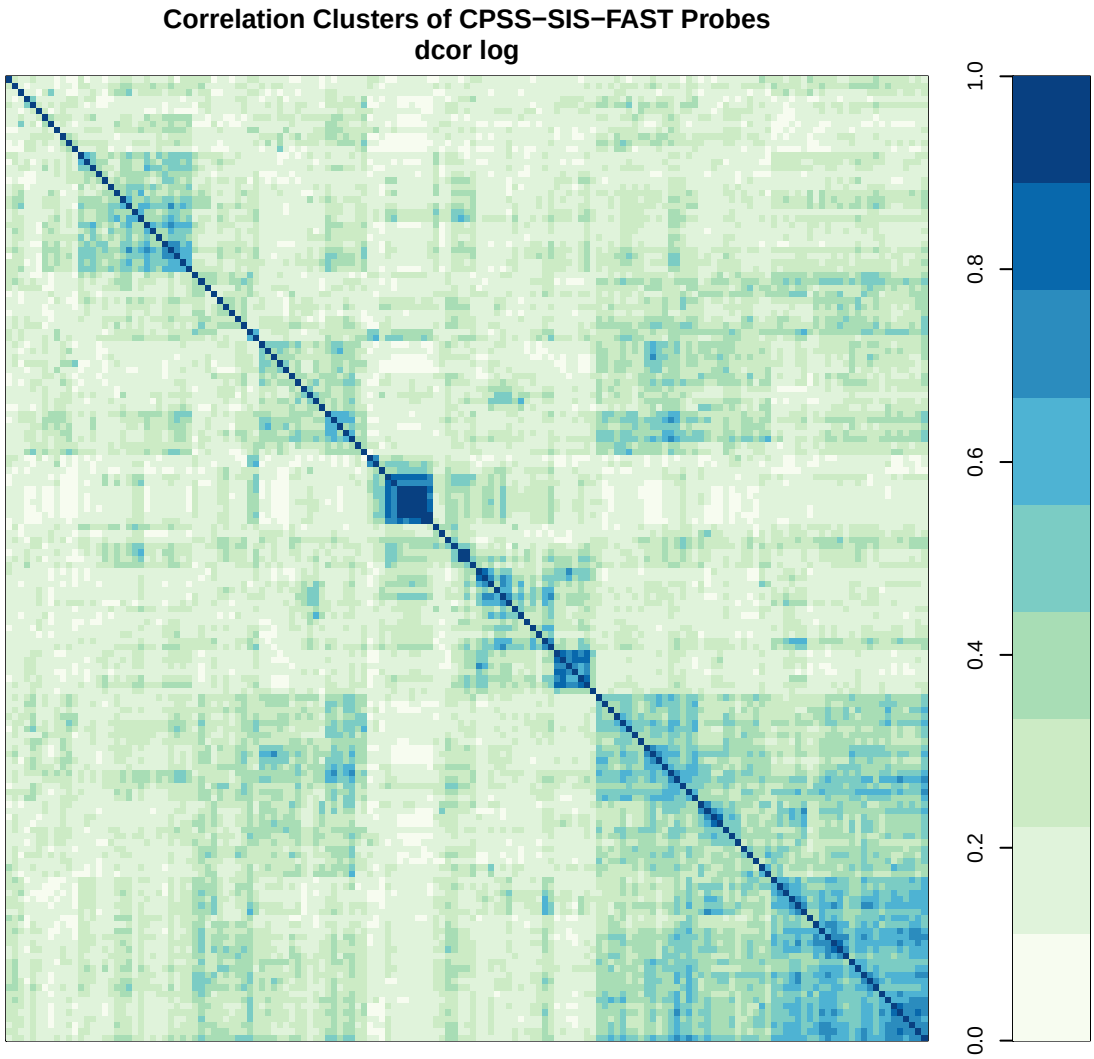
```
corPlot(x.sel.kcor, main = "Correlation Clusters of CPSS-SIS-FAST Probes\nKendall log",
    useRaster = FALSE)
```




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



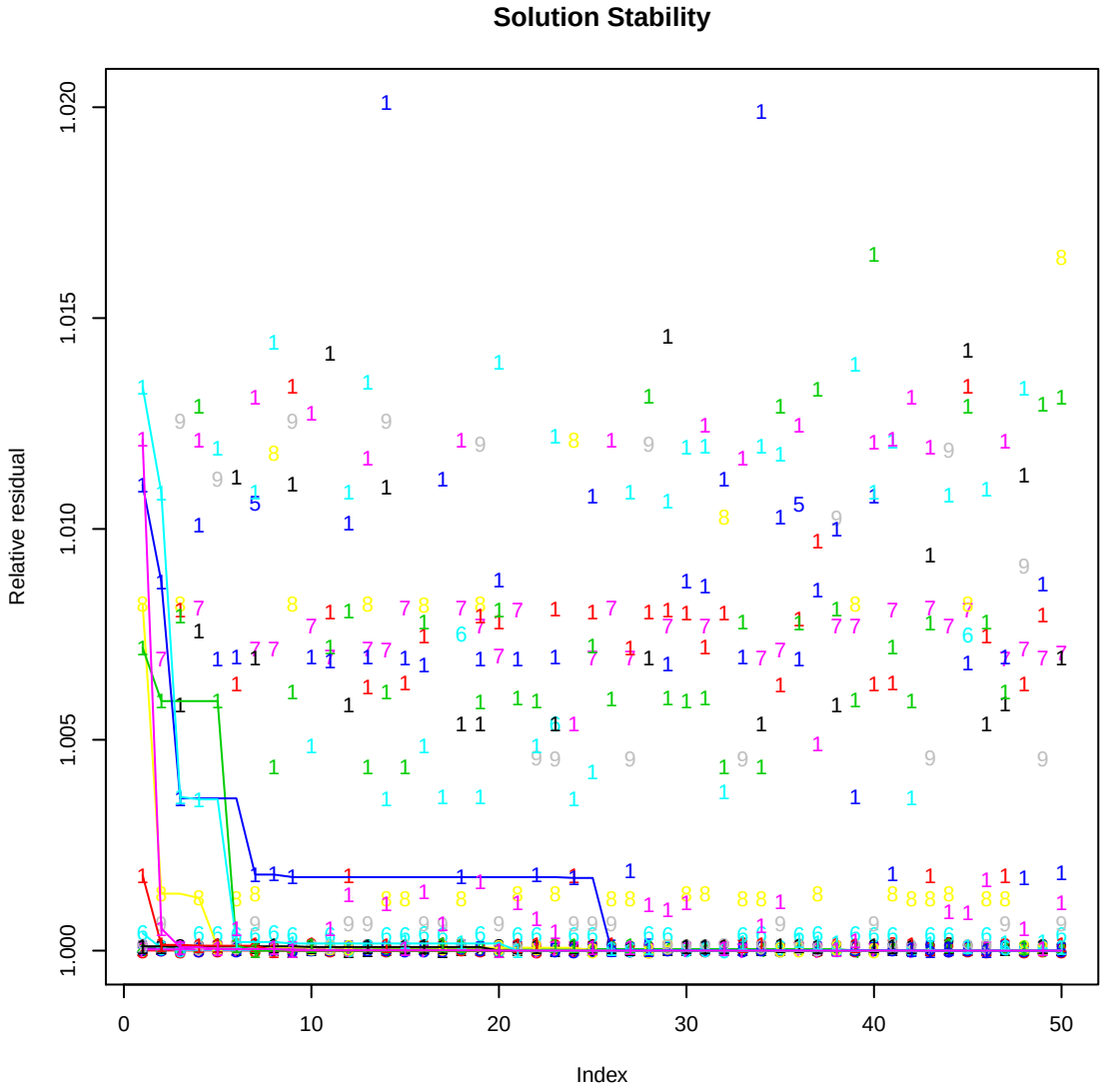

```
corPlot(x.sel.dcor, zlim = c(0, 1), pal = "GnBu", main = "Correlation Clusters of CPSS-SIS-FAST Probes\n
    useRaster = FALSE)
```



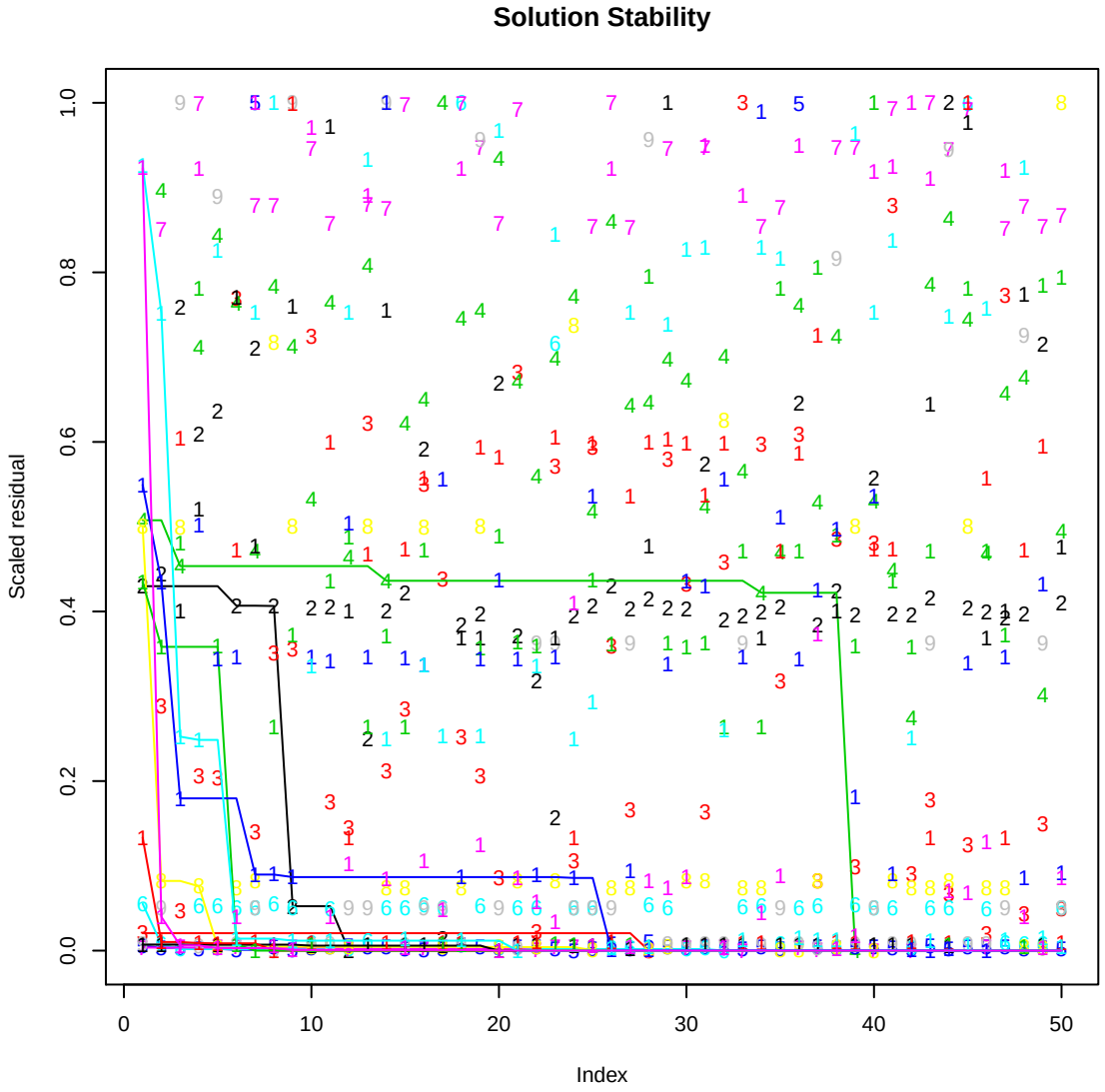
4 Factorization

4.1 Rank estimation

```
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)
}
```

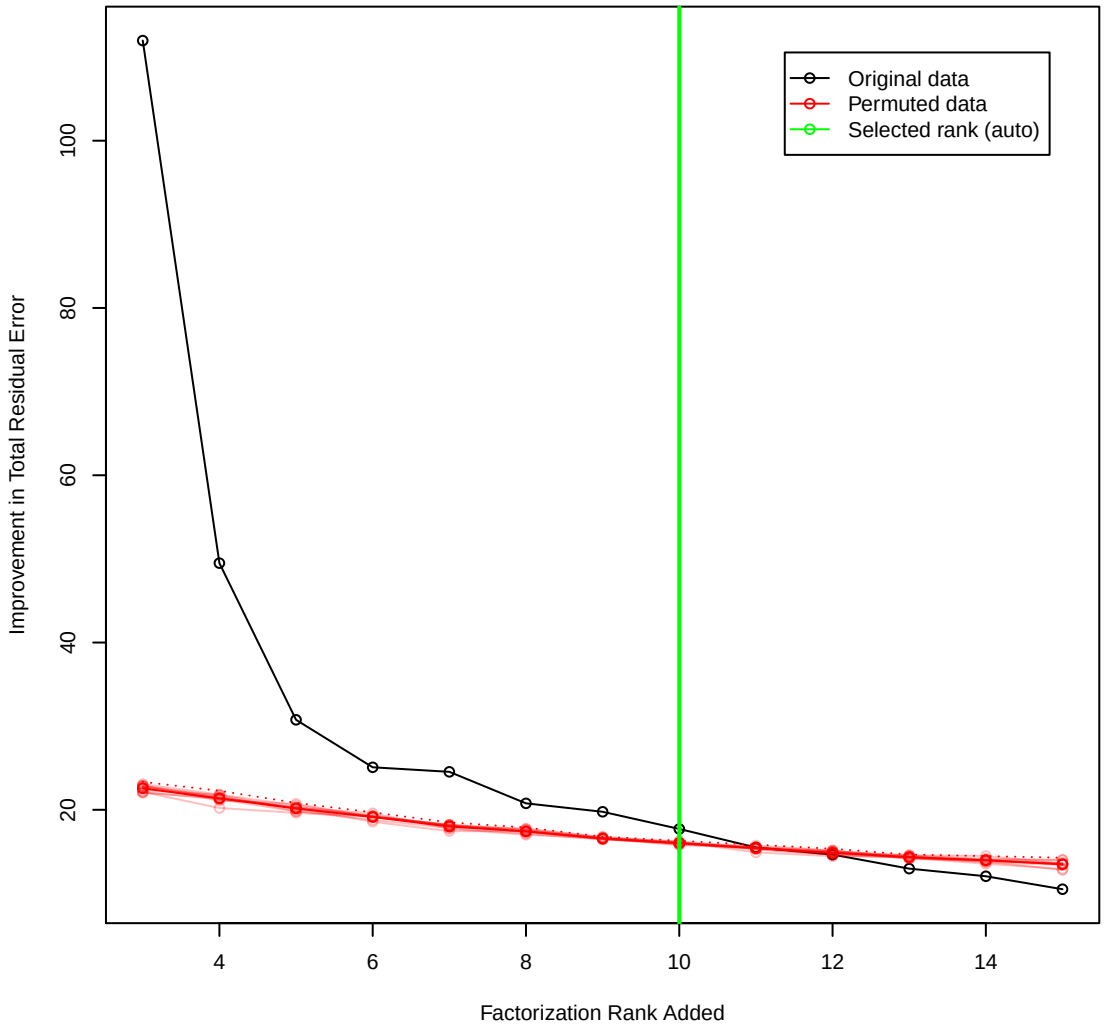



```
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_scaled),
    ylab = "Scaled residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_scaled[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_scaled[,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



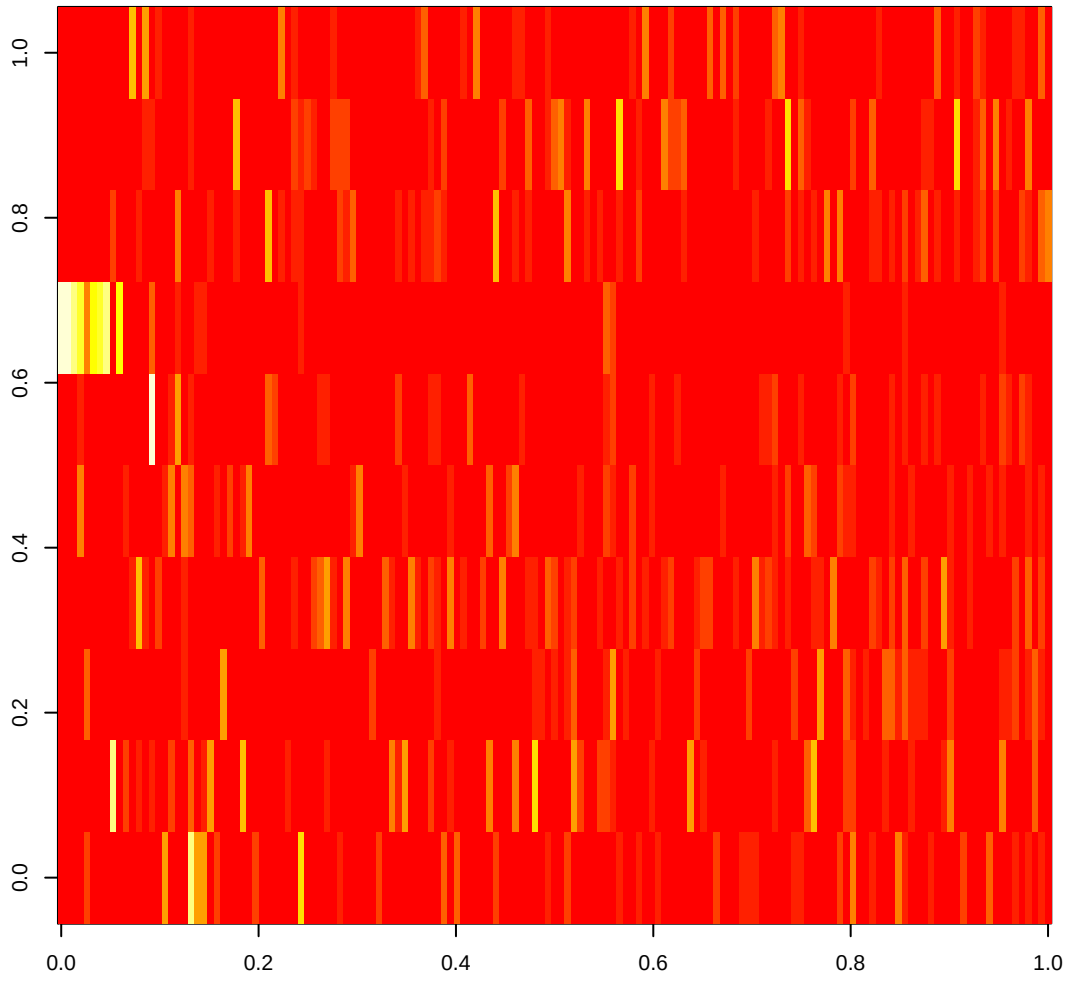

```
plot(nmf.rankrange[-1], -temp.orig_resids.delta, type = "o", col = "black",
    pch = 21, ylim = range(-c(temp.orig_resids.delta, temp.perm_resids.delta.mean)),
    xlab = "Factorization Rank Added", ylab = "Improvement in Total Residual Error")
lines(nmf.rankrange[-1], -temp.perm_resids.delta.mean, 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(temp.col == "green", "auto", "fixed"))), col = c("black", "red",
    temp.col), lty = "solid", pch = 21, inset = 0.05)
```

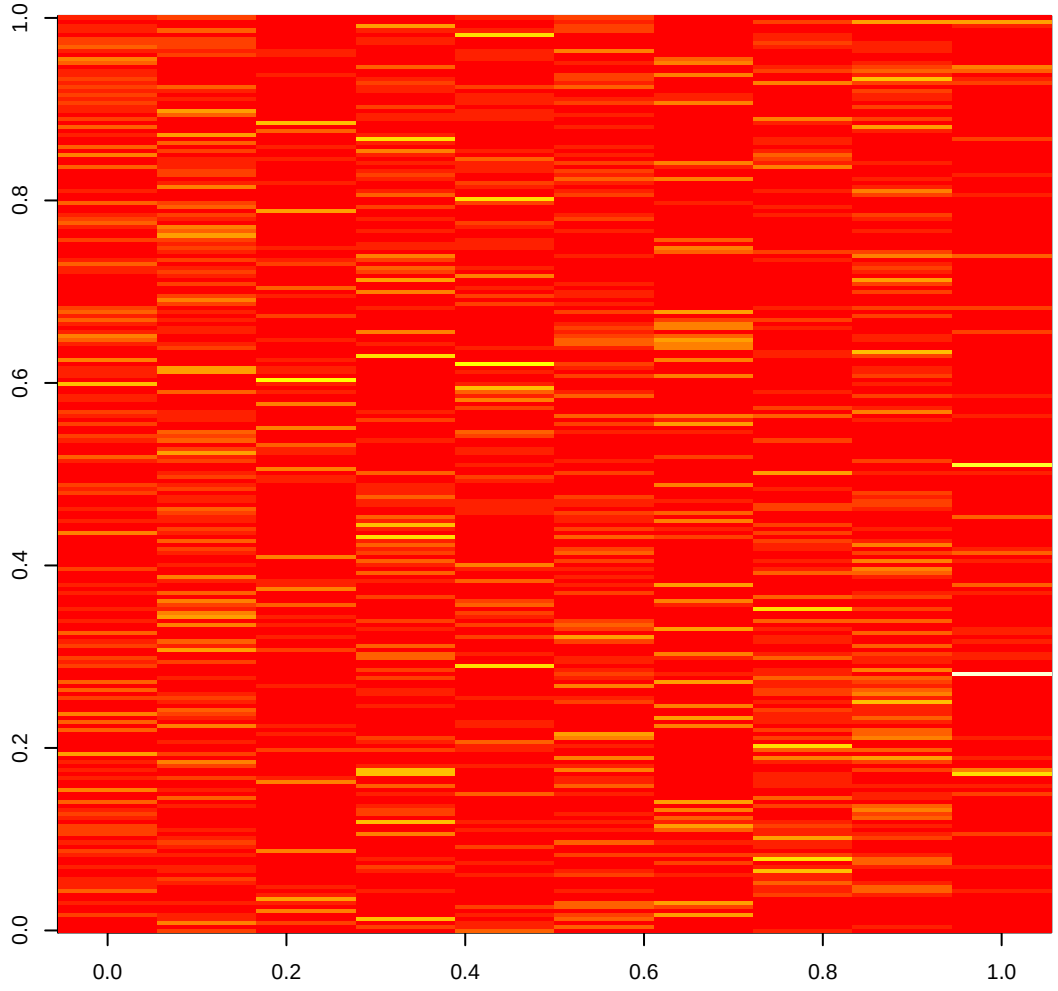


4.2 Fit

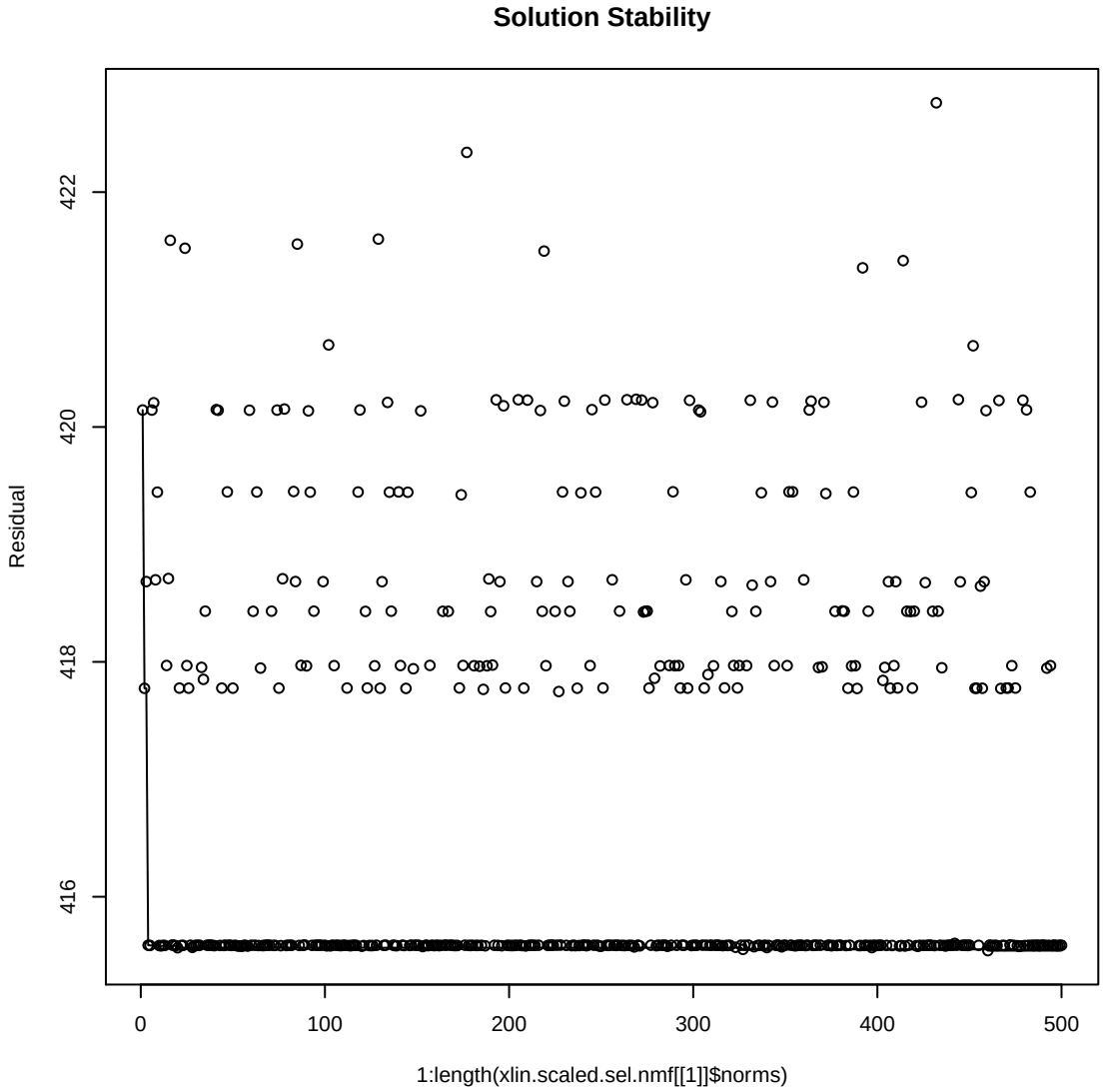
```
image(xlin.scaled.sel.nmf[[1]]$best_fit$W)
```




```
image(xlin.scaled.sel.nmf[[1]]$best_fit$H)
```




```
plot(1:length(xlin.scaled.sel.nmf[[1]]$norms), xlin.scaled.sel.nmf[[1]]$norms,
    ylab = "Residual", main = "Solution Stability")
lines(1:length(xlin.scaled.sel.nmf[[1]]$norms), cummin(xlin.scaled.sel.nmf[[1]]$norms))
```



4.3 Component CPV associations

4.3.1 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] -3.5570    0.0285   1.9751 -1.8    0.072
##
##                     exp(coef) exp(-coef) lower .95 upper .95
```

```
## coefs.diag_rec[, i]    0.0285       35.1  0.000594      1.37
##
## Concordance= 0.572  (se = 0.036 )
## Rsquare= 0.033   (max possible= 0.997 )
## Likelihood ratio test= 3.44  on 1 df,   p=0.0637
## Wald test            = 3.24  on 1 df,   p=0.0717
## Score (logrank) test = 3.28  on 1 df,   p=0.0703
##
## 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] -3.542     0.029    1.775 -2    0.046
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.029       34.5  0.000894     0.939
##
## Concordance= 0.566  (se = 0.036 )
## Rsquare= 0.041   (max possible= 0.997 )
## Likelihood ratio test= 4.31  on 1 df,   p=0.038
## Wald test            = 3.98  on 1 df,   p=0.046
## Score (logrank) test = 4.03  on 1 df,   p=0.0447
##
## 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] -1.01e+01  4.09e-05  4.96e+00 -2.04    0.042
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]  4.09e-05      24458  2.43e-09     0.687
##
## Concordance= 0.552  (se = 0.036 )
## Rsquare= 0.043   (max possible= 0.997 )
## Likelihood ratio test= 4.58  on 1 df,   p=0.0323
## Wald test            = 4.14  on 1 df,   p=0.0418
## Score (logrank) test = 4.2  on 1 df,   p=0.0405
##
## 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]    7.23   1378.00     1.82 3.96  7.4e-05
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      1378   0.000726      38.6     49193
##
```

```
## Concordance= 0.654  (se = 0.036 )
## Rsquare= 0.118   (max possible= 0.997 )
## Likelihood ratio test= 13  on 1 df,   p=0.000305
## Wald test            = 15.7  on 1 df,   p=7.41e-05
## Score (logrank) test = 16.2  on 1 df,   p=5.65e-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] -6.86920   0.00104  4.02468 -1.71    0.088
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]   0.00104        962   3.9e-07      2.77
##
## Concordance= 0.531  (se = 0.036 )
## Rsquare= 0.03   (max possible= 0.997 )
## Likelihood ratio test= 3.15  on 1 df,   p=0.0758
## Wald test            = 2.91  on 1 df,   p=0.0879
## Score (logrank) test = 2.93  on 1 df,   p=0.0872
##
## 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]   6.74    841.80     2.30 2.93   0.0034
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]       842    0.00119       9.3     76174
##
## Concordance= 0.589  (se = 0.036 )
## Rsquare= 0.074   (max possible= 0.997 )
## Likelihood ratio test= 7.98  on 1 df,   p=0.00473
## Wald test            = 8.59  on 1 df,   p=0.00339
## Score (logrank) test = 8.73  on 1 df,   p=0.00313
##
## 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] -2.3624    0.0942   1.5579 -1.52     0.13
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]    0.0942       10.6   0.00445         2
##
## Concordance= 0.582  (se = 0.035 )
## Rsquare= 0.024   (max possible= 0.997 )
```

```
## Likelihood ratio test= 2.5  on 1 df,   p=0.114
## Wald test            = 2.3  on 1 df,   p=0.129
## Score (logrank) test = 2.33  on 1 df,   p=0.127
##
## 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]  4.26     70.85     1.26 3.37  0.00075
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      70.8     0.0141      5.95       844
##
## Concordance= 0.62  (se = 0.036 )
## Rsquare= 0.084   (max possible= 0.997 )
## Likelihood ratio test= 9.12  on 1 df,   p=0.00253
## Wald test            = 11.4  on 1 df,   p=0.000751
## Score (logrank) test = 11.9  on 1 df,   p=0.00055
##
## 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]  3.39     29.55     1.76 1.92    0.054
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      29.6     0.0338     0.939       930
##
## Concordance= 0.556  (se = 0.036 )
## Rsquare= 0.033   (max possible= 0.997 )
## Likelihood ratio test= 3.44  on 1 df,   p=0.0637
## Wald test            = 3.7  on 1 df,   p=0.0543
## Score (logrank) test = 3.74  on 1 df,   p=0.0533
##
## 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]     9.83  18640.33     2.83 3.48  0.00051
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     18640   5.36e-05        73   4760882
##
## Concordance= 0.658  (se = 0.033 )
## Rsquare= 0.082   (max possible= 0.997 )
## Likelihood ratio test= 8.95  on 1 df,   p=0.00277
## Wald test            = 12.1  on 1 df,   p=0.000507
## Score (logrank) test = 13.1  on 1 df,   p=0.000298
```

4.3.2 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] -5.2585    0.0052   2.2693 -2.32     0.02
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]    0.0052        192  6.09e-05     0.445
##
## Concordance= 0.594  (se = 0.038 )
## Rsquare= 0.052   (max possible= 0.995 )
## Likelihood ratio test= 5.9  on 1 df,   p=0.0151
## Wald test            = 5.37  on 1 df,   p=0.0205
## Score (logrank) test = 5.47  on 1 df,   p=0.0194
##
## 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] -7.615649  0.000493  2.163756 -3.52  0.00043
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]  0.000493       2030  7.09e-06    0.0342
##
## Concordance= 0.641  (se = 0.038 )
## Rsquare= 0.125   (max possible= 0.995 )
## Likelihood ratio test= 14.7  on 1 df,   p=0.00013
## Wald test            = 12.4  on 1 df,   p=0.000432
## Score (logrank) test = 12.9  on 1 df,   p=0.000333
##
## 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] -1.93e+01  4.33e-09  6.10e+00 -3.16   0.0016
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]  4.33e-09   2.31e+08  2.81e-14  0.000669
##
## Concordance= 0.614  (se = 0.037 )
## Rsquare= 0.105   (max possible= 0.995 )
## Likelihood ratio test= 12.2  on 1 df,   p=0.000483
```

```
## Wald test            = 9.98  on 1 df,   p=0.00158
## Score (logrank) test = 10.4  on 1 df,   p=0.00128
##
## 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]     9.88  19496.05     1.87 5.29  1.2e-07
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     19496   5.13e-05       500    759461
##
## Concordance= 0.679  (se = 0.037 )
## Rsquare= 0.186   (max possible= 0.995 )
## Likelihood ratio test= 22.6  on 1 df,   p=1.99e-06
## Wald test            = 27.9  on 1 df,   p=1.25e-07
## Score (logrank) test = 30  on 1 df,   p=4.37e-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] -6.26163   0.00191  4.19477 -1.49     0.14
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]   0.00191        524  5.13e-07       7.1
##
## Concordance= 0.55  (se = 0.037 )
## Rsquare= 0.022   (max possible= 0.995 )
## Likelihood ratio test= 2.43  on 1 df,   p=0.119
## Wald test            = 2.23  on 1 df,   p=0.136
## Score (logrank) test = 2.23  on 1 df,   p=0.135
##
## 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]   6.08    435.71     2.35 2.59   0.0096
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]       436     0.0023      4.38     43330
##
## Concordance= 0.584  (se = 0.037 )
## Rsquare= 0.056   (max possible= 0.995 )
## Likelihood ratio test= 6.29  on 1 df,   p=0.0122
## Wald test            = 6.71  on 1 df,   p=0.00961
## Score (logrank) test = 6.8  on 1 df,   p=0.00913
##
```

```
## 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] -1.535     0.215    1.590 -0.97     0.33
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     0.215       4.64   0.00955      4.86
##
## Concordance= 0.576  (se = 0.037 )
## Rsquare= 0.009   (max possible= 0.995 )
## Likelihood ratio test= 0.98  on 1 df,   p=0.321
## Wald test            = 0.93  on 1 df,   p=0.334
## Score (logrank) test = 0.94  on 1 df,   p=0.333
##
## 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]   4.68    107.82     1.34 3.49  0.00049
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]       108    0.00927      7.77      1496
##
## Concordance= 0.612  (se = 0.037 )
## Rsquare= 0.086   (max possible= 0.995 )
## Likelihood ratio test= 9.85  on 1 df,   p=0.0017
## Wald test            = 12.2  on 1 df,   p=0.000487
## Score (logrank) test = 12.8  on 1 df,   p=0.00035
##
## 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]   5.16    174.23     1.79 2.88   0.0039
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]       174    0.00574      5.23      5807
##
## Concordance= 0.584  (se = 0.038 )
## Rsquare= 0.065   (max possible= 0.995 )
## Likelihood ratio test= 7.44  on 1 df,   p=0.00639
## Wald test            = 8.32  on 1 df,   p=0.00392
## Score (logrank) test = 8.48  on 1 df,   p=0.00358
##
## 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]    10.84  50933.98     2.92 3.71    2e-04
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     50934   1.96e-05       167  15552585
##
## Concordance= 0.657  (se = 0.035 )
## Rsquare= 0.09   (max possible= 0.995 )
## Likelihood ratio test= 10.4  on 1 df,   p=0.00128
## Wald test            = 13.8  on 1 df,   p=0.000205
## Score (logrank) test = 14.7  on 1 df,   p=0.000128
```

4.3.3 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] -4.65951   0.00947  2.59694 -1.79    0.073
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]   0.00947        106  5.83e-05      1.54
##
## Concordance= 0.587  (se = 0.041 )
## Rsquare= 0.041   (max possible= 0.997 )
## Likelihood ratio test= 3.42  on 1 df,   p=0.0645
## Wald test            = 3.22  on 1 df,   p=0.0728
## Score (logrank) test = 3.25  on 1 df,   p=0.0713
##
## 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] -6.47681   0.00154  2.30229 -2.81   0.0049
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]   0.00154        650  1.69e-05      0.14
##
## Concordance= 0.619  (se = 0.041 )
## Rsquare= 0.103   (max possible= 0.997 )
## Likelihood ratio test= 8.8  on 1 df,   p=0.003
## Wald test            = 7.91  on 1 df,   p=0.00491
```

```
## Score (logrank) test = 8.06  on 1 df,   p=0.00453
##
## 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] -1.59e+01  1.29e-07  7.08e+00 -2.24    0.025
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]  1.29e-07    7754803  1.21e-13     0.137
##
## Concordance= 0.602  (se = 0.041 )
## Rsquare= 0.07   (max possible= 0.997 )
## Likelihood ratio test= 5.89  on 1 df,   p=0.0152
## Wald test            = 5.02  on 1 df,   p=0.025
## Score (logrank) test = 5.06  on 1 df,   p=0.0245
##
## 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]   6.52    681.39     2.01 3.25   0.0012
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]       681    0.00147      13.3     34914
##
## Concordance= 0.646  (se = 0.041 )
## Rsquare= 0.106   (max possible= 0.997 )
## Likelihood ratio test= 9.07  on 1 df,   p=0.00259
## Wald test            = 10.6  on 1 df,   p=0.00116
## Score (logrank) test = 10.9  on 1 df,   p=0.000954
##
## 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] -1.97      0.14     4.48 -0.44     0.66
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      0.14       7.14  2.14e-05       918
##
## Concordance= 0.53  (se = 0.041 )
## Rsquare= 0.002   (max possible= 0.997 )
## Likelihood ratio test= 0.2  on 1 df,   p=0.656
## Wald test            = 0.19  on 1 df,   p=0.661
## Score (logrank) test = 0.19  on 1 df,   p=0.661
##
```

```
## 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]  2.98     19.78     2.47 1.21     0.23
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      19.8     0.0506     0.155      2528
##
## Concordance= 0.536  (se = 0.041 )
## Rsquare= 0.017   (max possible= 0.997 )
## Likelihood ratio test= 1.42  on 1 df,   p=0.233
## Wald test            = 1.45  on 1 df,   p=0.228
## Score (logrank) test = 1.46  on 1 df,   p=0.227
##
## 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] 0.467     1.595    1.676 0.28     0.78
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      1.59      0.627    0.0597      42.6
##
## Concordance= 0.479  (se = 0.04 )
## Rsquare= 0.001   (max possible= 0.997 )
## Likelihood ratio test= 0.08  on 1 df,   p=0.782
## Wald test            = 0.08  on 1 df,   p=0.781
## Score (logrank) test = 0.08  on 1 df,   p=0.781
##
## 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]  2.93     18.65     1.57 1.86    0.063
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      18.7     0.0536     0.857       406
##
## Concordance= 0.565  (se = 0.041 )
## Rsquare= 0.038   (max possible= 0.997 )
## Likelihood ratio test= 3.1  on 1 df,   p=0.0783
## Wald test            = 3.47  on 1 df,   p=0.0626
## Score (logrank) test = 3.52  on 1 df,   p=0.0608
##
## 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]  4.05     57.37     1.94 2.09    0.037
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      57.4     0.0174      1.29      2558
##
## Concordance= 0.572  (se = 0.041 )
## Rsquare= 0.048   (max possible= 0.997 )
## Likelihood ratio test= 3.97  on 1 df,   p=0.0462
## Wald test            = 4.37  on 1 df,   p=0.0366
## Score (logrank) test = 4.42  on 1 df,   p=0.0356
##
## 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]  4.49     89.25     3.75 1.2     0.23
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      89.3     0.0112    0.0576    138229
##
## Concordance= 0.575  (se = 0.04 )
## Rsquare= 0.016   (max possible= 0.997 )
## Likelihood ratio test= 1.31  on 1 df,   p=0.253
## Wald test            = 1.44  on 1 df,   p=0.231
## Score (logrank) test = 1.44  on 1 df,   p=0.23
```

4.3.4 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 = -7.138, p-value = 9.463e-13
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.4033
##
##
## $mg.2
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
```

```
## z = 0.5827, p-value = 0.5601
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.0328
##
##
## $mg.3
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 4.367, p-value = 1.259e-05
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##   tau
## 0.247
##
##
## $mg.4
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -1.106, p-value = 0.2689
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.06235
##
##
## $mg.5
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 1.834, p-value = 0.06659
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.1044
##
##
## $mg.6
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.495, p-value = 0.0004747
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.1994
##
```

```
##
## $mg.7
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -4.004, p-value = 6.228e-05
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.2346
##
##
## $mg.8
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -0.8174, p-value = 0.4137
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.04674
##
##
## $mg.9
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.366, p-value = 0.0007635
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.1893
##
##
## $mg.10
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 0.3805, p-value = 0.7036
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## 0.02252
```

4.4 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
```

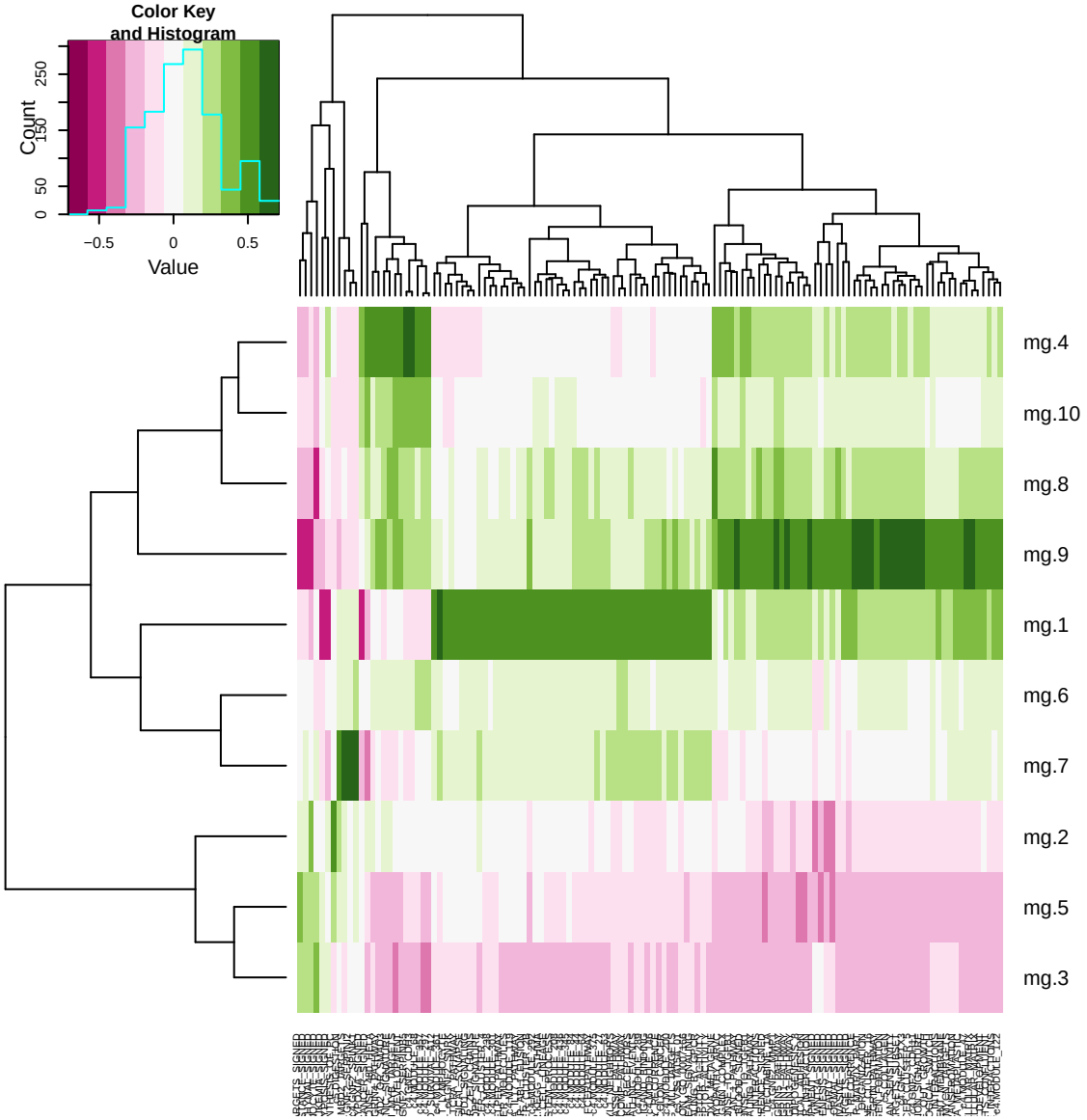
```
##          pure.p pure.p.BY pure.p.Holm  pure.s surv.diag_dsd.c
## mg.1  9.463e-13 1.620e-10   3.785e-11 -7.1381          -5.259
## mg.2  5.601e-01 1.000e+00   1.000e+00  0.5827          -7.616
## mg.3  1.259e-05 7.183e-04   4.784e-04  4.3671         -19.257
## mg.4  2.689e-01 1.000e+00   1.000e+00 -1.1056           9.878
## mg.5  6.659e-02 4.221e-01   1.000e+00  1.8344          -6.262
## mg.6  4.747e-04 1.034e-02   1.614e-02 -3.4947           6.077
## mg.7  6.228e-05 2.665e-03   2.304e-03 -4.0040          -1.535
## mg.8  4.137e-01 1.000e+00   1.000e+00 -0.8174           4.680
## mg.9  7.635e-04 1.452e-02   2.443e-02 -3.3657           5.160
## mg.10 7.036e-01 1.000e+00   1.000e+00  0.3805          10.838
##       surv.diag_dsd.p surv.diag_dsd.p.BY surv.diag_dsd.p.Holm
## mg.1        1.510e-02           0.130060            3.322e-01
## mg.2        1.296e-04           0.004437            4.667e-03
## mg.3        4.832e-04           0.010337            1.614e-02
## mg.4        1.987e-06           0.000170            7.749e-05
## mg.5        1.194e-01           0.659186            1.000e+00
## mg.6        1.216e-02           0.115629            2.797e-01
## mg.7        3.213e-01           1.000000            1.000e+00
## mg.8        1.697e-03           0.026399            5.090e-02
## mg.9        6.393e-03           0.064357            1.534e-01
## mg.10       1.280e-03           0.021910            3.969e-02
##       surv.diag_rec.c surv.diag_rec.p surv.diag_rec.p.BY
## mg.1           -3.557       0.0637444           0.422075
## mg.2           -3.542       0.0379596           0.295294
## mg.3          -10.105       0.0323402           0.263560
## mg.4            7.228       0.0003047           0.008692
## mg.5           -6.869       0.0757802           0.462191
## mg.6            6.736       0.0047268           0.050560
## mg.7           -2.362       0.1138622           0.649552
## mg.8            4.261       0.0025314           0.033878
## mg.9            3.386       0.0637490           0.422075
## mg.10           9.833       0.0027713           0.033878
##       surv.diag_rec.p.Holm surv.recr_dsd.c surv.recr_dsd.p
## mg.1               1.00000         -4.6595        0.064534
```

	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	0.0000	-7.1381	-5.258	0.3322	-3.557	1.0000	-4.6595	1.0000
mg.2	1.0000	0.5827	-7.616	0.0047	-3.542	0.7212	-6.4768	0.0781
mg.3	0.0005	4.3671	-19.257	0.0161	-10.105	0.6468	-15.8638	0.3322
mg.4	1.0000	-1.1056	9.878	0.0001	7.228	0.0107	6.5241	0.0734
mg.5	1.0000	1.8344	-6.262	1.0000	-6.869	1.0000	-1.9654	1.0000
mg.6	0.0161	-3.4947	6.077	0.2797	6.736	0.1182	2.9845	1.0000
mg.7	0.0023	-4.0040	-1.535	1.0000	-2.362	1.0000	0.4668	1.0000
mg.8	1.0000	-0.8174	4.681	0.0509	4.261	0.0734	2.9260	1.0000
mg.9	0.0244	-3.3657	5.160	0.1534	3.386	1.0000	4.0496	0.8320
mg.10	1.0000	0.3805	10.838	0.0397	9.833	0.0748	4.4915	1.0000

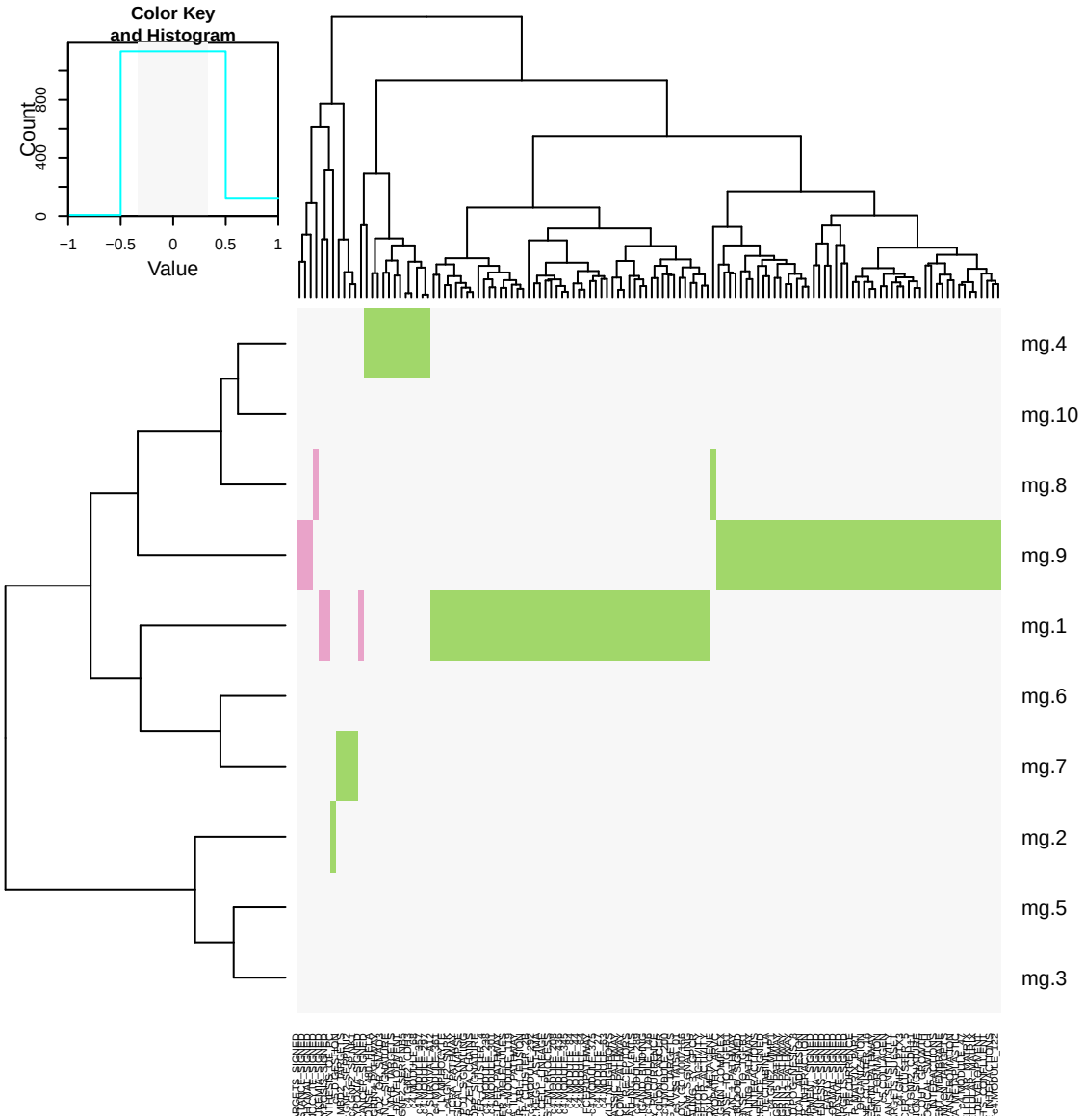
```
## mg.2                0.72123          -6.4768         0.003005
## mg.3                0.64680         -15.8638         0.015199
## mg.4                0.01067           6.5241         0.002594
## mg.5                1.00000          -1.9654         0.655637
## mg.6                0.11817           2.9845         0.233436
## mg.7                1.00000           0.4668         0.782394
## mg.8                0.07341           2.9260         0.078318
## mg.9                1.00000           4.0496         0.046220
## mg.10               0.07483           4.4915         0.253255
##       surv.recr_dsd.p.BY surv.recr_dsd.p.Holm
## mg.1             0.42207               1.00000
## mg.2             0.03428               0.07812
## mg.3             0.13006               0.33223
## mg.4             0.03388               0.07341
## mg.5             1.00000               1.00000
## mg.6             1.00000               1.00000
## mg.7             1.00000               1.00000
## mg.8             0.46219               1.00000
## mg.9             0.34392               0.83197
## mg.10            1.00000               1.00000
```

4.5 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))
```




```
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
```

4.5.1 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: 600.81095574187
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.4 + mg.6 + mg.8"
## Evidence weight: 0.274925723557039
## Worst IC: 612.196719247104
## 2 models within 2 IC units.
## 64 models to reach 95% of evidence weight.
```

```
coef(diag_rec.asreg.result)
```

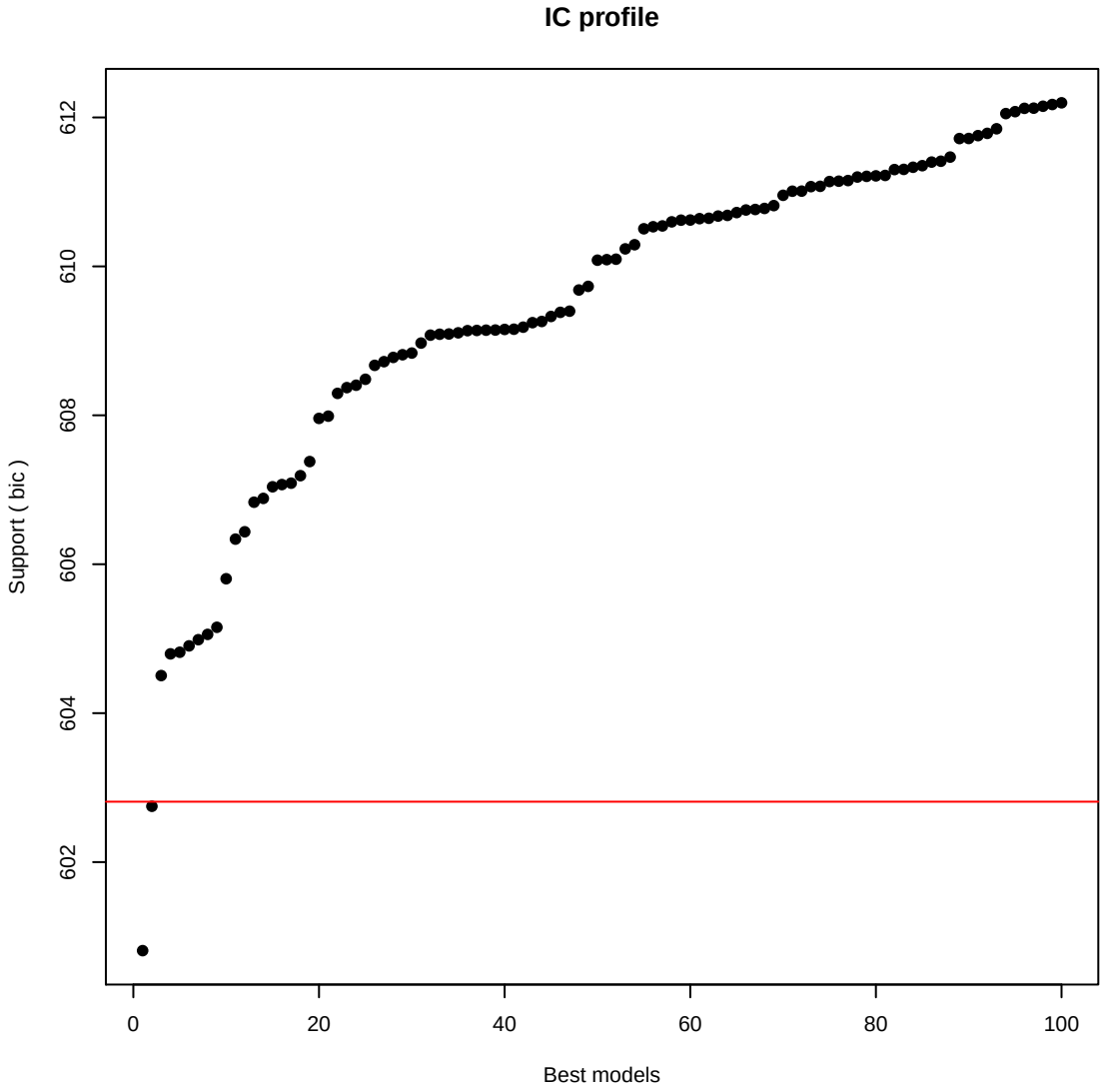
```
##       Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.9  -0.02239          0.03995        17    0.08675           0.3965
## mg.5  -0.01487          0.16224        18    0.08745           0.7990
## mg.2   0.08803          0.09721        20    0.09777           0.6185
## mg.7  -0.09325          0.07285        20    0.10078           0.5355
## mg.1  -0.14424          0.15933        23    0.10736           0.7919
## mg.10 -0.01818          0.43686        26    0.11493           1.3112
## mg.3  -0.74240          2.66317        30    0.14288           3.2374
## mg.6   4.30787         10.54936        57    0.72295           6.4434
## mg.8   4.11619          2.80175        77    0.91557           3.3206
## mg.4   6.45579          4.98099        83    0.95575           4.4275
```

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

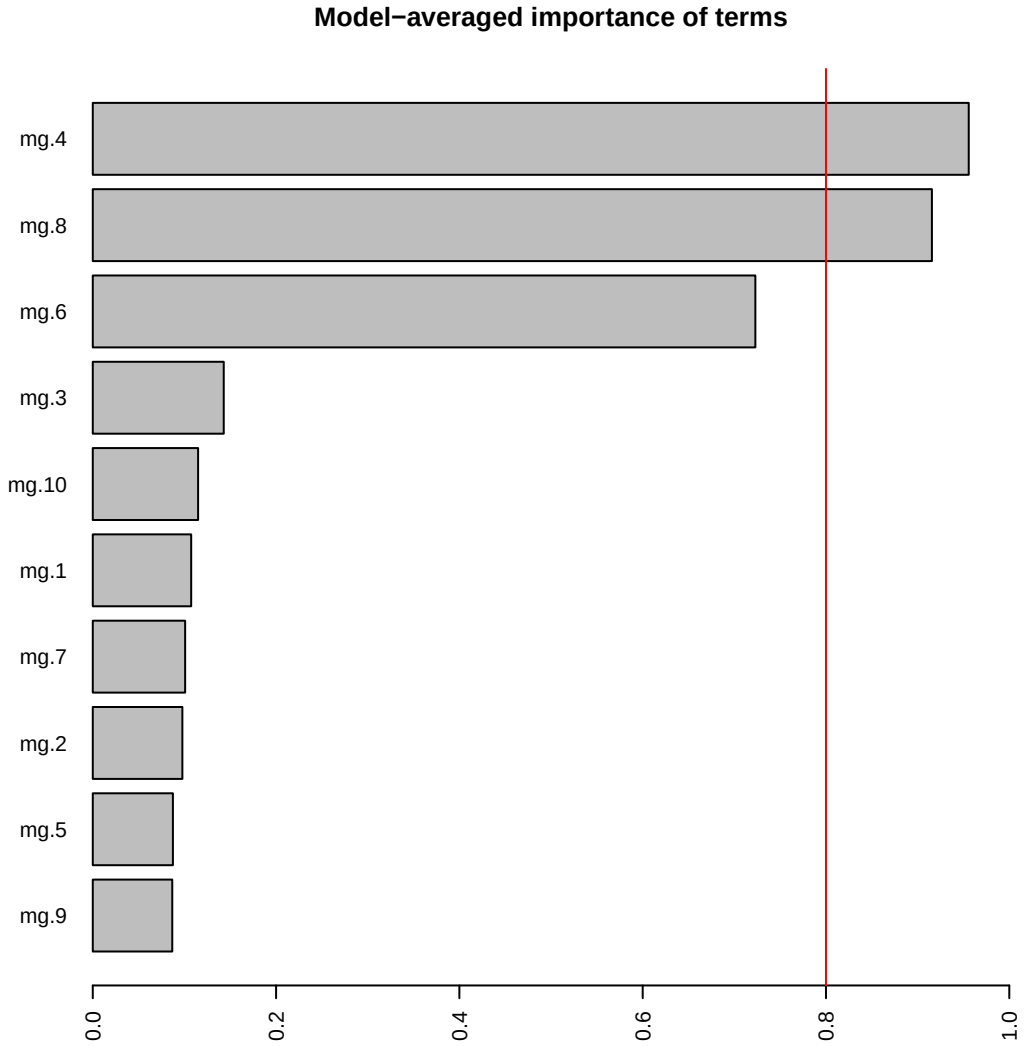
```
## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 104, number of events= 77
##
##        coef exp(coef) se(coef)    z Pr(>|z|)
## mg.4   6.54    694.04     1.86 3.52  0.00042
## mg.6   5.86    350.43     2.26 2.60  0.00942
## mg.8   4.60     99.57     1.36 3.38  0.00072
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.4     694.0    0.00144     18.25     26388
## mg.6     350.4    0.00285      4.20     29210
## mg.8      99.6    0.01004      6.92      1432
##
## Concordance= 0.699  (se = 0.036 )
## Rsquare= 0.238   (max possible= 0.997 )
## Likelihood ratio test= 28.3  on 3 df,   p=3.15e-06
## Wald test            = 32  on 3 df,   p=5.16e-07
## Score (logrank) test = 34.5  on 3 df,   p=1.54e-07
```

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

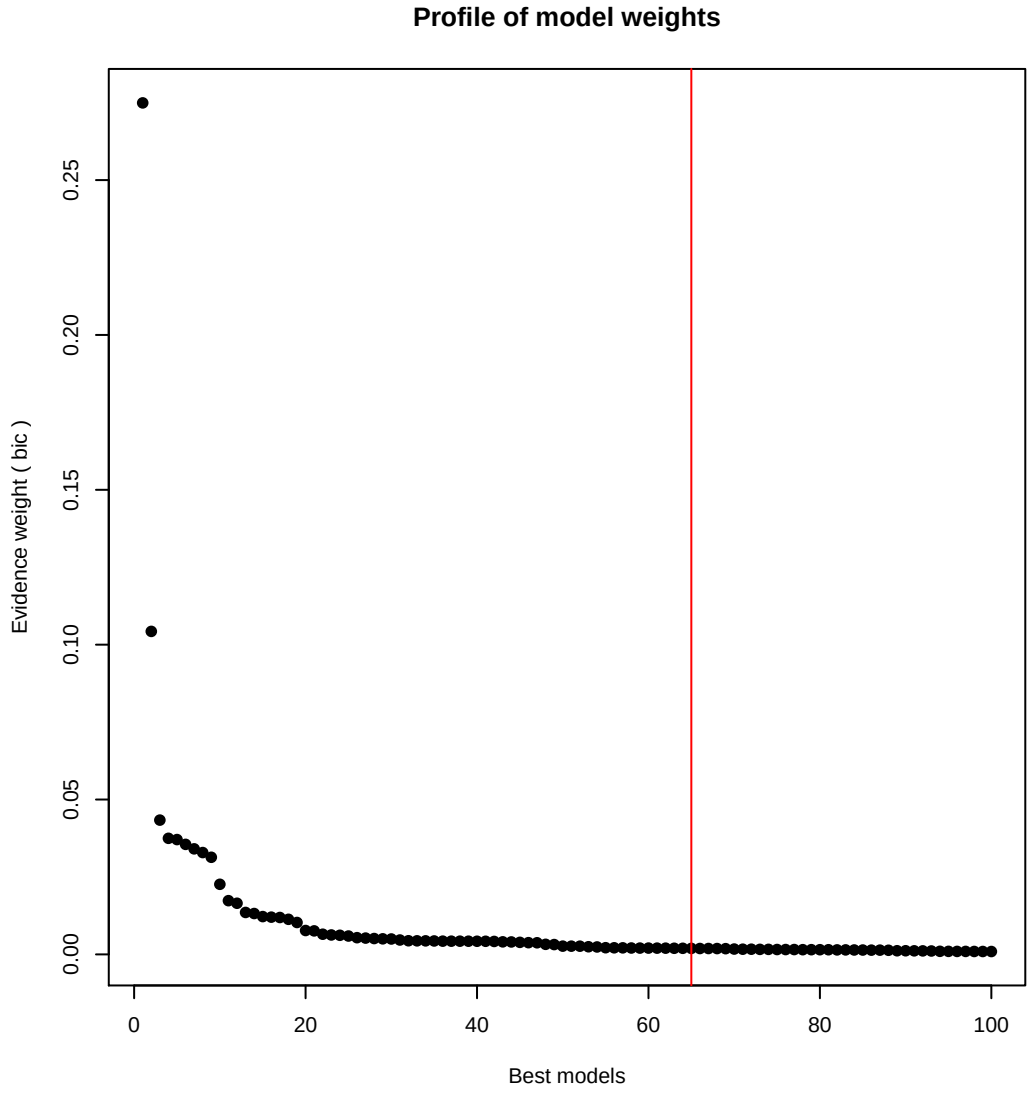
All-subsets regression




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




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




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

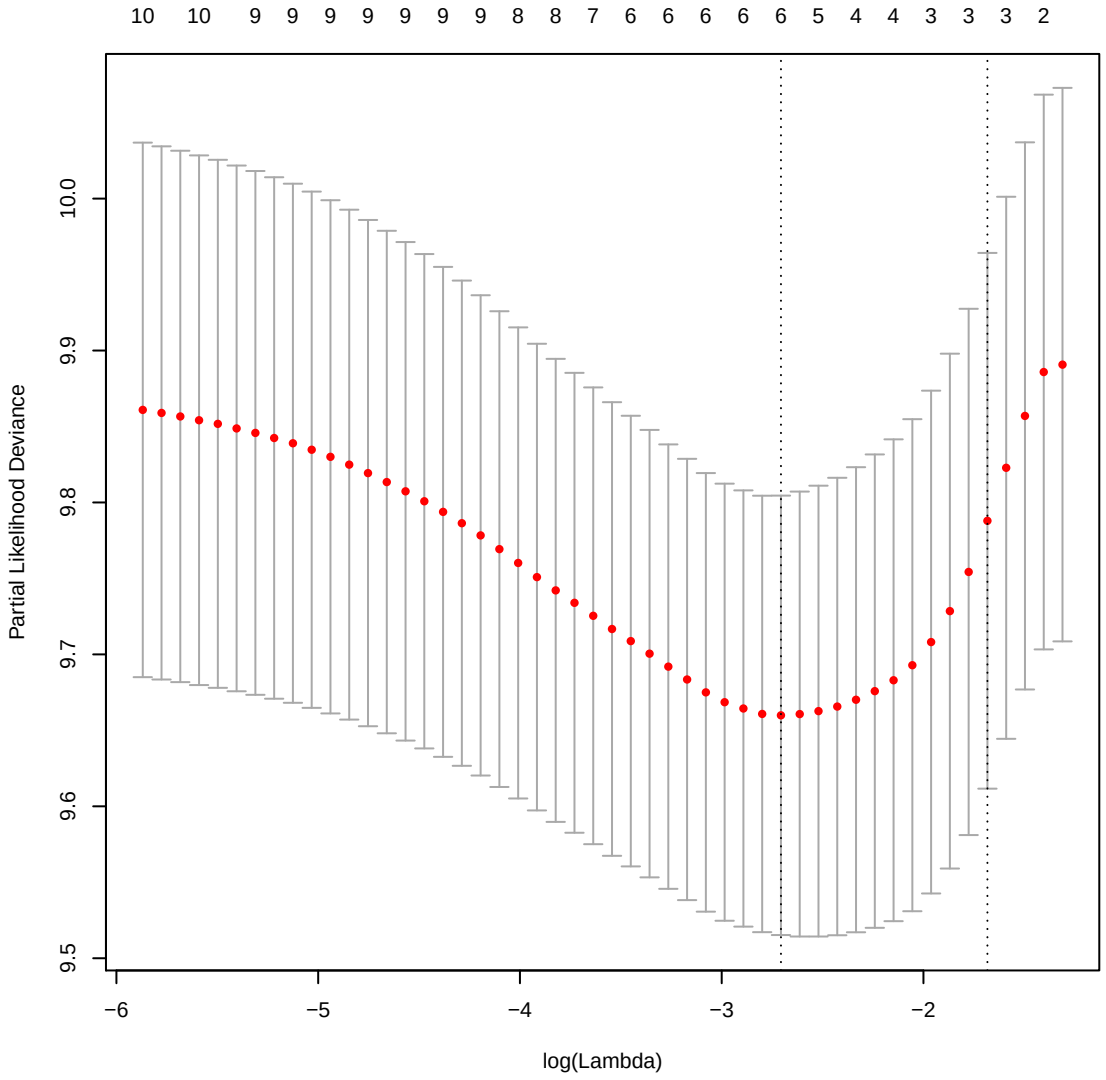
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##           1
## mg.1  .
## mg.2  .
## mg.3  .
## mg.4  2.580
## mg.5  .
## mg.6  1.008
## mg.7  .
## mg.8  1.353
## mg.9  .
## mg.10 .
```

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

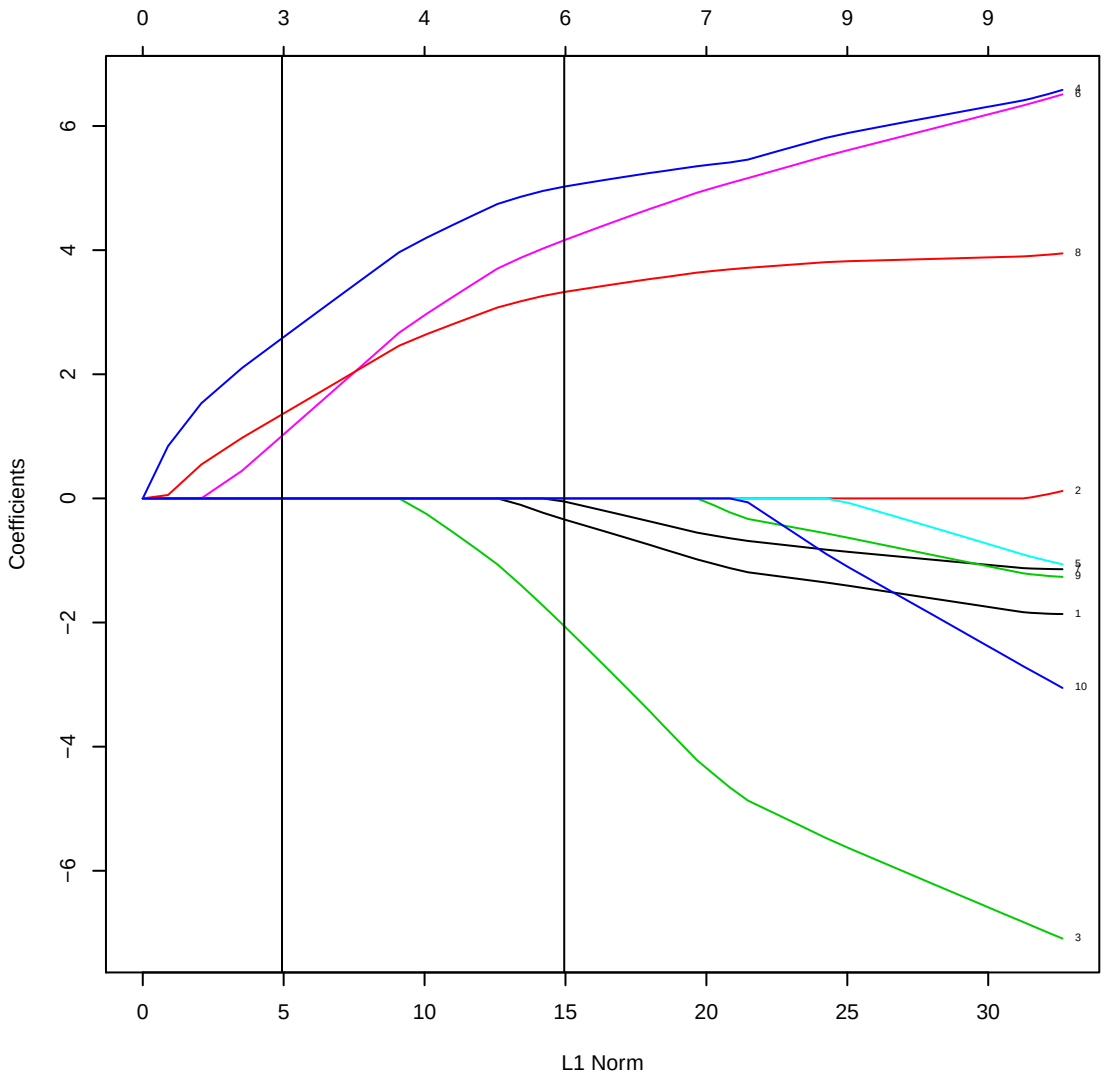
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##              1
## mg.1  -0.33945
## mg.2   .
## mg.3  -2.05651
## mg.4   5.02355
## mg.5   .
## mg.6   4.15986
## mg.7  -0.05016
## mg.8   3.32575
## mg.9   .
## mg.10  .
```

```
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)))
```



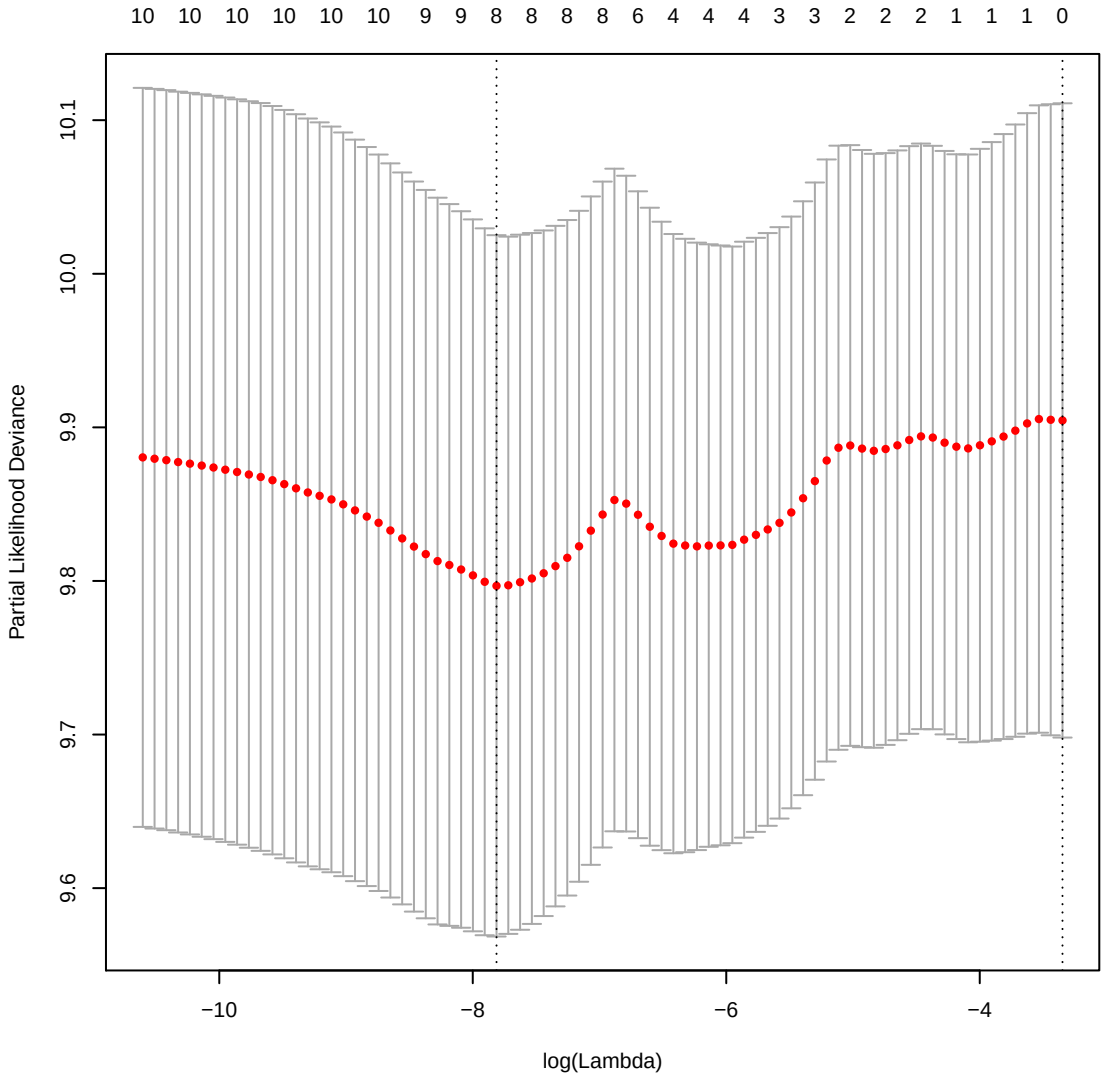

```
diag_rec.adaglmnet.coef.1se/diag_rec.adaglmnet.weights

## 10 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  .
## mg.10 .
```

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

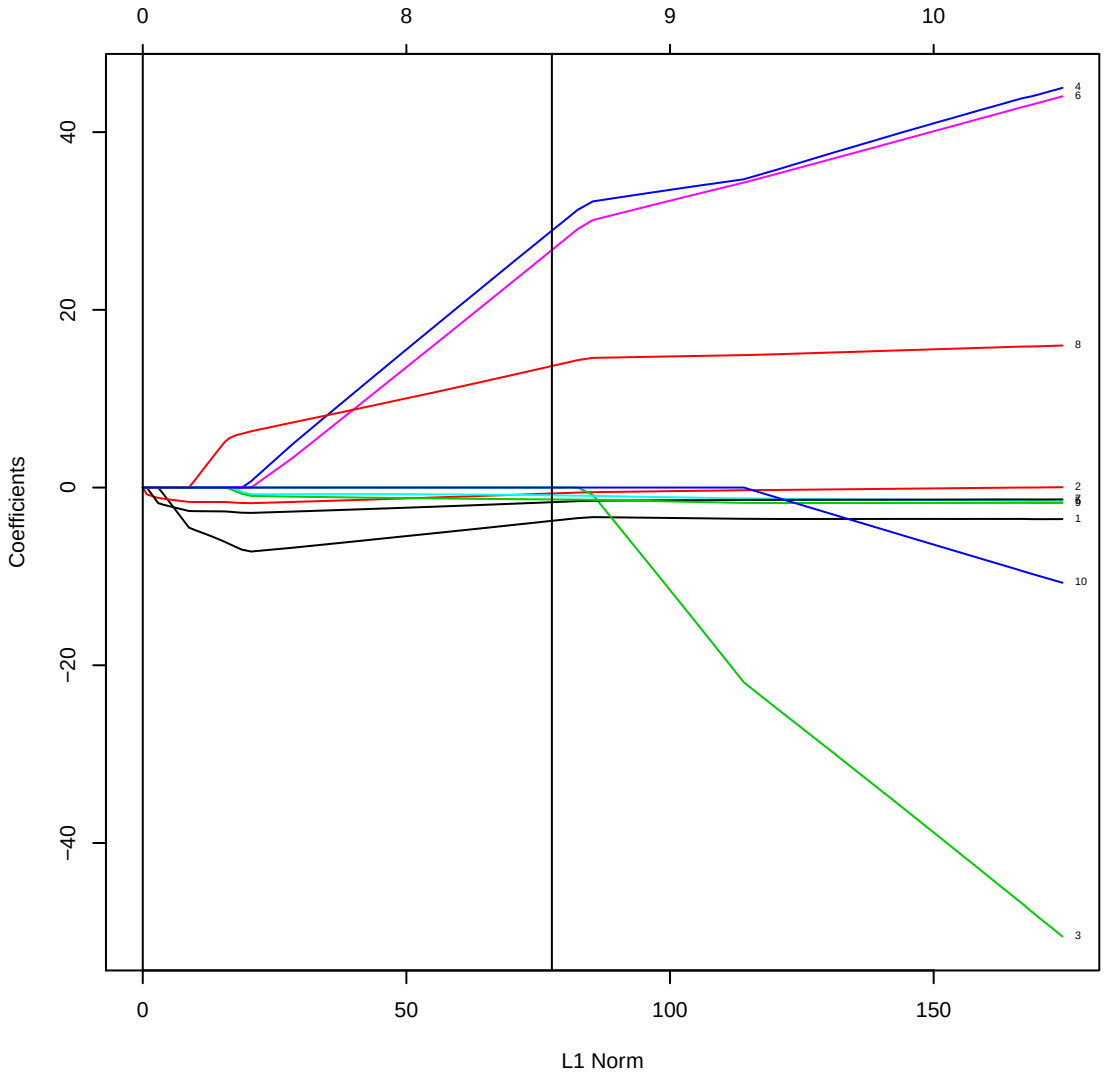
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1   -7.009
## mg.2   -0.230
## mg.3    .
## mg.4  197.871
## mg.5   -1.102
## mg.6  180.918
## mg.7   -1.877
## mg.8   55.114
## mg.9   -1.725
## mg.10   .
```

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

Adaptive LASSO




```
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)))
```



4.5.2 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: 552.465400870989
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.3 + mg.4 + mg.8"
## Evidence weight: 0.0866221510107644
## Worst IC: 560.1274663063
## 6 models within 2 IC units.
## 77 models to reach 95% of evidence weight.
```

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

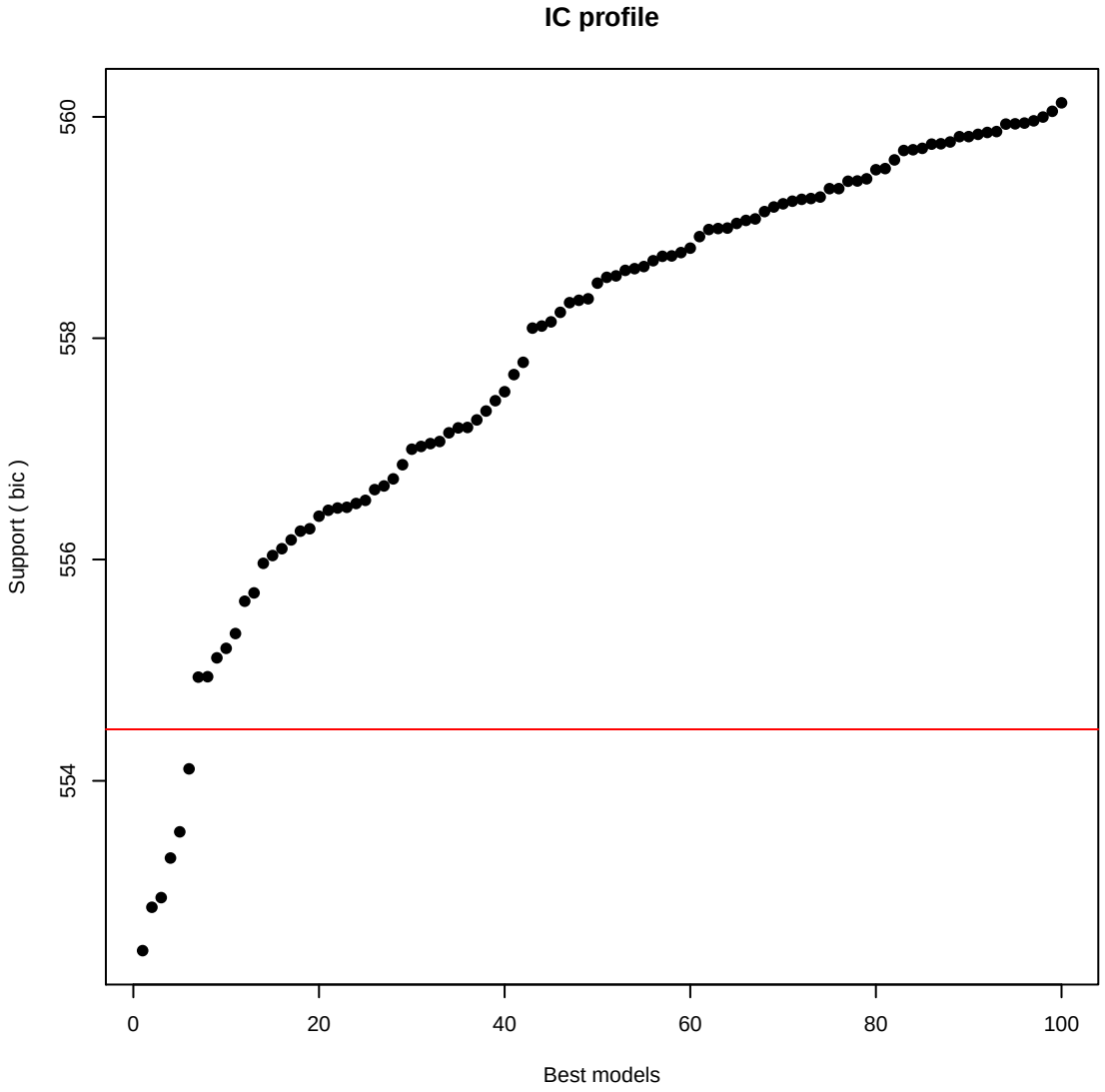
```
##       Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.7   0.02323          0.02722        15    0.07466           0.3271
## mg.9   0.04018          0.05541        18    0.08239           0.4667
## mg.5   0.19730          0.40254        20    0.09114           1.2578
## mg.10 -0.53773          1.43943        21    0.12133           2.3786
## mg.2  -0.86787          2.49543        37    0.21793           3.1318
## mg.1  -1.85822          7.51145        46    0.36031           5.4335
## mg.6   1.81195          6.59272        42    0.38631           5.0904
## mg.3  -9.95225         78.70608        68    0.66636          17.5884
## mg.8   3.34411          5.12754        62    0.75772           4.4893
## mg.4   8.49231          6.95242        93    0.96790           5.2274
```

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

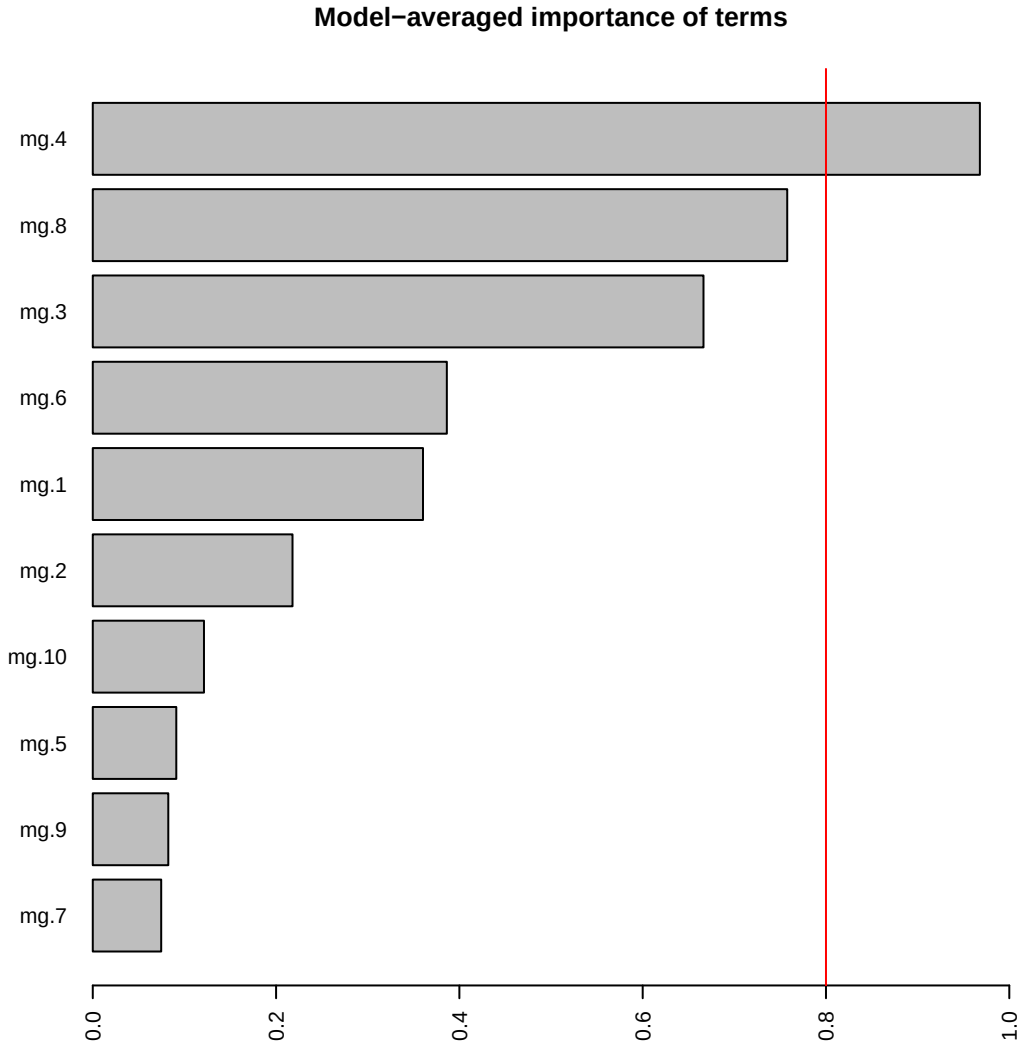
```
## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 110, number of events= 70
##
##           coef exp(coef)  se(coef)     z Pr(>|z|)
## mg.3 -1.26e+01  3.43e-06  6.18e+00 -2.03   0.0419
## mg.4  9.35e+00  1.15e+04  1.95e+00  4.79  1.7e-06
## mg.8  4.29e+00  7.27e+01  1.48e+00  2.90   0.0037
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.3  3.43e-06   2.92e+05  1.87e-11  6.29e-01
## mg.4  1.15e+04   8.68e-05  2.51e+02  5.29e+05
## mg.8  7.27e+01   1.38e-02  4.02e+00  1.32e+03
##
## Concordance= 0.713  (se = 0.038 )
## Rsquare= 0.287   (max possible= 0.995 )
## Likelihood ratio test= 37.3  on 3 df,   p=4.04e-08
## Wald test            = 39.4  on 3 df,   p=1.45e-08
## Score (logrank) test = 42.9  on 3 df,   p=2.54e-09
```

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

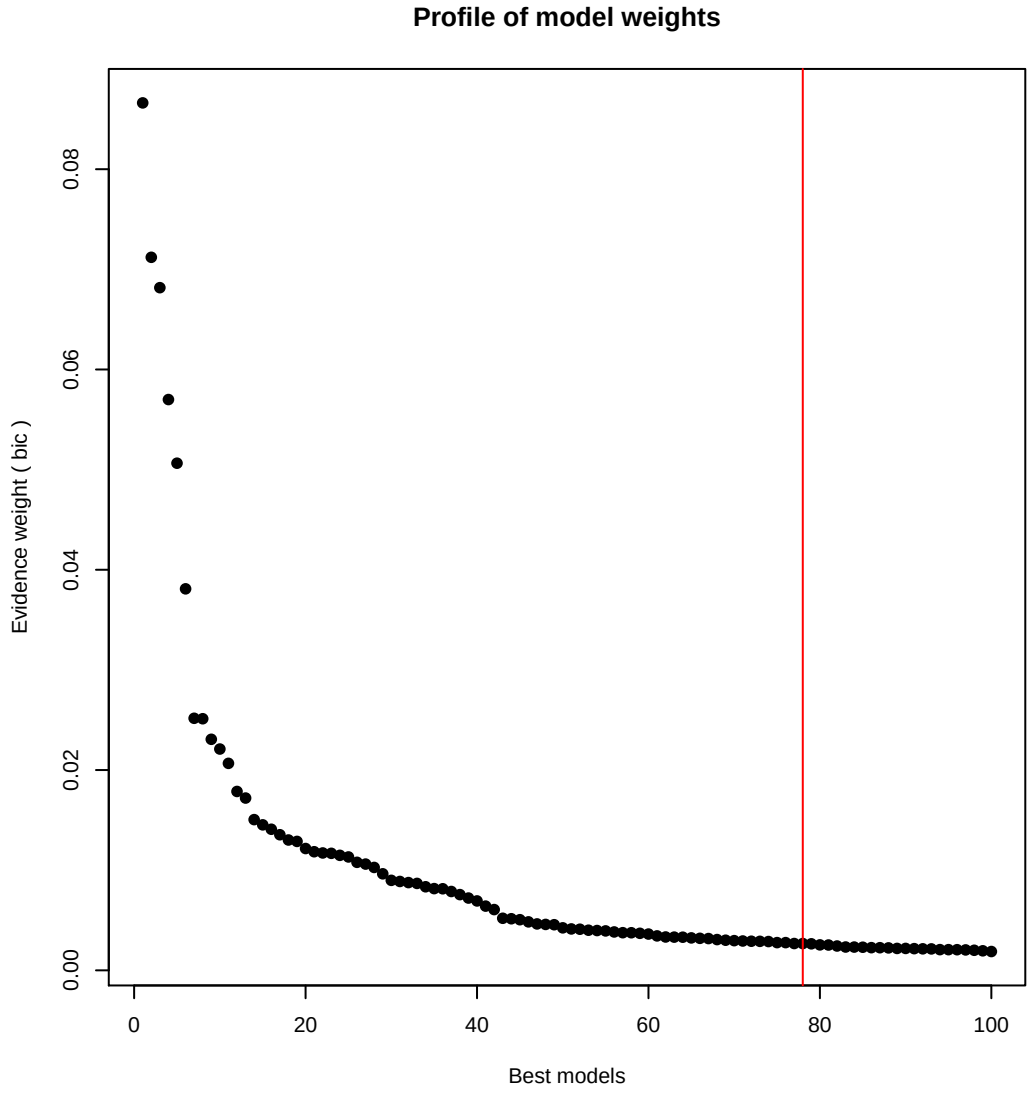
All-subsets regression




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




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




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

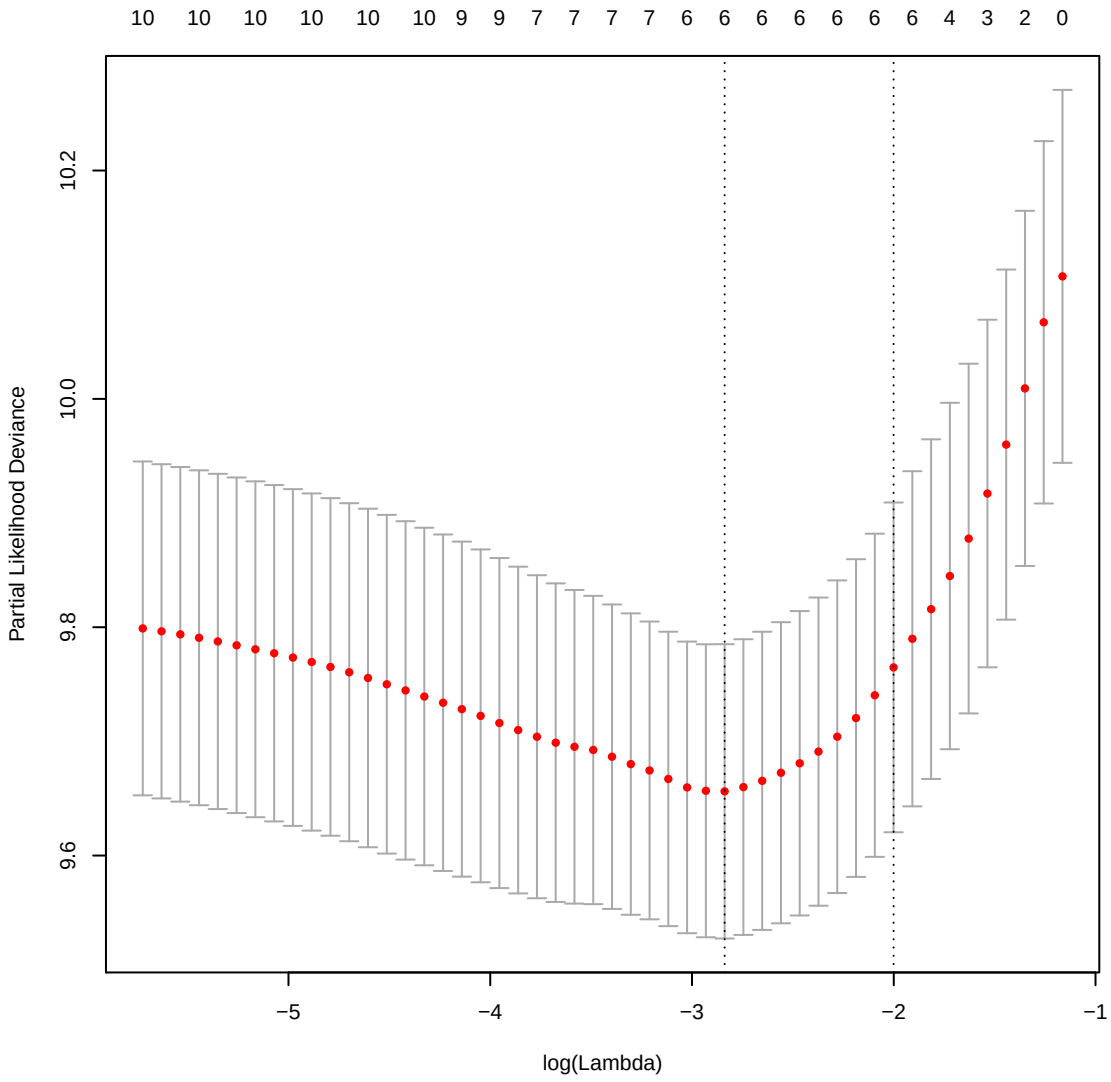
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  -0.5540
## mg.2  -0.4526
## mg.3  -5.1821
## mg.4   5.5790
## mg.5   .
## mg.6   0.7854
## mg.7   .
## mg.8   1.8326
## mg.9   .
## mg.10  .
```

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

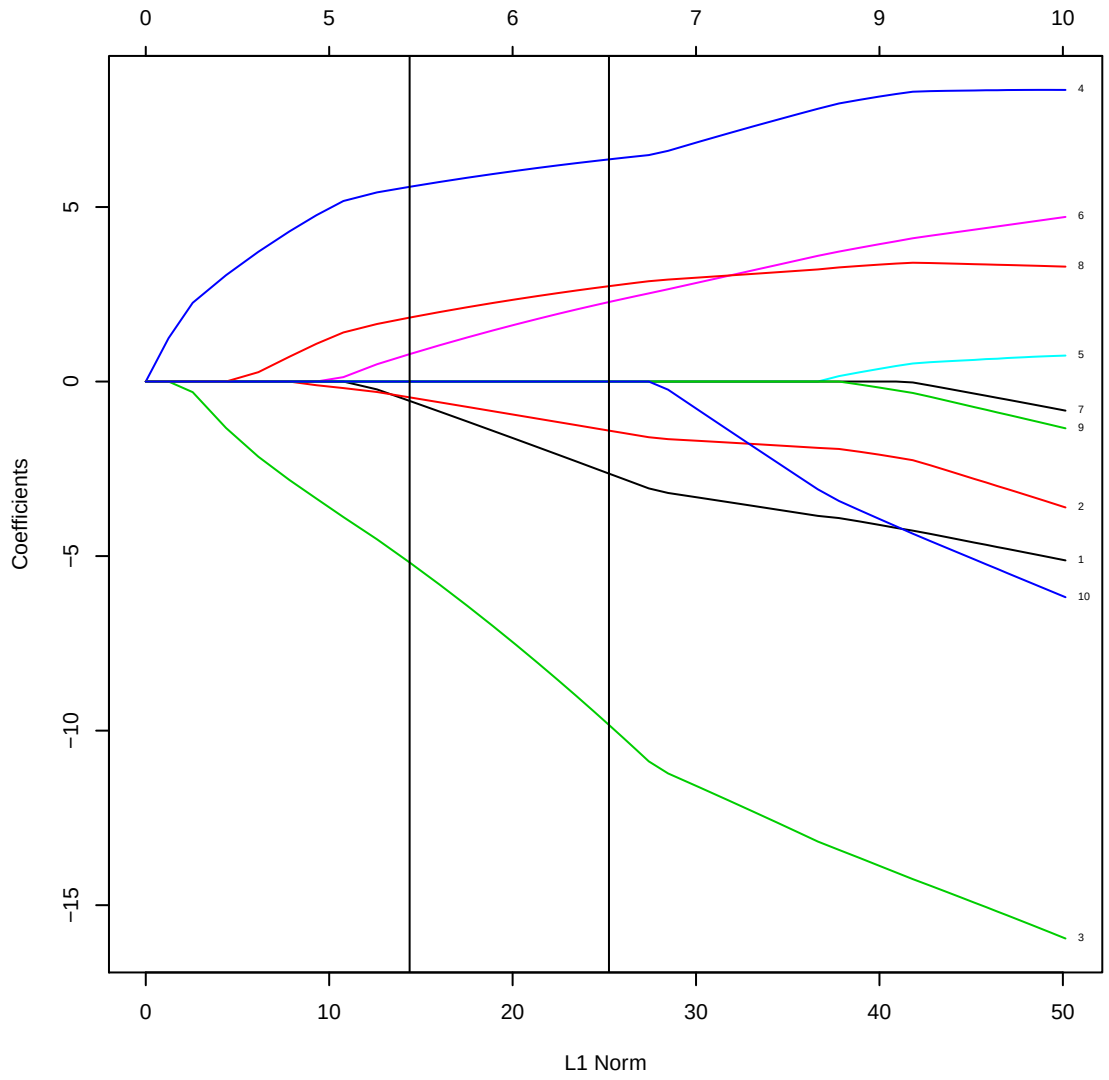
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##            1
## mg.1  -2.637
## mg.2  -1.405
## mg.3  -9.836
## mg.4   6.365
## mg.5   .
## mg.6   2.278
## mg.7   .
## mg.8   2.733
## mg.9   .
## mg.10  .
```

```
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)))
```



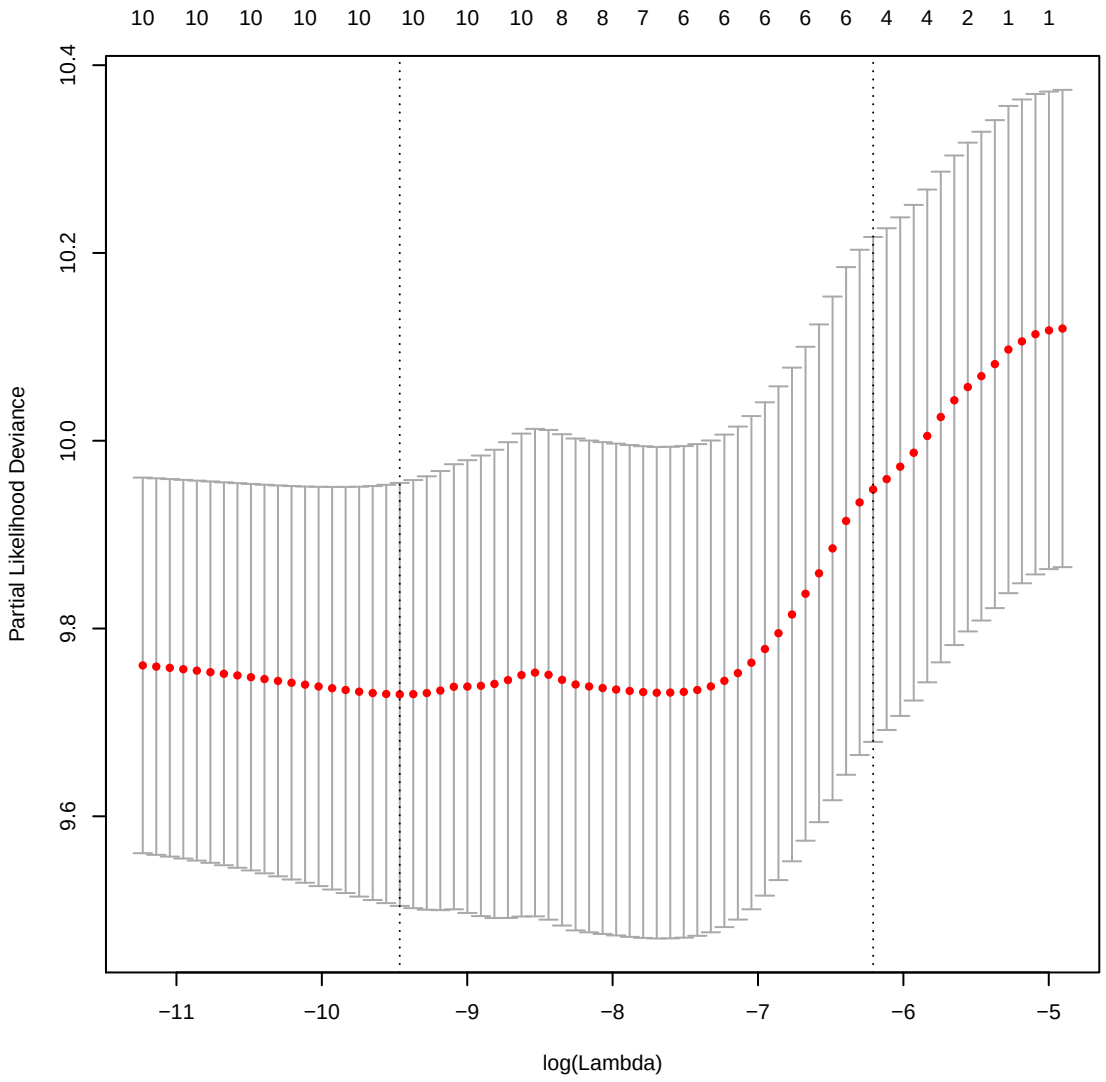

```
diag_dsd.adaglmnet.coef.1se/diag_dsd.adaglmnet.weights

## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1    .
## mg.2  -51.698
## mg.3    .
## mg.4    .
## mg.5    .
## mg.6    .
## mg.7   -1.333
## mg.8   21.975
## mg.9    4.222
## mg.10   .
```

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

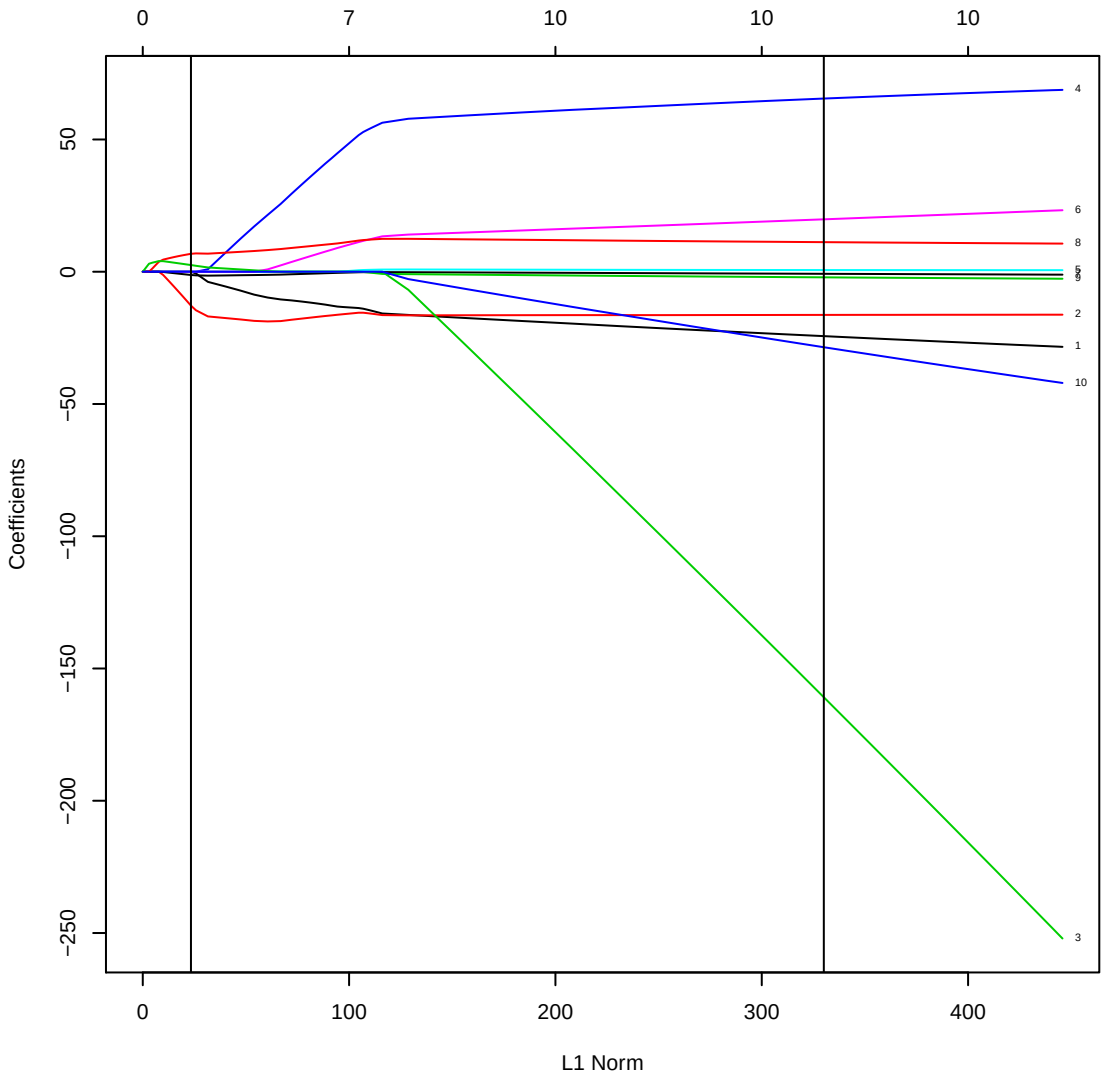
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                1
## mg.1   -132.1598
## mg.2    -65.8289
## mg.3  -2654.4258
## mg.4    544.3288
## mg.5      0.4865
## mg.6     97.1846
## mg.7     -0.8463
## mg.8     36.2655
## mg.9     -3.5305
## mg.10  -190.7520
```

```
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)))
```



4.5.3 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: 452.406312804879
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.1 + mg.2"
## Evidence weight: 0.0707778648644386
## Worst IC: 458.842886678192
## 9 models within 2 IC units.
## 84 models to reach 95% of evidence weight.
```

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

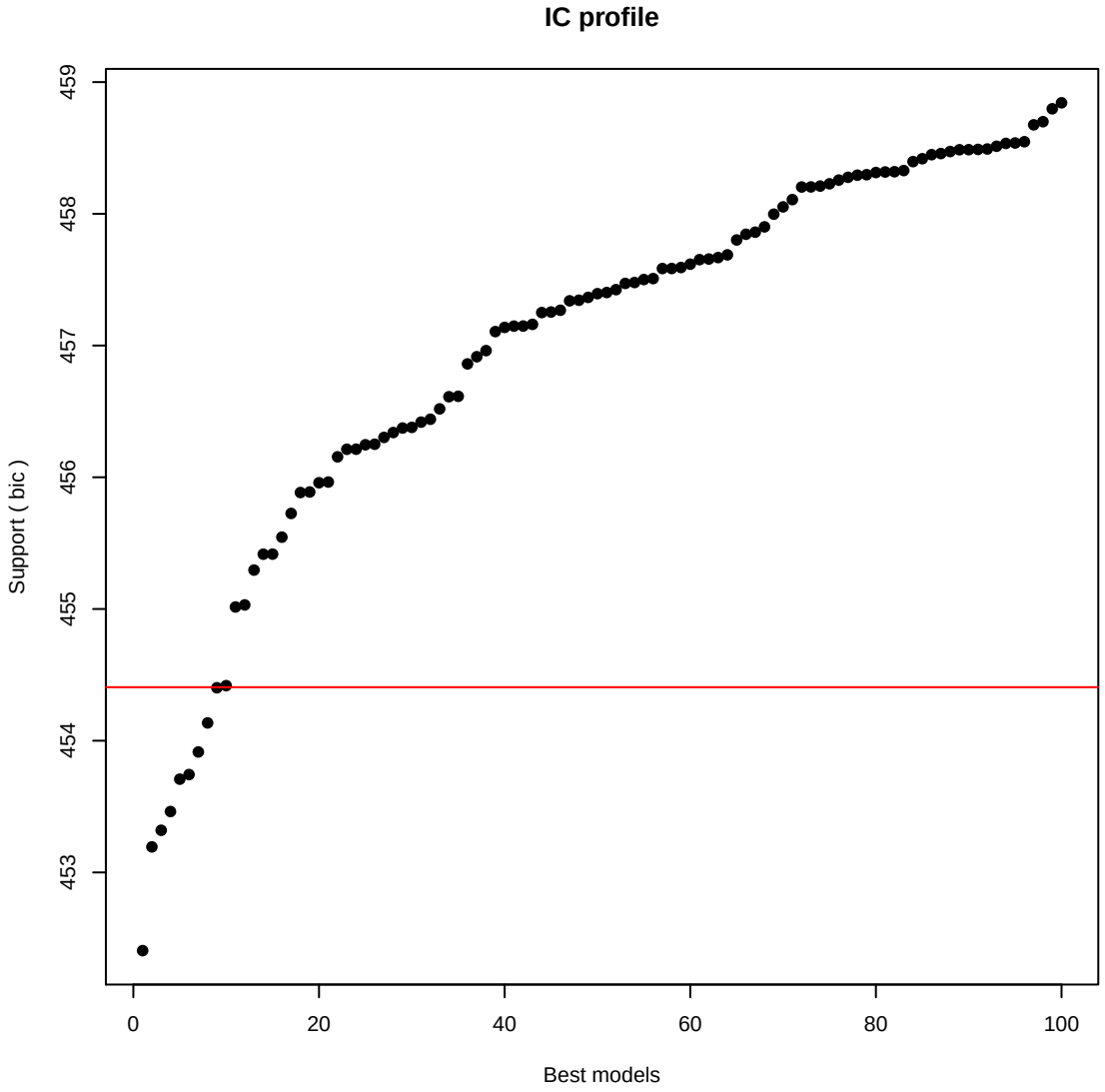
```
##       Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.5  -0.02553          0.06789        10    0.05309           0.5188
## mg.9   0.11944          0.11911        15    0.08092           0.6871
## mg.6   0.12963          0.16476        16    0.08278           0.8081
## mg.7   0.25576          0.32091        19    0.11864           1.1278
## mg.8   0.57051          1.08970        28    0.20356           2.0783
## mg.10 -1.69567          9.54010        26    0.20759           6.1493
## mg.3  -3.85670         38.20229        38    0.31091          12.3053
## mg.1  -2.79647         11.81787        46    0.48219           6.8441
## mg.2  -3.34977         13.06581        49    0.54994           7.1964
## mg.4   4.05292         11.53270        72    0.67898           6.7610
```

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

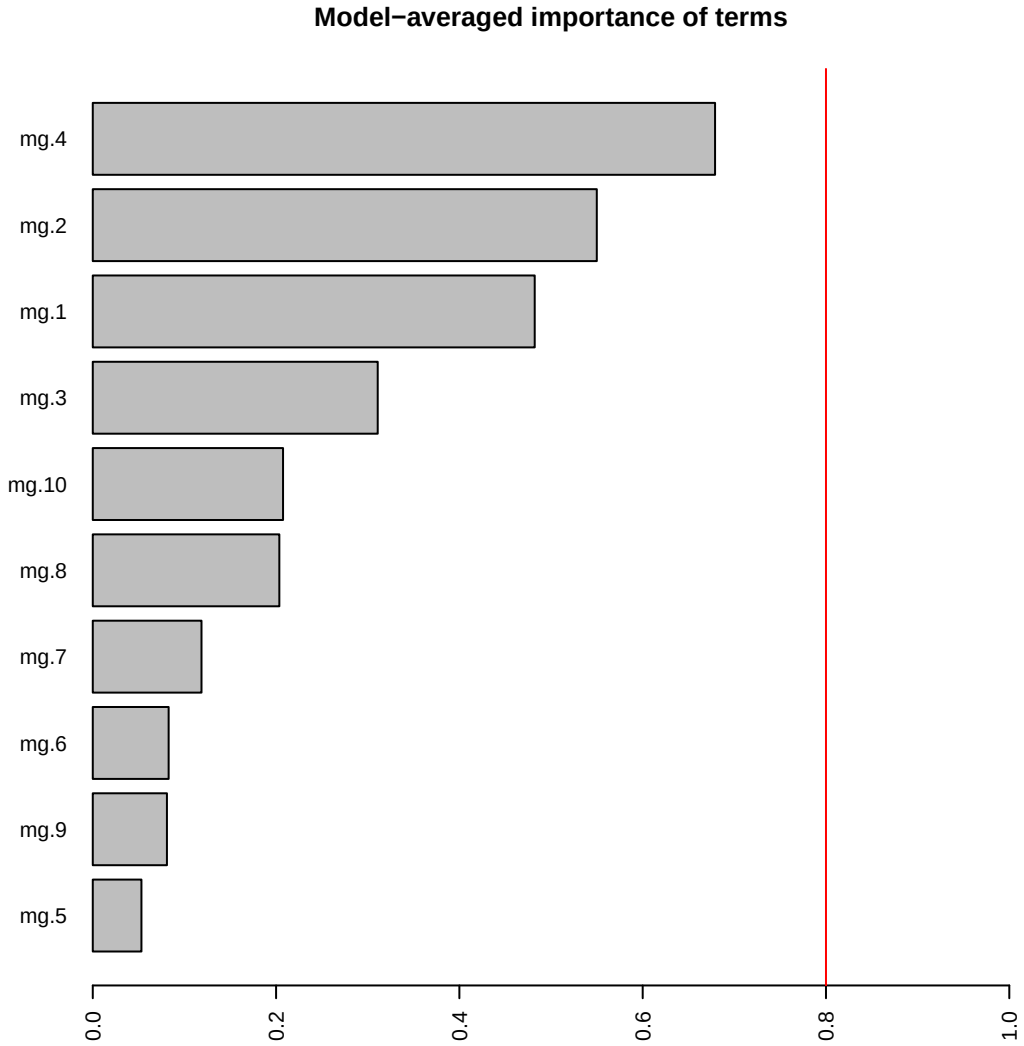
```
## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 81, number of events= 64
##
##          coef exp(coef) se(coef)     z Pr(>|z|)
## mg.1 -5.58702   0.00375  2.52667 -2.21    0.027
## mg.2 -6.80090   0.00111  2.19640 -3.10    0.002
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.1   0.00375        267  2.65e-05    0.5300
## mg.2   0.00111        899  1.50e-05    0.0824
##
## Concordance= 0.646  (se = 0.041 )
## Rsquare= 0.159   (max possible= 0.997 )
## Likelihood ratio test= 14  on 2 df,   p=0.000903
## Wald test            = 14  on 2 df,   p=0.000911
## Score (logrank) test = 14.4  on 2 df,   p=0.000746
```

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

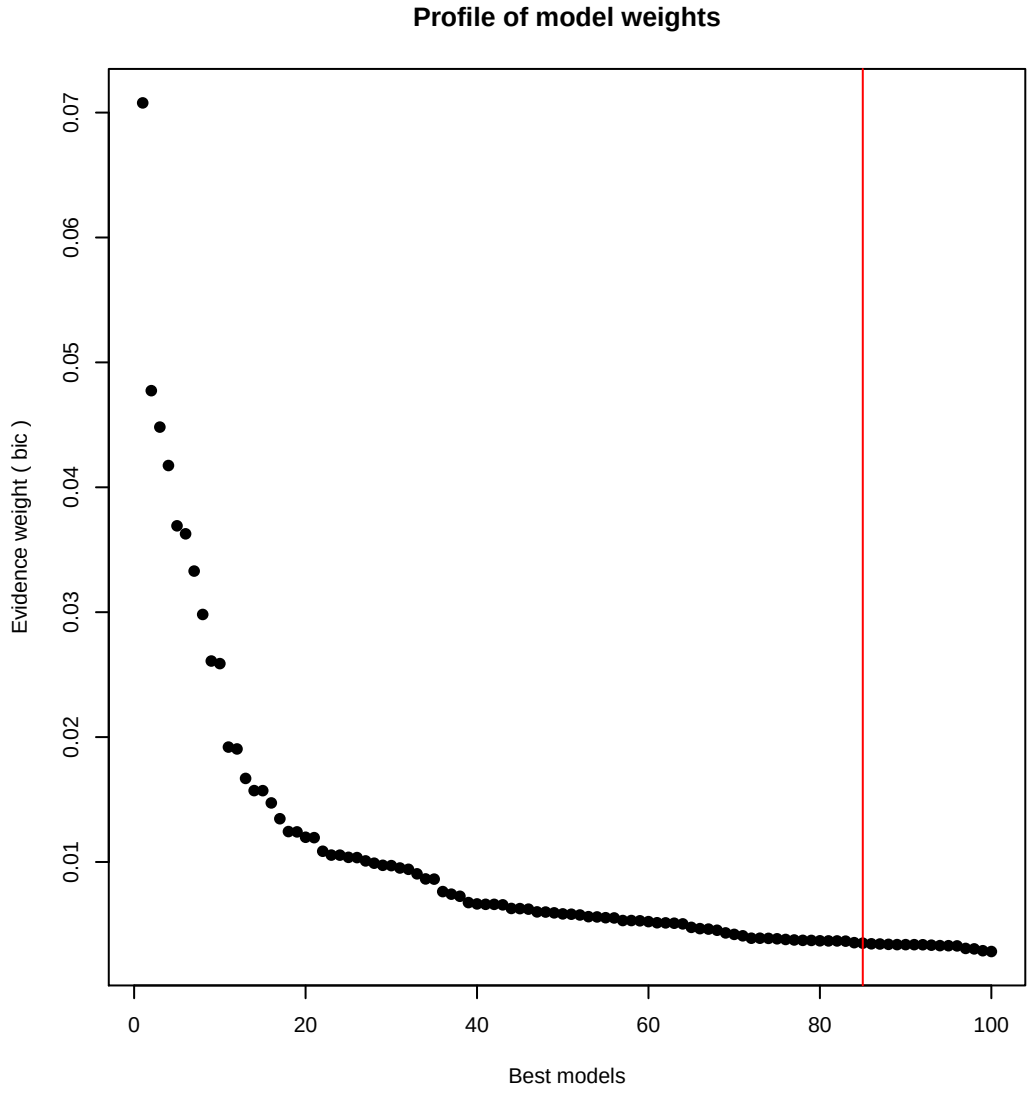
All-subsets regression




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




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




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

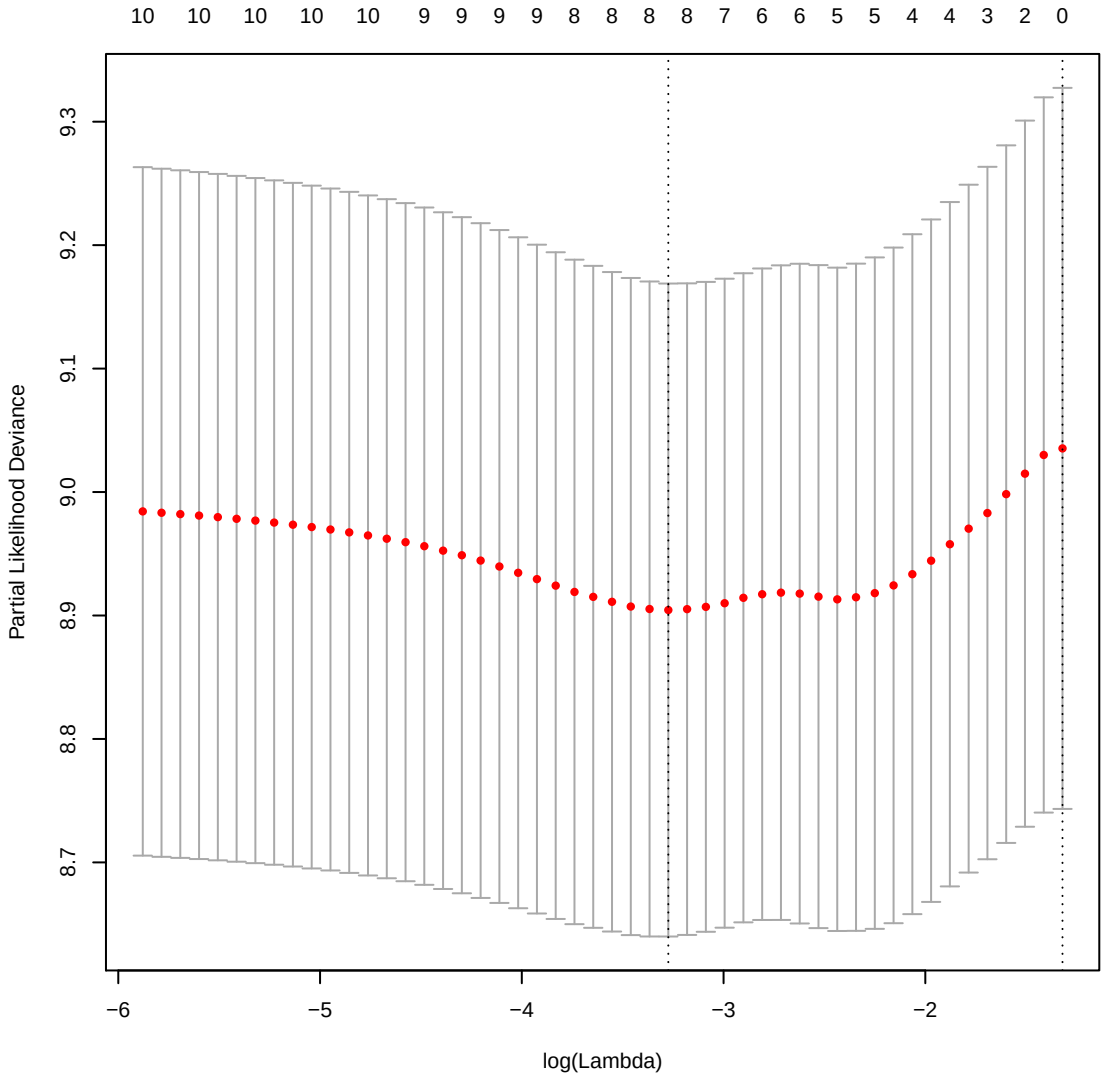
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##       1
## mg.1  .
## mg.2  .
## mg.3  .
## mg.4  .
## mg.5  .
## mg.6  .
## mg.7  .
## mg.8  .
## mg.9  .
## mg.10 .
```

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

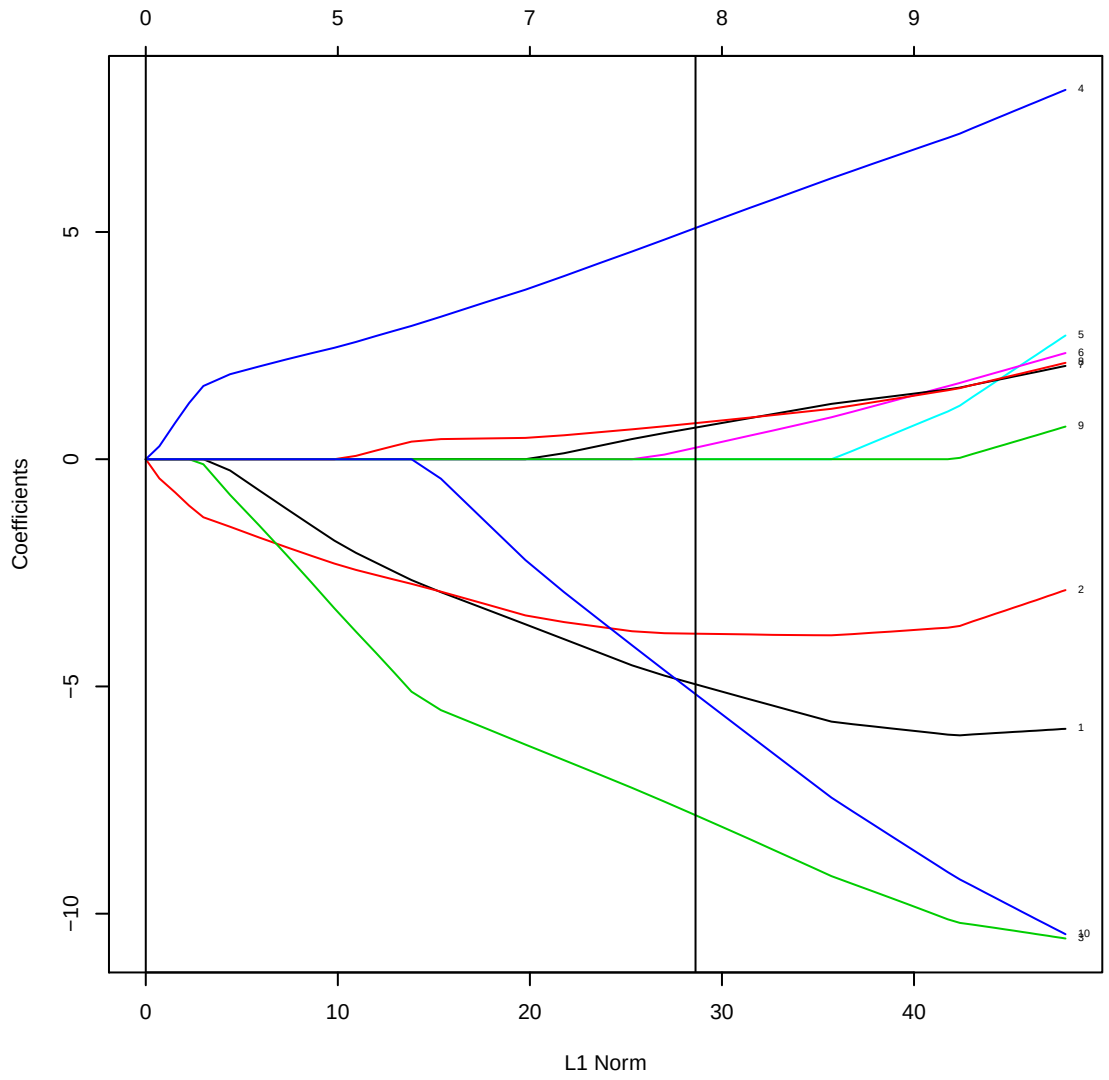
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  -4.9527
## mg.2  -3.8393
## mg.3  -7.8332
## mg.4   5.0886
## mg.5   .
## mg.6   0.2528
## mg.7   0.6955
## mg.8   0.7937
## mg.9   .
## mg.10 -5.1692
```

```
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)))
```



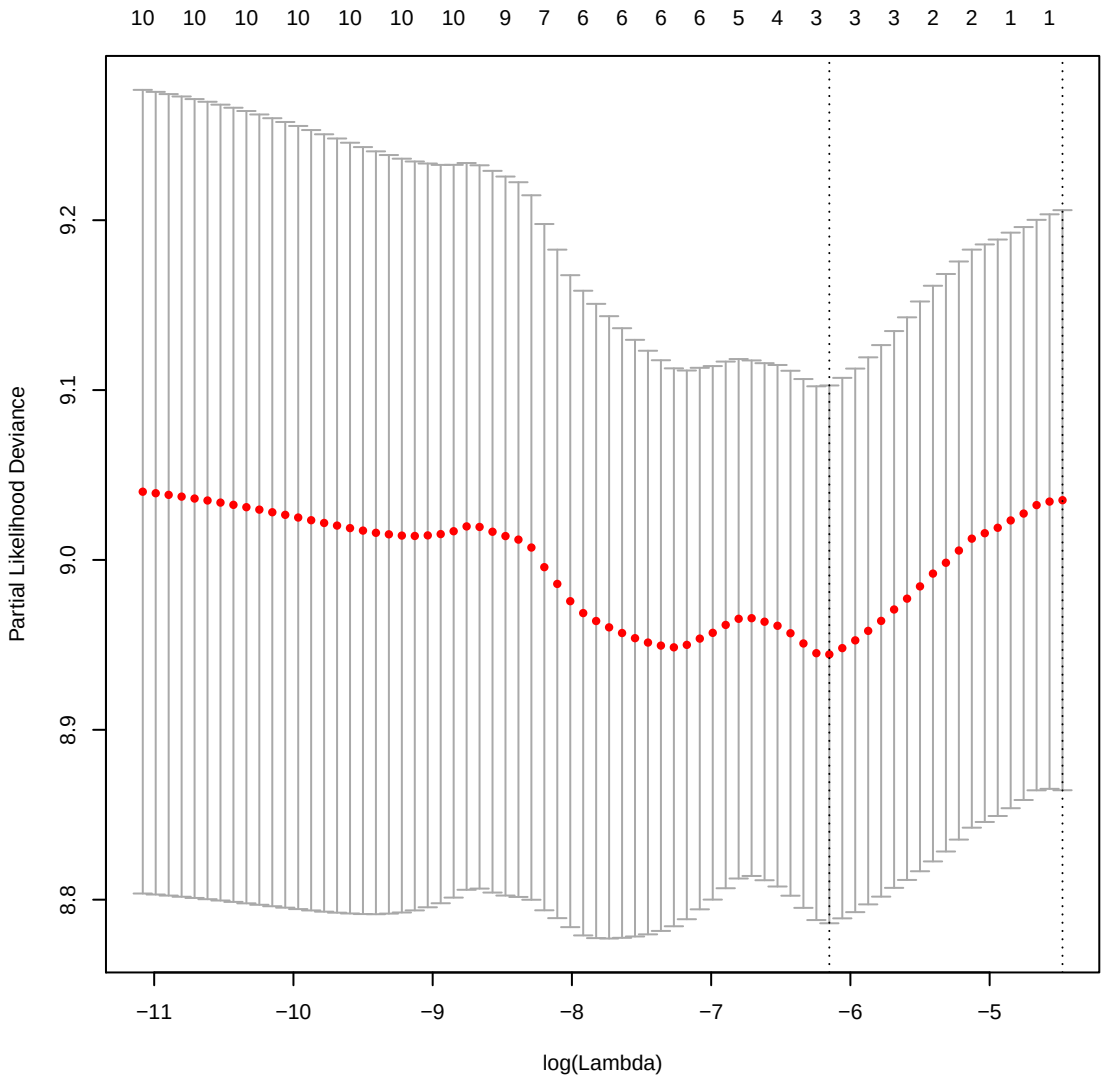

```
recr_dsd.adaglmnet.coef.1se/recr_dsd.adaglmnet.weights

## 10 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  .
## mg.10 .
```

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

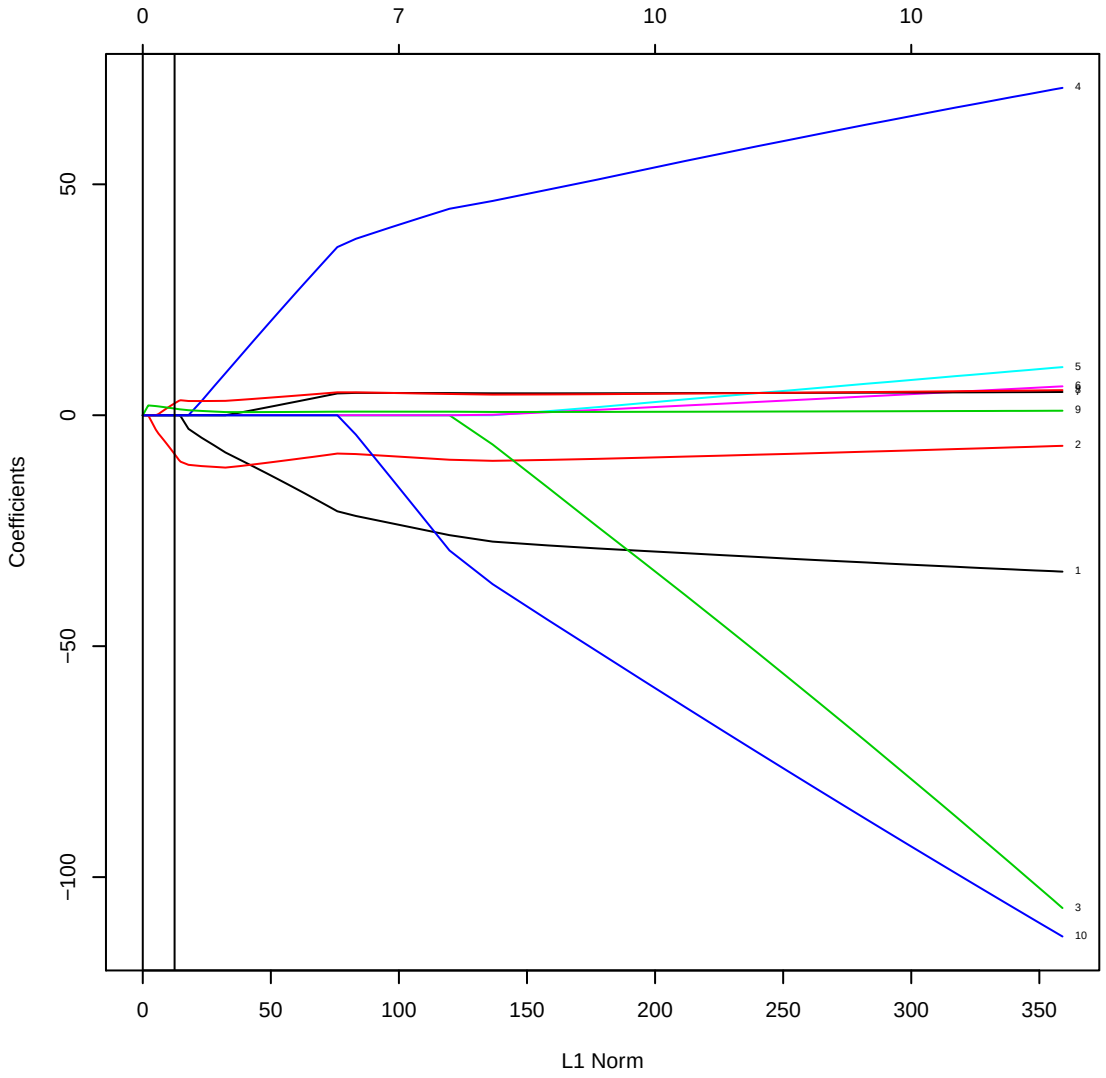
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1    .
## mg.2  -20.787
## mg.3    .
## mg.4    .
## mg.5    .
## mg.6    .
## mg.7    .
## mg.8    6.089
## mg.9    1.462
## mg.10   .
```

```
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)))
```



5 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_US.UTF-8          LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8           LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8       LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8          LC_NAME=en_US.UTF-8
##  [9] LC_ADDRESS=en_US.UTF-8        LC_TELEPHONE=en_US.UTF-8
## [11] LC_MEASUREMENT=en_US.UTF-8    LC_IDENTIFICATION=en_US.UTF-8
##
## attached base packages:
```

```
## [1] parallel  splines   methods   stats     graphics  grDevices utils
## [8] datasets  base
##
## other attached packages:
##  [1] snmfl_1.0          ahaz_1.14          survival_2.37-7
##  [4] gplots_2.14.2      RColorBrewer_1.0-5 energy_1.6.2
##  [7] glmnet_1.9-8       Matrix_1.1-4       glmulti_1.0.7
## [10] rJava_0.9-6
##
## loaded via a namespace (and not attached):
## [1] bitops_1.0-6       boot_1.3-11        caTools_1.17.1
## [4] gdata_2.13.3       grid_3.1.1         gtools_3.4.1
## [7] KernSmooth_2.23-12 lattice_0.20-29    Rcpp_0.11.3

sessionInfo()

## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8           LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8       LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8          LC_NAME=en_US.UTF-8
##  [9] LC_ADDRESS=en_US.UTF-8        LC_TELEPHONE=en_US.UTF-8
## [11] LC_MEASUREMENT=en_US.UTF-8    LC_IDENTIFICATION=en_US.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_0.20.4
## [10] synchronicity_1.1.4 bigmemory_4.4.6     BH_1.54.0-5
## [13] bigmemory.sri_0.1.3 Biobase_2.26.0      BiocGenerics_0.12.1
## [16] cluster_1.15.2      rngtools_1.2.4      pkgmaker_0.22
## [19] registry_0.2        energy_1.6.2        survival_2.37-7
## [22] knitr_1.8
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6       boot_1.3-11        caTools_1.17.1
##  [4] codetools_0.2-8    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_2.23-12 lattice_0.20-29    MASS_7.3-33
## [22] munsell_0.4.2      plyr_1.8.1         proto_0.3-10
## [25] Rcpp_0.11.3        reshape2_1.4       scales_0.2.4
## [28] stringr_0.6.2      tools_3.1.1
```