SIS NMF All Together Now

November 26, 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
```

```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")
```

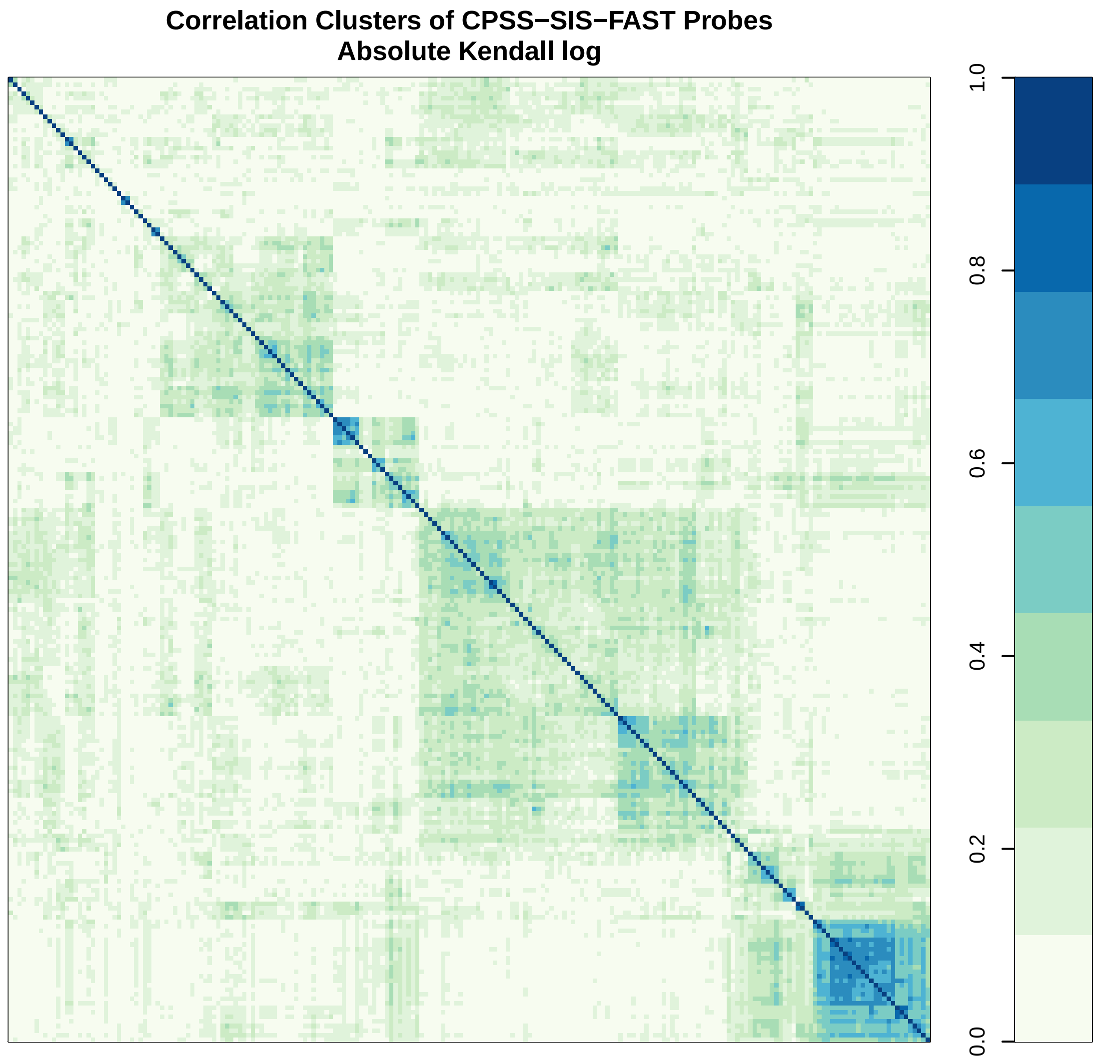
2 Probe selection

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

##
## FALSE  TRUE
## 12787   213

mean(cpss.sis$sel)

## [1] 0.01638
```

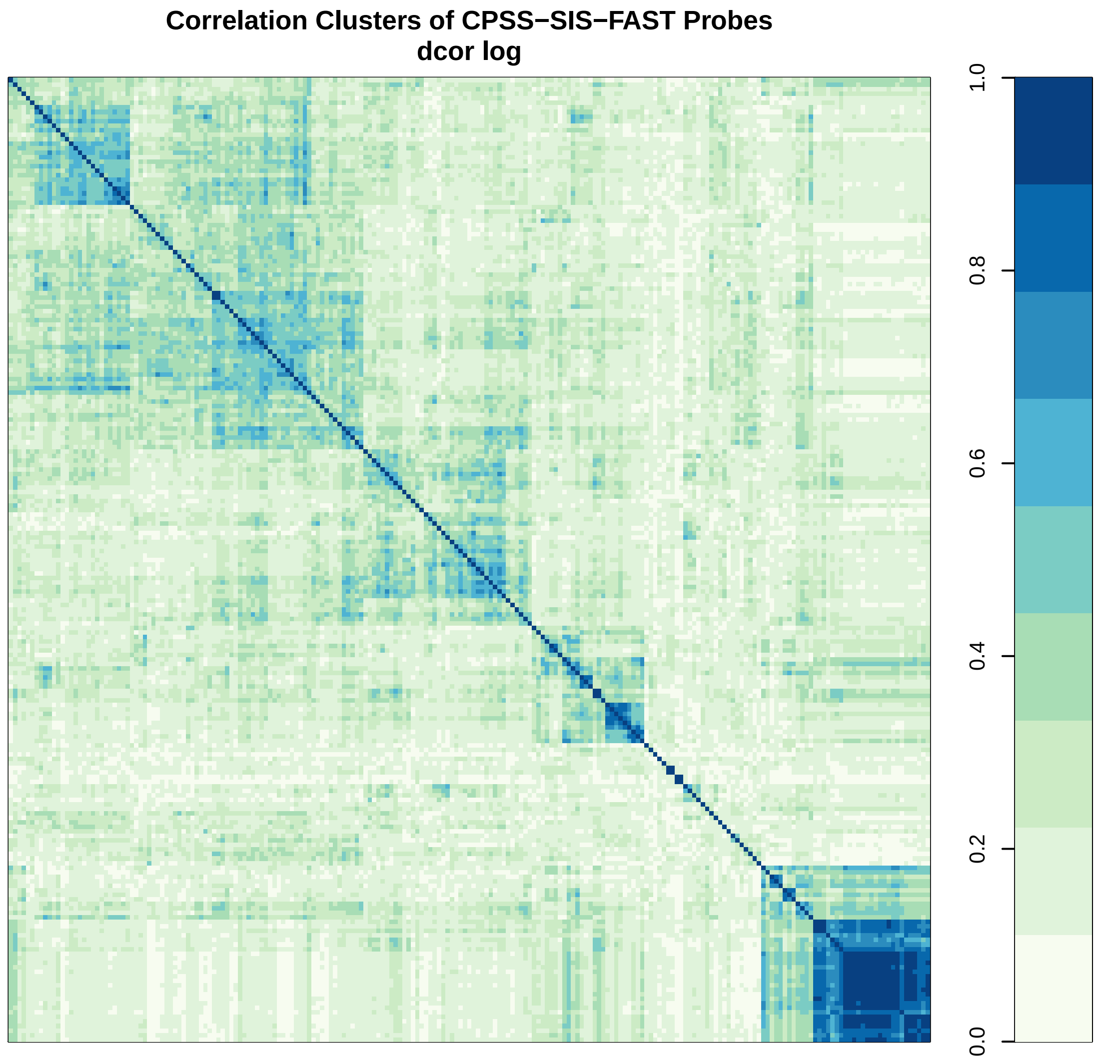
3 Expression correlation

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


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



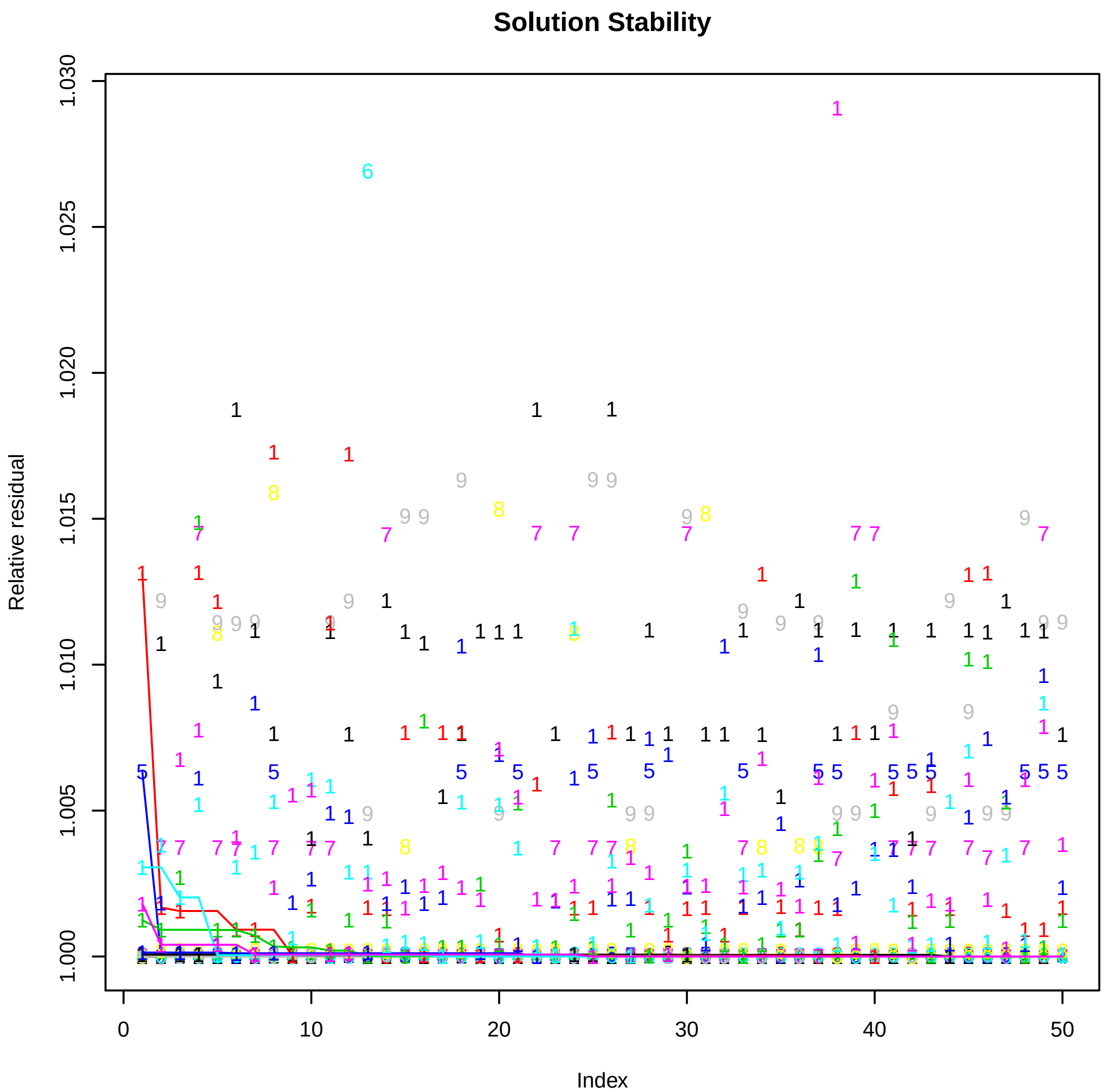

```
corPlot(x.sel.dcor, zlim = c(0, 1), pal = "GnBu", main = "Correlation Clusters of CPSS-SIS-FAST Probes\
    useRaster = FALSE)
```



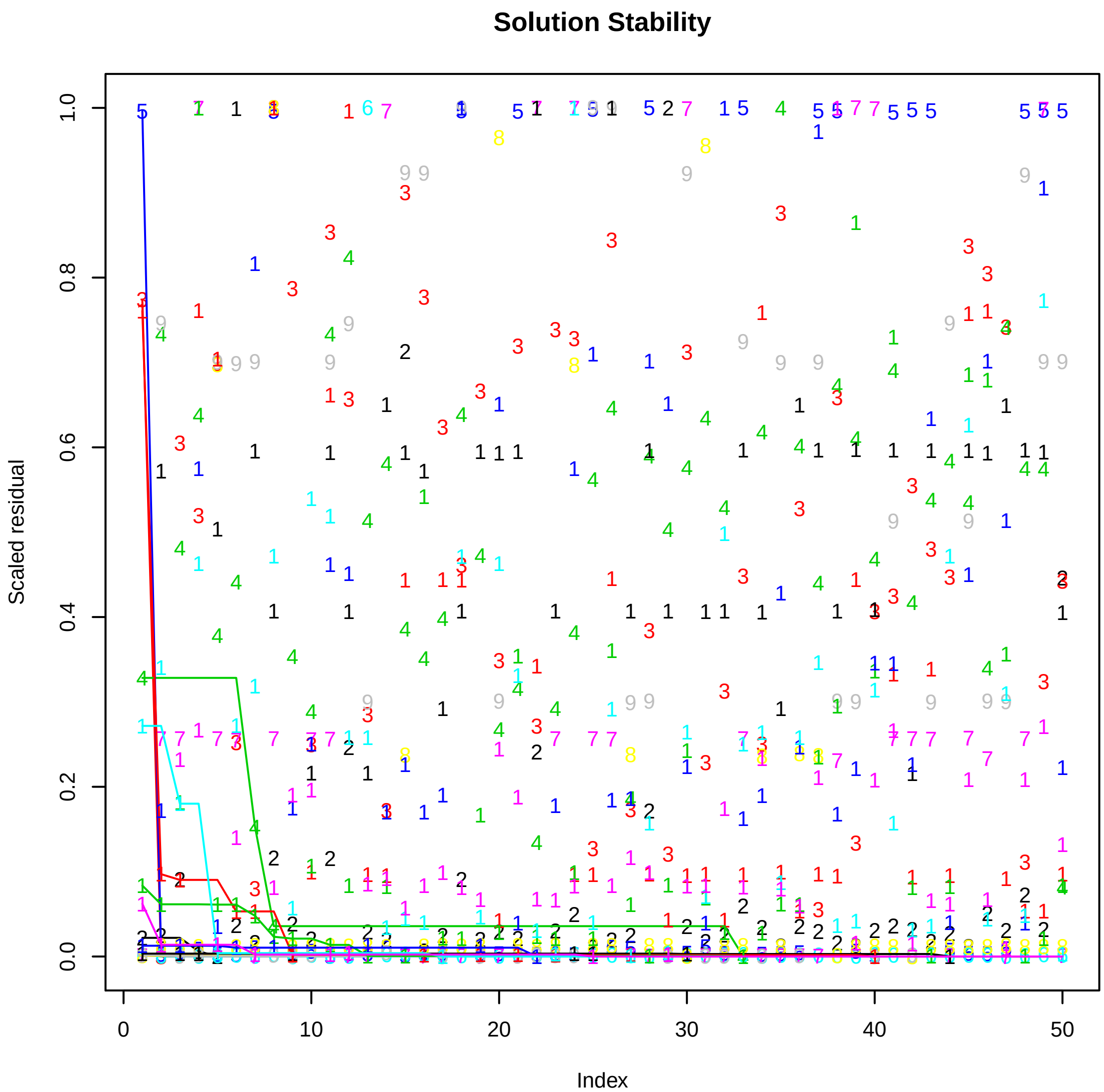
4 Factorization

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)
}
```

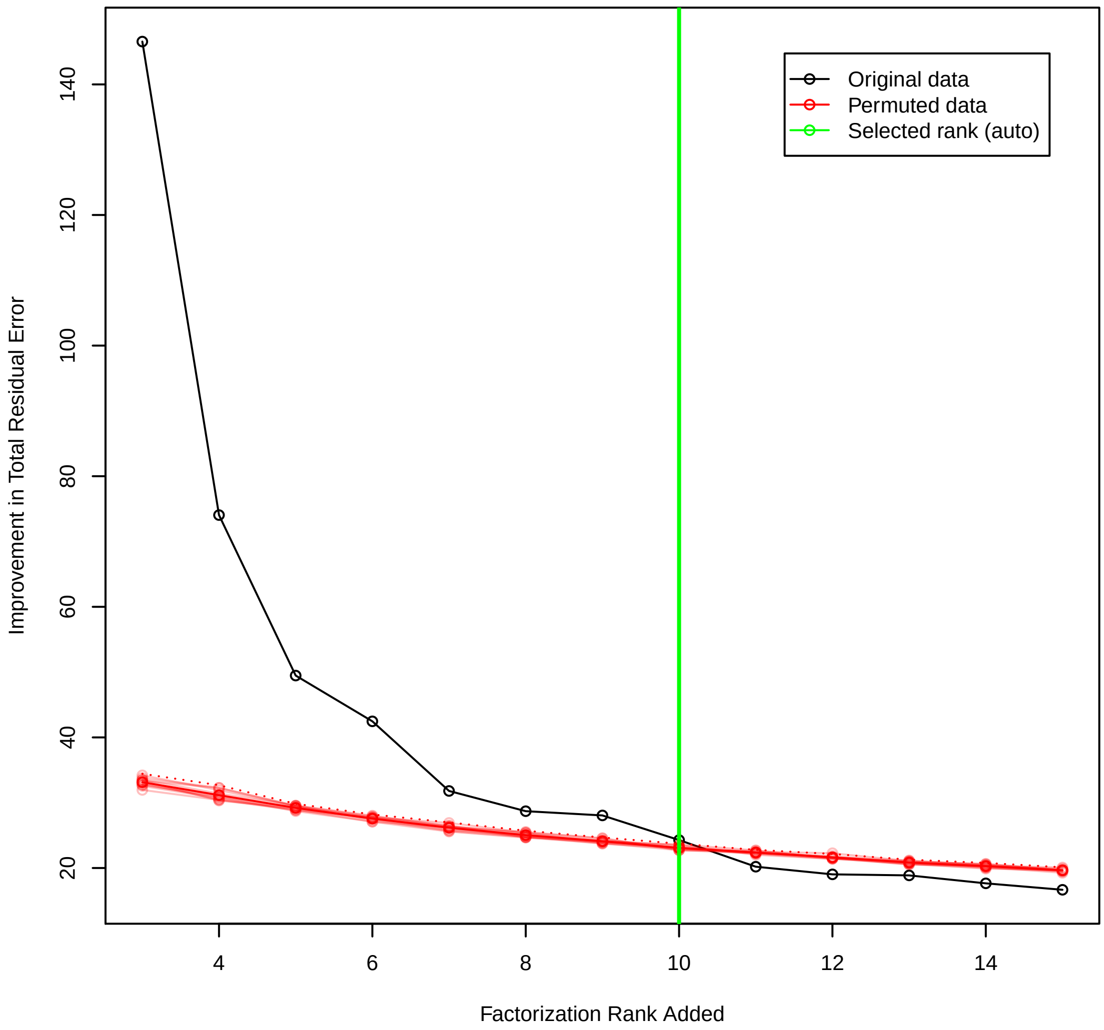



```r
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_scaled),
    ylab = "Scaled residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_scaled[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_scaled[,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



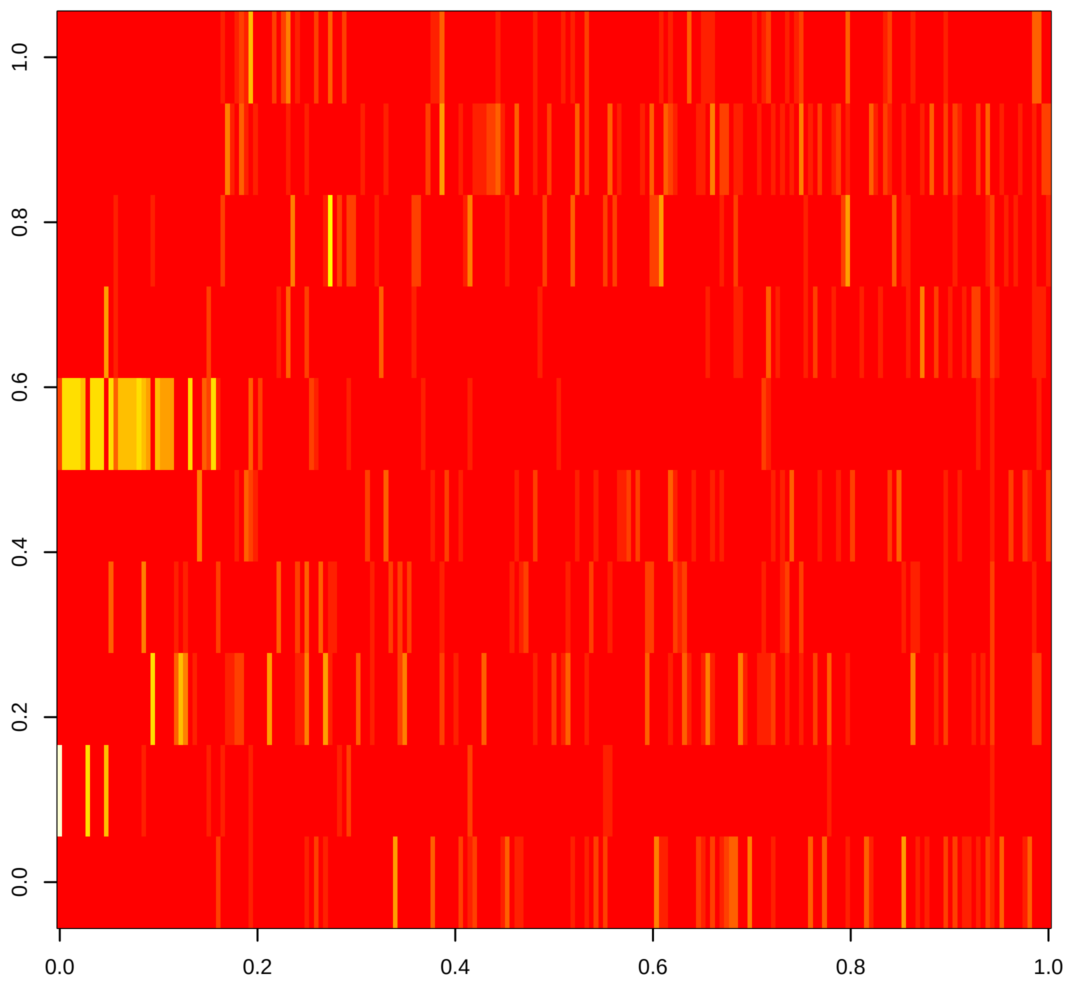

```
plot(nmf.rankrange[-1], -temp.orig_resids.delta, type = "o", col = "black",
    pch = 21, ylim = range(-c(temp.orig_resids.delta, temp.perm_resids.delta.mean)),
    xlab = "Factorization Rank Added", ylab = "Improvement in Total Residual Error")
lines(nmf.rankrange[-1], -temp.perm_resids.delta.mean,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

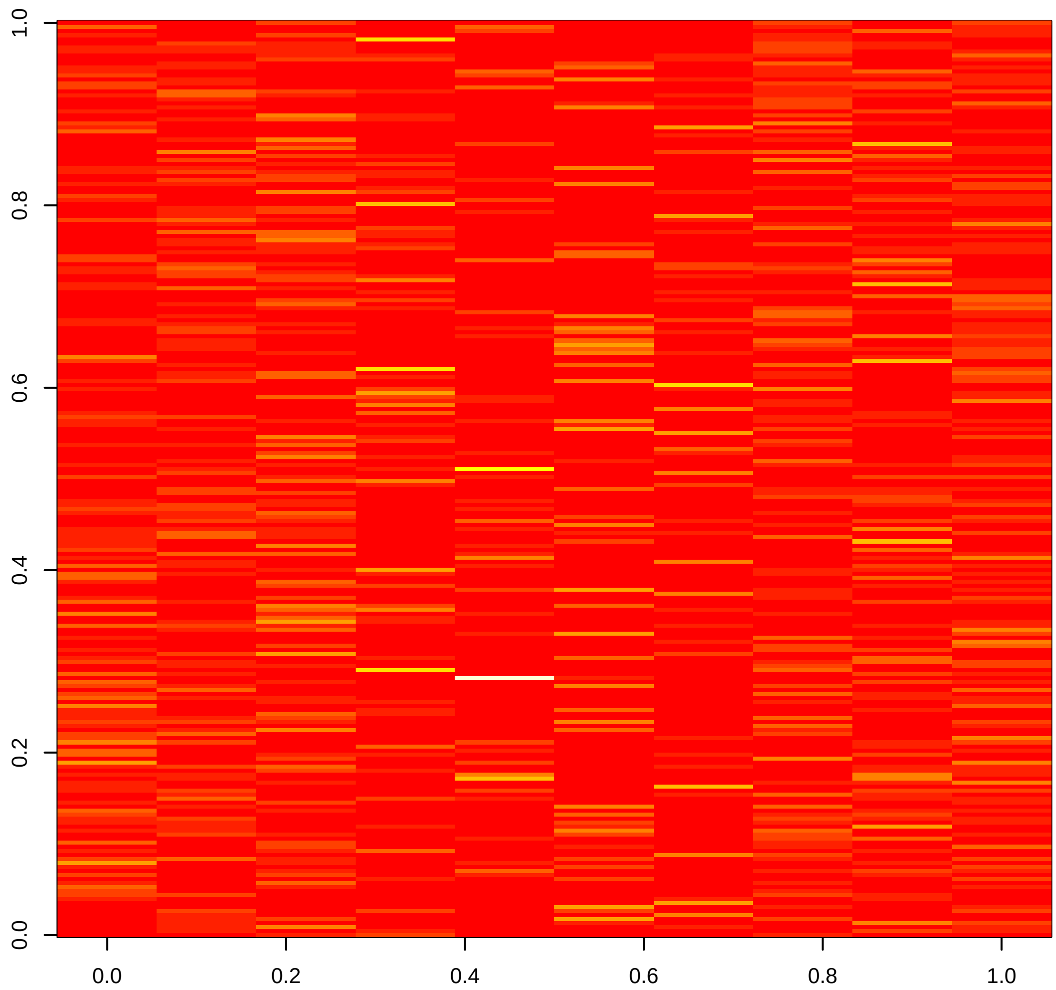


4.2 Fit

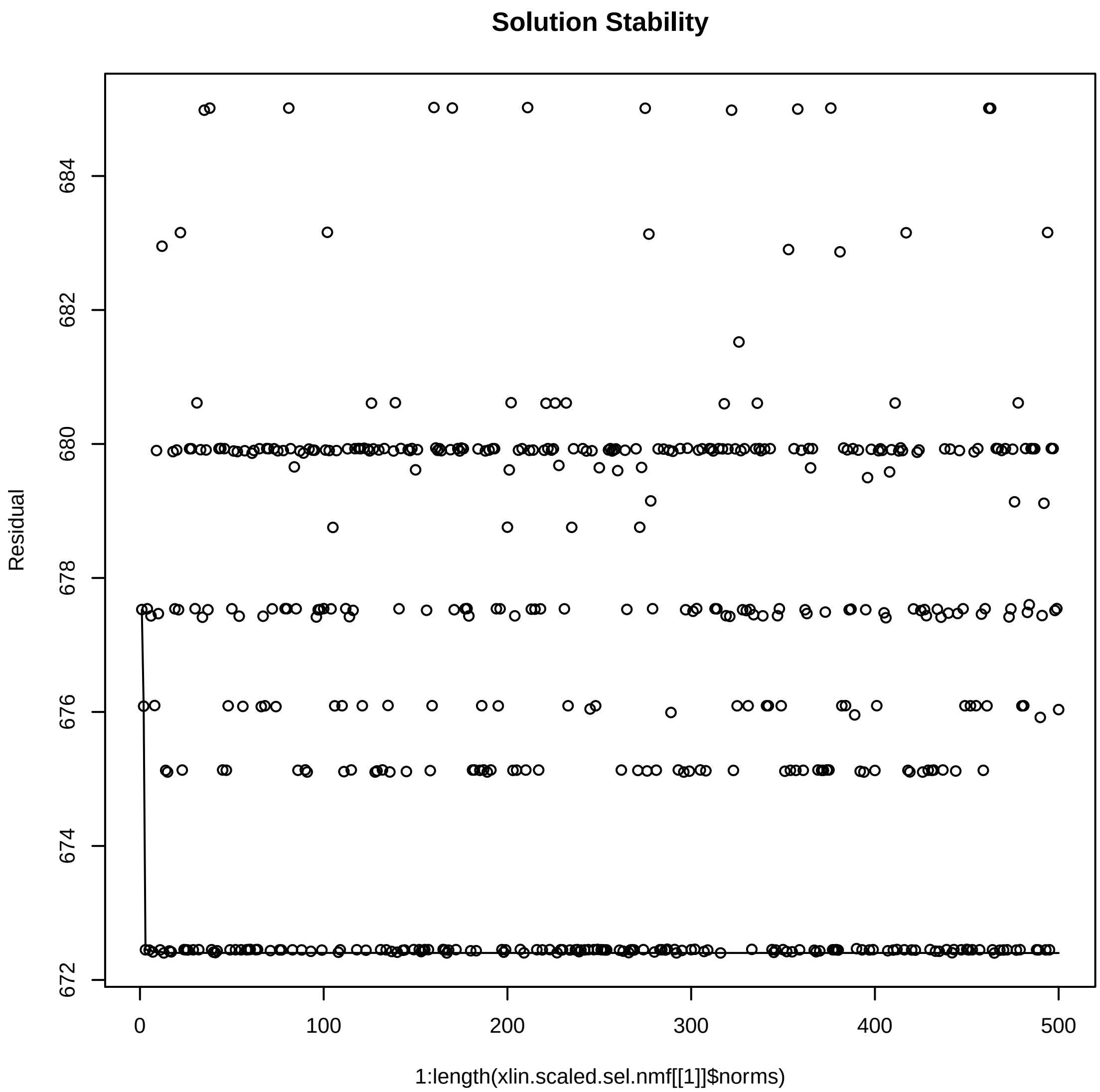
```
image(xlin.scaled.sel.nmf[[1]]$best_fit$W)
```




```
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]   4.97    144.69     1.59 3.12   0.0018
##
##                     exp(coef) exp(-coef) lower .95 upper .95
```

```
## coefs.diag_rec[, i]       145    0.00691      6.38      3284
##
## Concordance= 0.584  (se = 0.036 )
## Rsquare= 0.081   (max possible= 0.997 )
## Likelihood ratio test= 8.74  on 1 df,   p=0.00311
## Wald test            = 9.75  on 1 df,   p=0.00179
## Score (logrank) test = 10  on 1 df,   p=0.00157
##
## 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]  2.56     12.95     2.11 1.21     0.23
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      12.9     0.0772     0.206       814
##
## Concordance= 0.536  (se = 0.036 )
## Rsquare= 0.014   (max possible= 0.997 )
## Likelihood ratio test= 1.45  on 1 df,   p=0.229
## Wald test            = 1.47  on 1 df,   p=0.226
## Score (logrank) test = 1.47  on 1 df,   p=0.225
##
## 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] -2.190     0.112    1.563 -1.4     0.16
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.112       8.94   0.00523      2.39
##
## Concordance= 0.531  (se = 0.036 )
## Rsquare= 0.02   (max possible= 0.997 )
## Likelihood ratio test= 2.07  on 1 df,   p=0.15
## Wald test            = 1.96  on 1 df,   p=0.161
## Score (logrank) test = 1.98  on 1 df,   p=0.16
##
## 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] -7.462599  0.000574  4.100738 -1.82    0.069
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]  0.000574       1742  1.86e-07      1.78
##
```

```
## Concordance= 0.534  (se = 0.036 )
## Rsquare= 0.036   (max possible= 0.997 )
## Likelihood ratio test= 3.76  on 1 df,   p=0.0525
## Wald test            = 3.31  on 1 df,   p=0.0688
## Score (logrank) test = 3.35  on 1 df,   p=0.0673
##
## 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]    8.13   3399.79     2.36 3.44  0.00058
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]      3400   0.000294      33.2    348309
##
## Concordance= 0.614  (se = 0.034 )
## Rsquare= 0.085   (max possible= 0.997 )
## Likelihood ratio test= 9.18  on 1 df,   p=0.00244
## Wald test            = 11.8  on 1 df,   p=0.000576
## Score (logrank) test = 12.6  on 1 df,   p=0.000385
##
## 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] -1.889     0.151    1.455 -1.3     0.19
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]     0.151       6.61   0.00873      2.62
##
## Concordance= 0.561  (se = 0.035 )
## Rsquare= 0.017   (max possible= 0.997 )
## Likelihood ratio test= 1.82  on 1 df,   p=0.177
## Wald test            = 1.68  on 1 df,   p=0.194
## Score (logrank) test = 1.7  on 1 df,   p=0.192
##
## 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] -7.680357  0.000462  4.829472 -1.59     0.11
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]  0.000462       2165  3.58e-08      5.96
##
## Concordance= 0.56  (se = 0.036 )
## Rsquare= 0.027   (max possible= 0.997 )
```

```
## Likelihood ratio test= 2.84  on 1 df,   p=0.0918
## Wald test            = 2.53  on 1 df,   p=0.112
## Score (logrank) test = 2.55  on 1 df,   p=0.111
##
## 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] -3.5193    0.0296   1.8904 -1.86    0.063
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]    0.0296       33.8  0.000729       1.2
##
## Concordance= 0.568  (se = 0.036 )
## Rsquare= 0.035   (max possible= 0.997 )
## Likelihood ratio test= 3.69  on 1 df,   p=0.0546
## Wald test            = 3.47  on 1 df,   p=0.0626
## Score (logrank) test = 3.5  on 1 df,   p=0.0613
##
## 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]   6.68    796.91     1.79 3.73    2e-04
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]       797    0.00125      23.7     26784
##
## Concordance= 0.649  (se = 0.036 )
## Rsquare= 0.108   (max possible= 0.997 )
## Likelihood ratio test= 11.8  on 1 df,   p=0.000579
## Wald test            = 13.9  on 1 df,   p=0.000195
## Score (logrank) test = 14.2  on 1 df,   p=0.000162
##
## 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]   4.80    121.23     1.87 2.57     0.01
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_rec[, i]       121    0.00825      3.11      4731
##
## Concordance= 0.577  (se = 0.036 )
## Rsquare= 0.057   (max possible= 0.997 )
## Likelihood ratio test= 6.13  on 1 df,   p=0.0133
## Wald test            = 6.59  on 1 df,   p=0.0103
## Score (logrank) test = 6.66  on 1 df,   p=0.00986
```

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]   5.74    309.95     1.57 3.66  0.00025
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]       310    0.00323      14.3      6703
##
## Concordance= 0.605  (se = 0.037 )
## Rsquare= 0.1   (max possible= 0.995 )
## Likelihood ratio test= 11.6  on 1 df,   p=0.000662
## Wald test            = 13.4  on 1 df,   p=0.000255
## Score (logrank) test = 13.8  on 1 df,   p=0.000202
##
## 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] 1.52      4.55     2.26 0.67      0.5
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      4.55       0.22    0.0544       381
##
## Concordance= 0.518  (se = 0.037 )
## Rsquare= 0.004   (max possible= 0.995 )
## Likelihood ratio test= 0.45  on 1 df,   p=0.504
## Wald test            = 0.45  on 1 df,   p=0.502
## Score (logrank) test = 0.45  on 1 df,   p=0.502
##
## 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] -4.4353    0.0119   1.8344 -2.42    0.016
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]    0.0119       84.4  0.000325     0.432
##
## Concordance= 0.585  (se = 0.038 )
## Rsquare= 0.058   (max possible= 0.995 )
## Likelihood ratio test= 6.54  on 1 df,   p=0.0105
```

```
## Wald test            = 5.85  on 1 df,   p=0.0156
## Score (logrank) test = 5.97  on 1 df,   p=0.0146
##
## 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] -0.691     0.501    3.743 -0.18     0.85
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]     0.501          2  0.000326       769
##
## Concordance= 0.474  (se = 0.037 )
## Rsquare= 0   (max possible= 0.995 )
## Likelihood ratio test= 0.03  on 1 df,   p=0.853
## Wald test            = 0.03  on 1 df,   p=0.853
## Score (logrank) test = 0.03  on 1 df,   p=0.853
##
## 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]    7.89   2669.83     2.19 3.6  0.00031
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      2670   0.000375      36.6    194908
##
## Concordance= 0.636  (se = 0.036 )
## Rsquare= 0.088   (max possible= 0.995 )
## Likelihood ratio test= 10.2  on 1 df,   p=0.00143
## Wald test            = 13  on 1 df,   p=0.000313
## Score (logrank) test = 13.9  on 1 df,   p=0.000195
##
## 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] -0.915     0.400    1.457 -0.63     0.53
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]       0.4        2.5     0.023      6.96
##
## Concordance= 0.549  (se = 0.036 )
## Rsquare= 0.004   (max possible= 0.995 )
## Likelihood ratio test= 0.41  on 1 df,   p=0.522
## Wald test            = 0.39  on 1 df,   p=0.53
## Score (logrank) test = 0.4  on 1 df,   p=0.529
##
```

```
## 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] -8.18078   0.00028  5.34100 -1.53     0.13
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]   0.00028       3572  7.96e-09      9.85
##
## Concordance= 0.571  (se = 0.037 )
## Rsquare= 0.024   (max possible= 0.995 )
## Likelihood ratio test= 2.65  on 1 df,   p=0.103
## Wald test            = 2.35  on 1 df,   p=0.126
## Score (logrank) test = 2.36  on 1 df,   p=0.124
##
## 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] -5.71723   0.00329  2.20548 -2.59   0.0095
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]   0.00329        304  4.36e-05     0.248
##
## Concordance= 0.601  (se = 0.038 )
## Rsquare= 0.065   (max possible= 0.995 )
## Likelihood ratio test= 7.45  on 1 df,   p=0.00635
## Wald test            = 6.72  on 1 df,   p=0.00953
## Score (logrank) test = 6.86  on 1 df,   p=0.00881
##
## 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]    9.14   9305.72     1.86 4.91    9e-07
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      9306   0.000107       243    356580
##
## Concordance= 0.67  (se = 0.037 )
## Rsquare= 0.166   (max possible= 0.995 )
## Likelihood ratio test= 20  on 1 df,   p=7.87e-06
## Wald test            = 24.1  on 1 df,   p=8.99e-07
## Score (logrank) test = 25.3  on 1 df,   p=4.85e-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] 0.292     1.340    1.965 0.15     0.88
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.diag_dsd[, i]      1.34      0.746    0.0285        63
##
## Concordance= 0.505  (se = 0.037 )
## Rsquare= 0   (max possible= 0.995 )
## Likelihood ratio test= 0.02  on 1 df,   p=0.882
## Wald test            = 0.02  on 1 df,   p=0.882
## Score (logrank) test = 0.02  on 1 df,   p=0.882
```

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]  3.78     43.60     1.66 2.28    0.023
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      43.6     0.0229       1.7      1120
##
## Concordance= 0.583  (se = 0.041 )
## Rsquare= 0.057   (max possible= 0.997 )
## Likelihood ratio test= 4.73  on 1 df,   p=0.0297
## Wald test            = 5.2  on 1 df,   p=0.0226
## Score (logrank) test = 5.27  on 1 df,   p=0.0217
##
## 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] 0.288     1.333    2.376 0.12      0.9
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      1.33       0.75    0.0127       140
##
## Concordance= 0.479  (se = 0.041 )
## Rsquare= 0   (max possible= 0.997 )
## Likelihood ratio test= 0.01  on 1 df,   p=0.904
## Wald test            = 0.01  on 1 df,   p=0.904
```

```
## Score (logrank) test = 0.01  on 1 df,   p=0.904
##
## 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] -3.1036    0.0449   1.9670 -1.58     0.11
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]    0.0449       22.3   0.00095      2.12
##
## Concordance= 0.565  (se = 0.041 )
## Rsquare= 0.032   (max possible= 0.997 )
## Likelihood ratio test= 2.65  on 1 df,   p=0.103
## Wald test            = 2.49  on 1 df,   p=0.115
## Score (logrank) test = 2.51  on 1 df,   p=0.113
##
## 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]    10.35  31285.03     3.84 2.69   0.0071
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]     31285    3.2e-05      16.7  58623053
##
## Concordance= 0.607  (se = 0.041 )
## Rsquare= 0.071   (max possible= 0.997 )
## Likelihood ratio test= 5.98  on 1 df,   p=0.0145
## Wald test            = 7.25  on 1 df,   p=0.0071
## Score (logrank) test = 7.5  on 1 df,   p=0.00616
##
## 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]  3.53     34.27     2.48 1.43     0.15
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      34.3     0.0292     0.266      4421
##
## Concordance= 0.576  (se = 0.04 )
## Rsquare= 0.023   (max possible= 0.997 )
## Likelihood ratio test= 1.85  on 1 df,   p=0.174
## Wald test            = 2.03  on 1 df,   p=0.154
## Score (logrank) test = 2.05  on 1 df,   p=0.152
##
```

```
## 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] 0.824     2.279    1.534 0.54     0.59
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]      2.28      0.439     0.113      46.1
##
## Concordance= 0.481  (se = 0.039 )
## Rsquare= 0.003   (max possible= 0.997 )
## Likelihood ratio test= 0.28  on 1 df,   p=0.597
## Wald test            = 0.29  on 1 df,   p=0.591
## Score (logrank) test = 0.29  on 1 df,   p=0.591
##
## 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] -1.015     0.362    5.619 -0.18     0.86
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]     0.362       2.76  5.98e-06     21955
##
## Concordance= 0.555  (se = 0.041 )
## Rsquare= 0   (max possible= 0.997 )
## Likelihood ratio test= 0.03  on 1 df,   p=0.856
## Wald test            = 0.03  on 1 df,   p=0.857
## Score (logrank) test = 0.03  on 1 df,   p=0.857
##
## 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] -7.331516  0.000655  2.692037 -2.72   0.0065
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]  0.000655       1528  3.35e-06     0.128
##
## Concordance= 0.615  (se = 0.041 )
## Rsquare= 0.096   (max possible= 0.997 )
## Likelihood ratio test= 8.15  on 1 df,   p=0.00431
## Wald test            = 7.42  on 1 df,   p=0.00646
## Score (logrank) test = 7.61  on 1 df,   p=0.00581
##
## 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]   6.44    627.68     2.02 3.19   0.0014
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]       628    0.00159      11.9     32983
##
## Concordance= 0.645  (se = 0.041 )
## Rsquare= 0.104   (max possible= 0.997 )
## Likelihood ratio test= 8.94  on 1 df,   p=0.00279
## Wald test            = 10.2  on 1 df,   p=0.00144
## Score (logrank) test = 10.5  on 1 df,   p=0.0012
##
## 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] -5.3826    0.0046   2.3061 -2.33     0.02
##
##                       exp(coef) exp(-coef) lower .95 upper .95
## coefs.recr_dsd[, i]    0.0046        218  5.01e-05     0.422
##
## Concordance= 0.594  (se = 0.041 )
## Rsquare= 0.071   (max possible= 0.997 )
## Likelihood ratio test= 5.98  on 1 df,   p=0.0144
## Wald test            = 5.45  on 1 df,   p=0.0196
## Score (logrank) test = 5.52  on 1 df,   p=0.0188
```

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 = -2.445, p-value = 0.01451
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.1387
##
##
## $mg.2
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
```

```
## z = -3.775, p-value = 0.0001601
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.2178
##
##
## $mg.3
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 1.736, p-value = 0.08255
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## 0.09819
##
##
## $mg.4
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 2.413, p-value = 0.01584
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.1385
##
##
## $mg.5
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 0.3436, p-value = 0.7311
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.0201
##
##
## $mg.6
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.799, p-value = 0.0001451
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## -0.226
##
```

```
##
## $mg.7
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 1.838, p-value = 0.06613
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.1053
##
##
## $mg.8
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -5.858, p-value = 4.685e-09
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## -0.332
##
##
## $mg.9
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -1.058, p-value = 0.29
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.05962
##
##
## $mg.10
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.262, p-value = 0.001108
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.1848
```

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
```

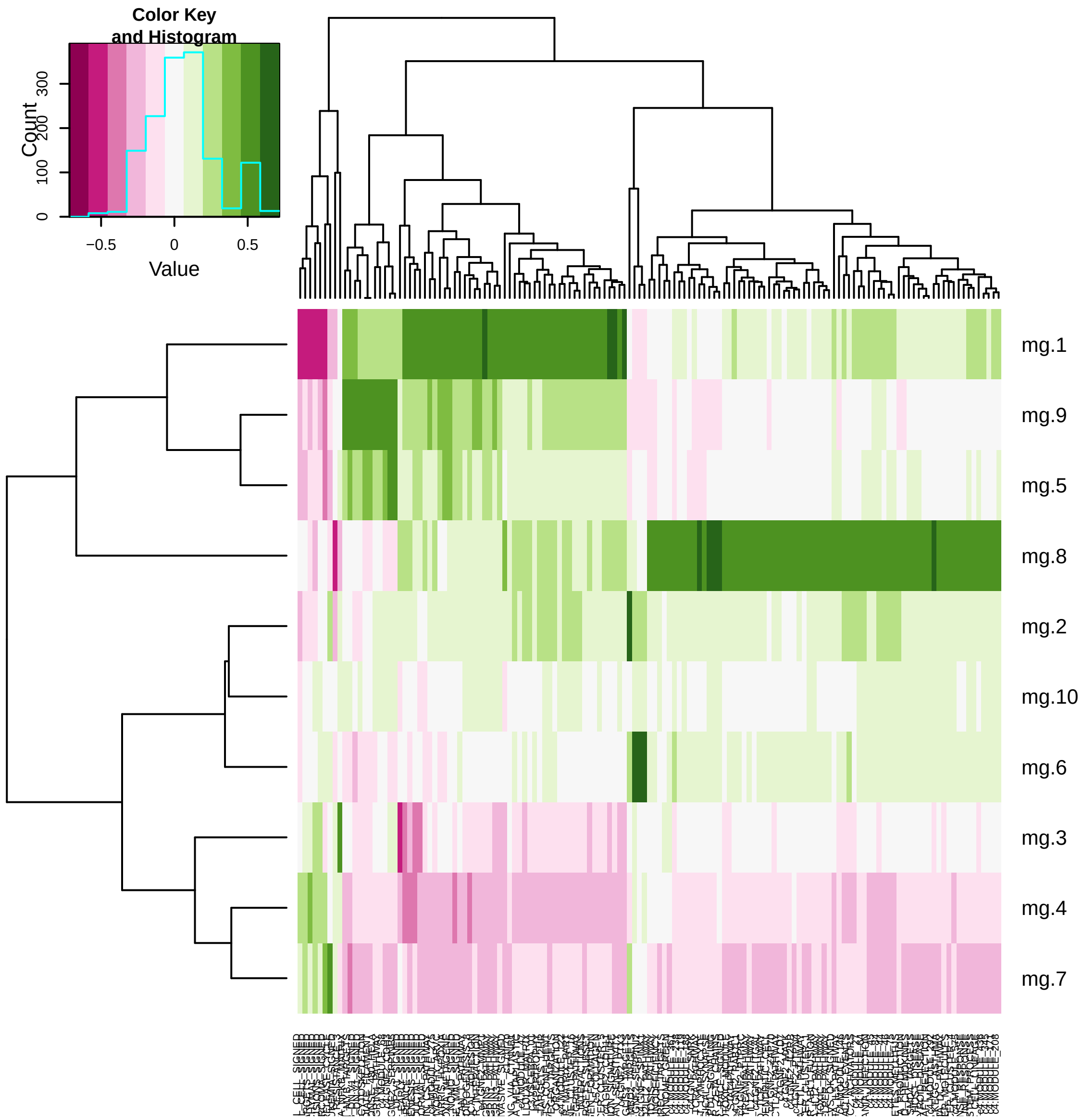
```
##          pure.p pure.p.BY pure.p.Holm  pure.s surv.diag_dsd.c
## mg.1  1.451e-02 1.379e-01   3.609e-01 -2.4445          5.7364
## mg.2  1.601e-04 6.849e-03   5.923e-03 -3.7749          1.5155
## mg.3  8.255e-02 5.887e-01   1.000e+00  1.7361         -4.4353
## mg.4  1.584e-02 1.426e-01   3.609e-01  2.4127         -0.6915
## mg.5  7.311e-01 1.000e+00   1.000e+00  0.3436          7.8898
## mg.6  1.451e-04 6.849e-03   5.512e-03 -3.7994         -0.9154
## mg.7  6.613e-02 4.920e-01   1.000e+00  1.8376         -8.1808
## mg.8  4.685e-09 8.017e-07   1.874e-07 -5.8580         -5.7172
## mg.9  2.900e-01 1.000e+00   1.000e+00 -1.0582          9.1384
## mg.10 1.108e-03 2.709e-02   3.767e-02 -3.2616          0.2924
##       surv.diag_dsd.p surv.diag_dsd.p.BY surv.diag_dsd.p.Holm
## mg.1        6.622e-04          0.0188889            0.0231777
## mg.2        5.038e-01          1.0000000            1.0000000
## mg.3        1.053e-02          0.1287507            0.2843711
## mg.4        8.525e-01          1.0000000            1.0000000
## mg.5        1.427e-03          0.0305191            0.0470781
## mg.6        5.222e-01          1.0000000            1.0000000
## mg.7        1.034e-01          0.6551620            1.0000000
## mg.8        6.352e-03          0.0836289            0.1778696
## mg.9        7.874e-06          0.0006738            0.0003071
## mg.10       8.820e-01          1.0000000            1.0000000
##       surv.diag_rec.c surv.diag_rec.p surv.diag_rec.p.BY
## mg.1            4.975       0.0031134            0.04844
## mg.2            2.561       0.2291157            1.00000
## mg.3           -2.190       0.1497507            0.91531
## mg.4           -7.463       0.0524712            0.42503
## mg.5            8.131       0.0024445            0.04648
## mg.6           -1.889       0.1770235            1.00000
## mg.7           -7.680       0.0917694            0.62822
## mg.8           -3.519       0.0546372            0.42503
## mg.9            6.681       0.0005792            0.01889
## mg.10           4.798       0.0133283            0.13792
##       surv.diag_rec.p.Holm surv.recr_dsd.c surv.recr_dsd.p
## mg.1               0.09340          3.7750        0.029718
```

	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	0.3609	-2.4445	5.7364	0.0232	4.975	0.0934	3.7750	0.6241
mg.2	0.0059	-3.7749	1.5155	1.0000	2.561	1.0000	0.2878	1.0000
mg.3	1.0000	1.7361	-4.4353	0.2844	-2.190	1.0000	-3.1036	1.0000
mg.4	0.3609	2.4127	-0.6915	1.0000	-7.463	1.0000	10.3509	0.3609
mg.5	1.0000	0.3436	7.8898	0.0471	8.132	0.0782	3.5344	1.0000
mg.6	0.0055	-3.7994	-0.9154	1.0000	-1.889	1.0000	0.8239	1.0000
mg.7	1.0000	1.8376	-8.1808	1.0000	-7.680	1.0000	-1.0154	1.0000
mg.8	0.0000	-5.8580	-5.7172	0.1779	-3.519	1.0000	-7.3315	0.1250
mg.9	1.0000	-1.0582	9.1384	0.0003	6.681	0.0208	6.4420	0.0866
mg.10	0.0377	-3.2616	0.2924	1.0000	4.798	0.3465	-5.3826	0.3609

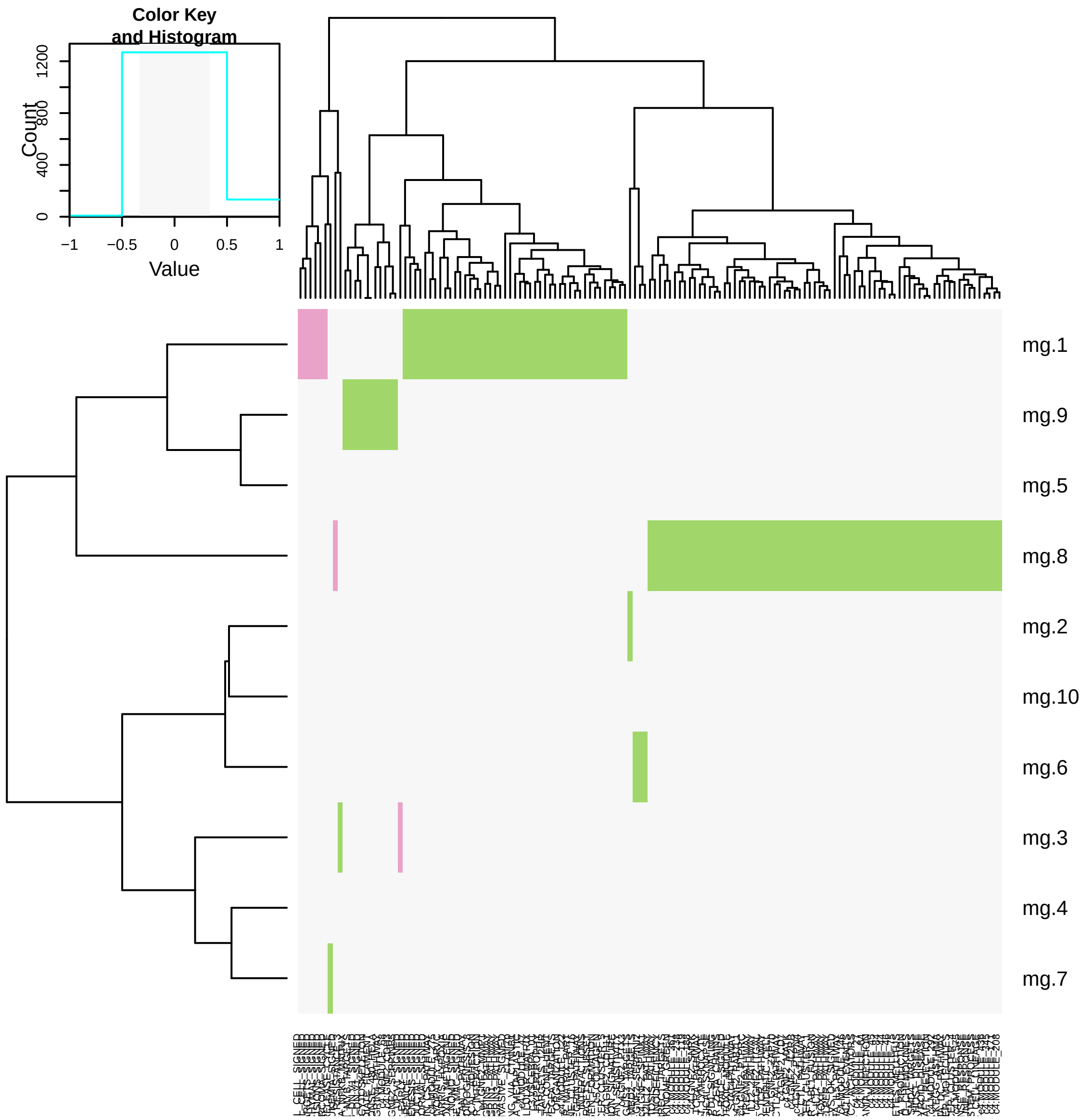
```
## mg.2               1.00000        0.2878       0.903616
## mg.3               1.00000       -3.1036       0.103227
## mg.4               1.00000       10.3509       0.014497
## mg.5               0.07822        3.5344       0.174125
## mg.6               1.00000        0.8239       0.597355
## mg.7               1.00000       -1.0154       0.855617
## mg.8               1.00000       -7.3315       0.004310
## mg.9               0.02085        6.4420       0.002793
## mg.10              0.34654       -5.3826       0.014435
##       surv.recr_dsd.p.BY surv.recr_dsd.p.Holm
## mg.1             0.25430              0.62408
## mg.2             1.00000              1.00000
## mg.3             0.65516              1.00000
## mg.4             0.13792              0.36088
## mg.5             1.00000              1.00000
## mg.6             1.00000              1.00000
## mg.7             1.00000              1.00000
## mg.8             0.06146              0.12498
## mg.9             0.04781              0.08659
## mg.10            0.13792              0.36088
```

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
```

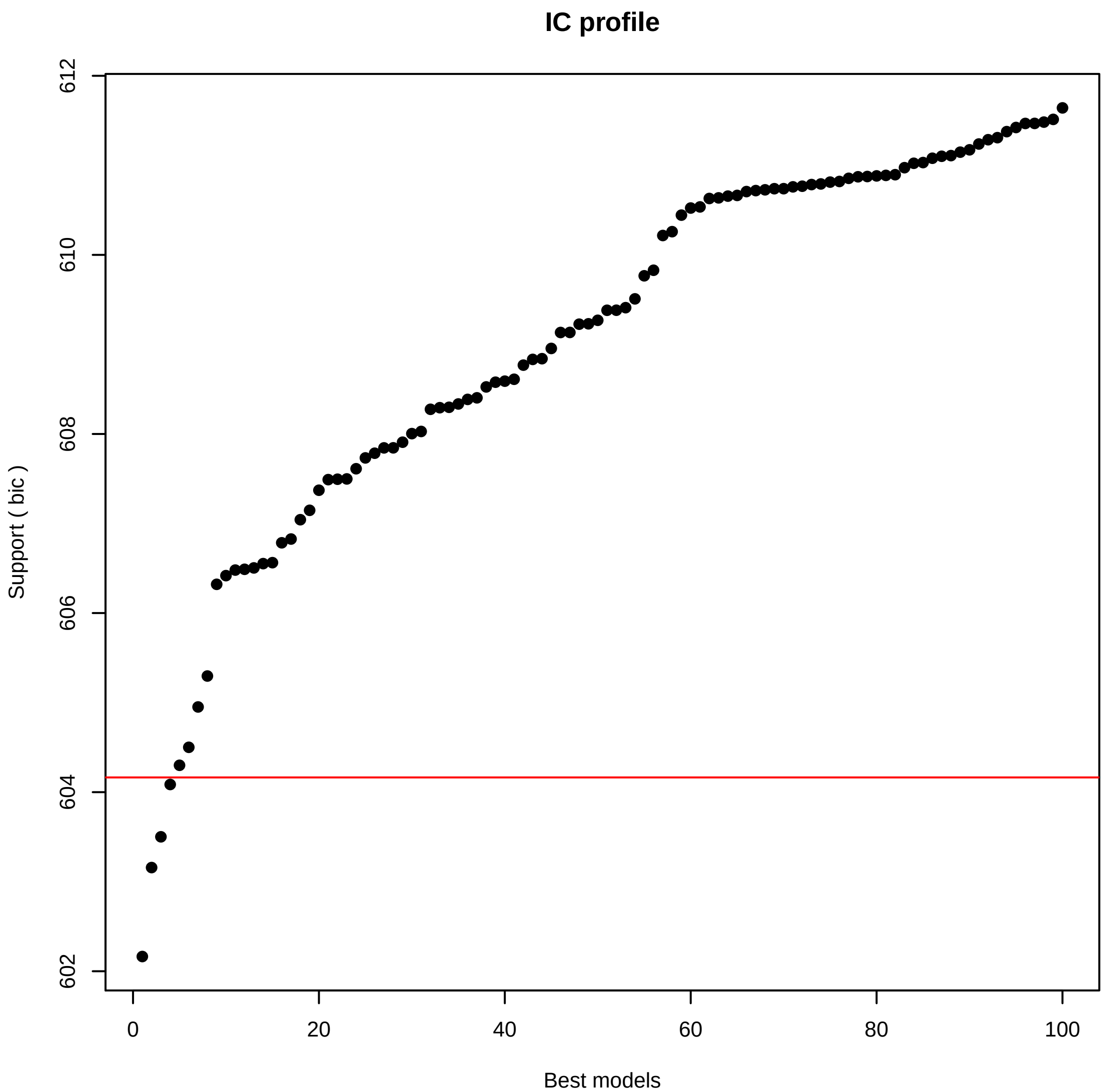
4.5.1 Outcome: Diagnosis to recurrence

```
print(diag_rec.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 100 models:
## Best IC: 602.164462773173
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.1 + mg.2 + mg.9 + mg.10"
## Evidence weight: 0.143308192215966
## Worst IC: 611.641624123008
## 4 models within 2 IC units.
## 69 models to reach 95% of evidence weight.

coef(diag_rec.asreg.result)

##        Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.6   -0.04198          0.03415        20    0.08642          0.3666
## mg.5    0.09014          0.16918        23    0.08991          0.8160
## mg.8    0.05482          0.09077        22    0.09363          0.5977
## mg.4   -0.32817          0.74775        25    0.10948          1.7156
## mg.3    0.31218          0.46168        23    0.13937          1.3480
## mg.7   -2.01919         12.22807        34    0.26043          6.9376
## mg.1    2.49155          6.77554        54    0.56766          5.1642
## mg.2    4.11725         10.05108        62    0.71812          6.2898
## mg.10   5.37163          7.23315        72    0.87203          5.3358
## mg.9    8.19185          5.18284        98    0.99664          4.5166
```

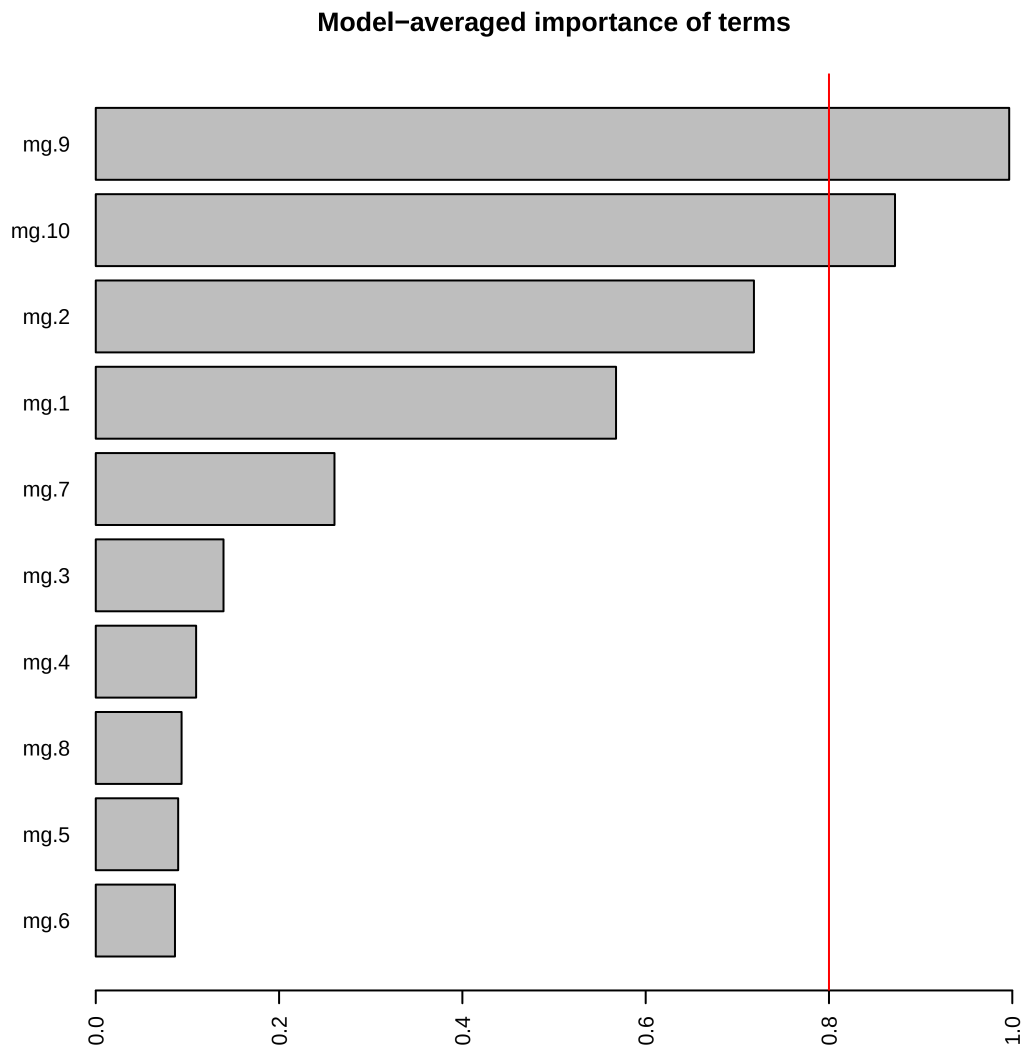
```r
summary(diag_rec.asreg.result@objects[[1]])

## Call:
## fitfunc(formula = as.formula(x), data = data)
##
##   n= 104, number of events= 77
##
##          coef exp(coef) se(coef)    z Pr(>|z|)
## mg.1     4.23     68.66     1.77 2.39   0.0170
## mg.2     5.67    290.86     2.16 2.63   0.0086
## mg.9     8.44   4647.50     2.06 4.11    4e-05
## mg.10    5.96    385.91     2.14 2.78   0.0054
##
##       exp(coef) exp(-coef) lower .95 upper .95
## mg.1       68.7   0.014564      2.13      2214
## mg.2      290.9   0.003438      4.22     20034
## mg.9     4647.5   0.000215     82.67    261263
## mg.10     385.9   0.002591      5.83     25560
##
## Concordance= 0.682  (se = 0.036 )
## Rsquare= 0.26   (max possible= 0.997 )
## Likelihood ratio test= 31.3  on 4 df,   p=2.68e-06
## Wald test            = 30.7  on 4 df,   p=3.59e-06
## Score (logrank) test = 32  on 4 df,   p=1.88e-06
```

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

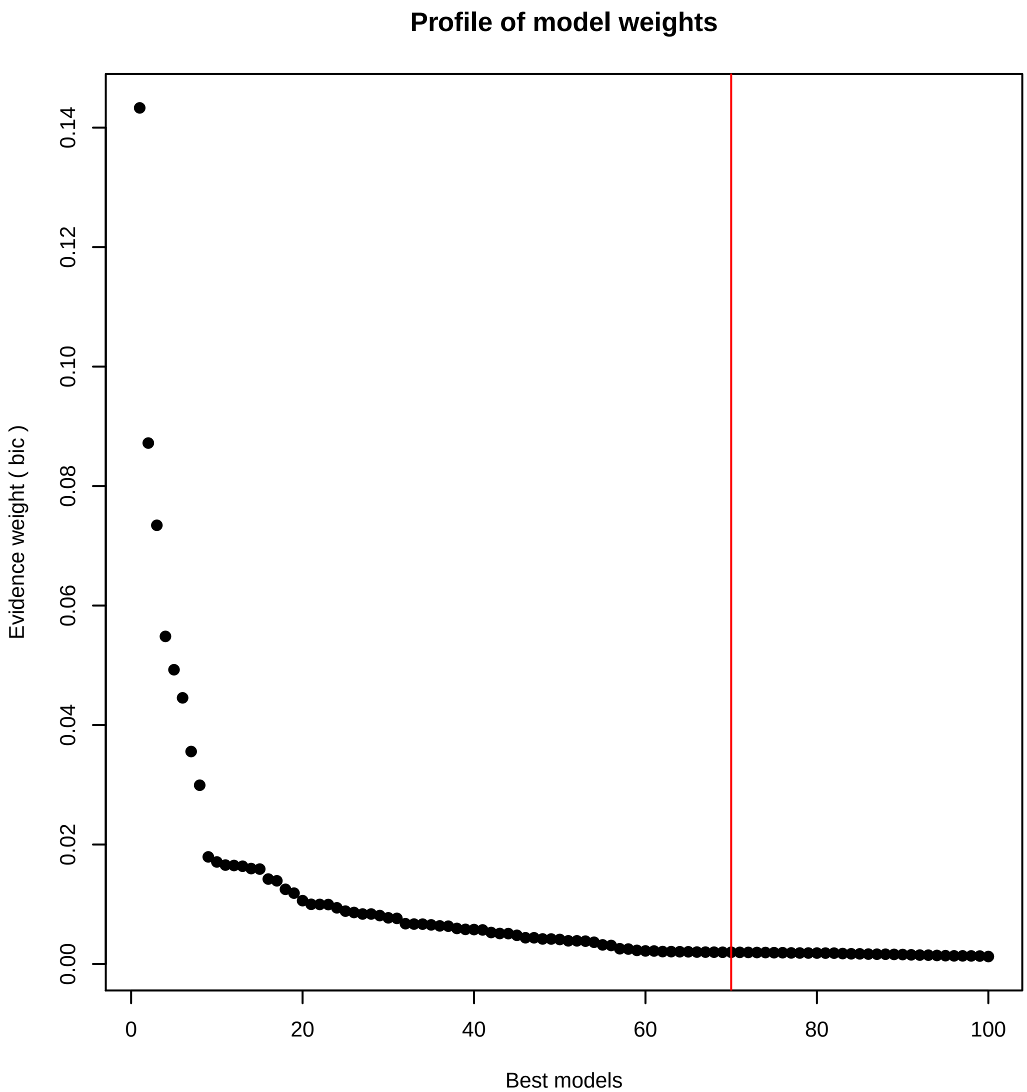
All-subsets regression




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




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




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

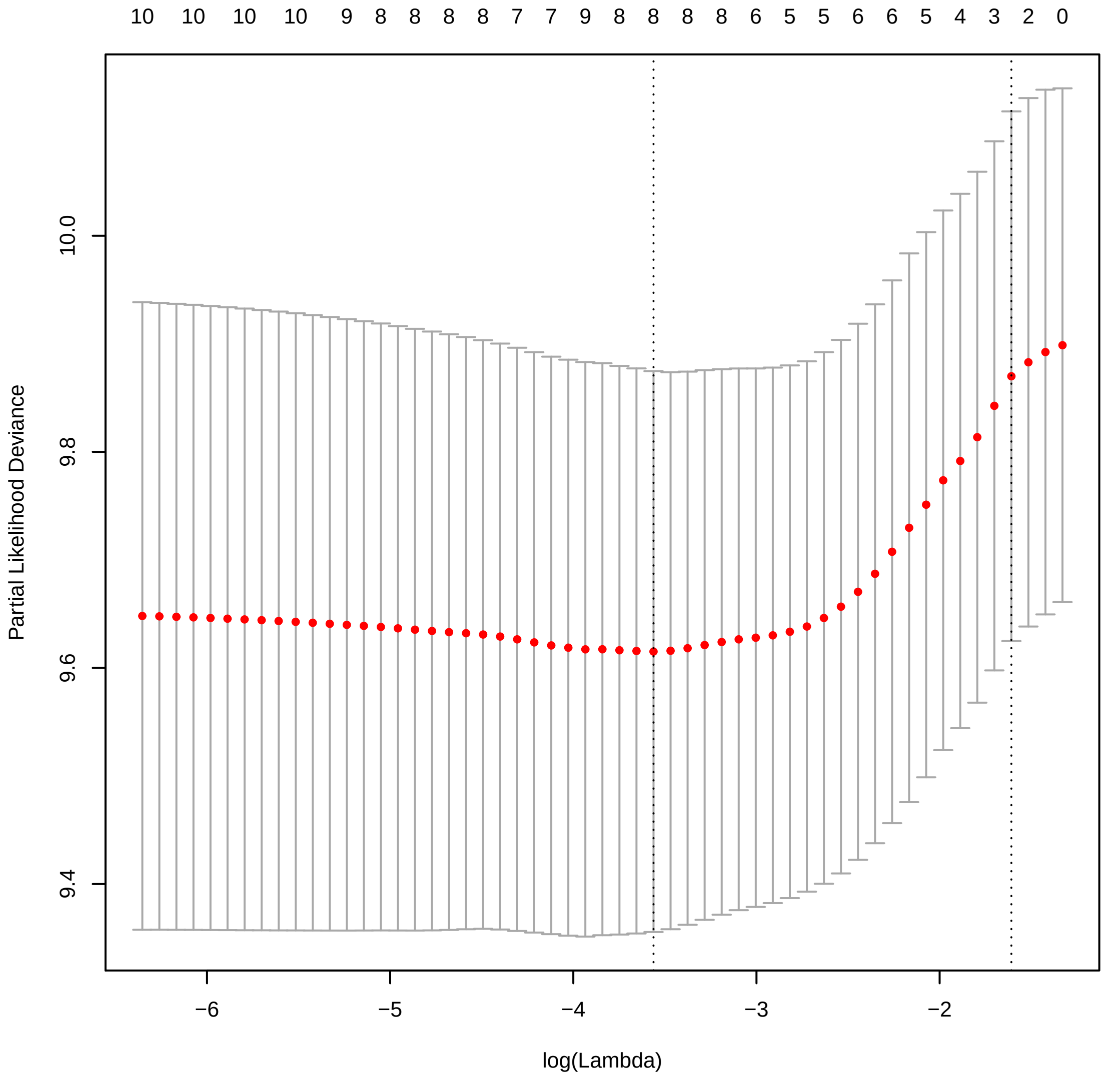
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  0.4796
## mg.2  .
## mg.3  .
## mg.4  .
## mg.5  .
## mg.6  .
## mg.7  .
## mg.8  .
## mg.9  1.8841
## mg.10 0.3015
```

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

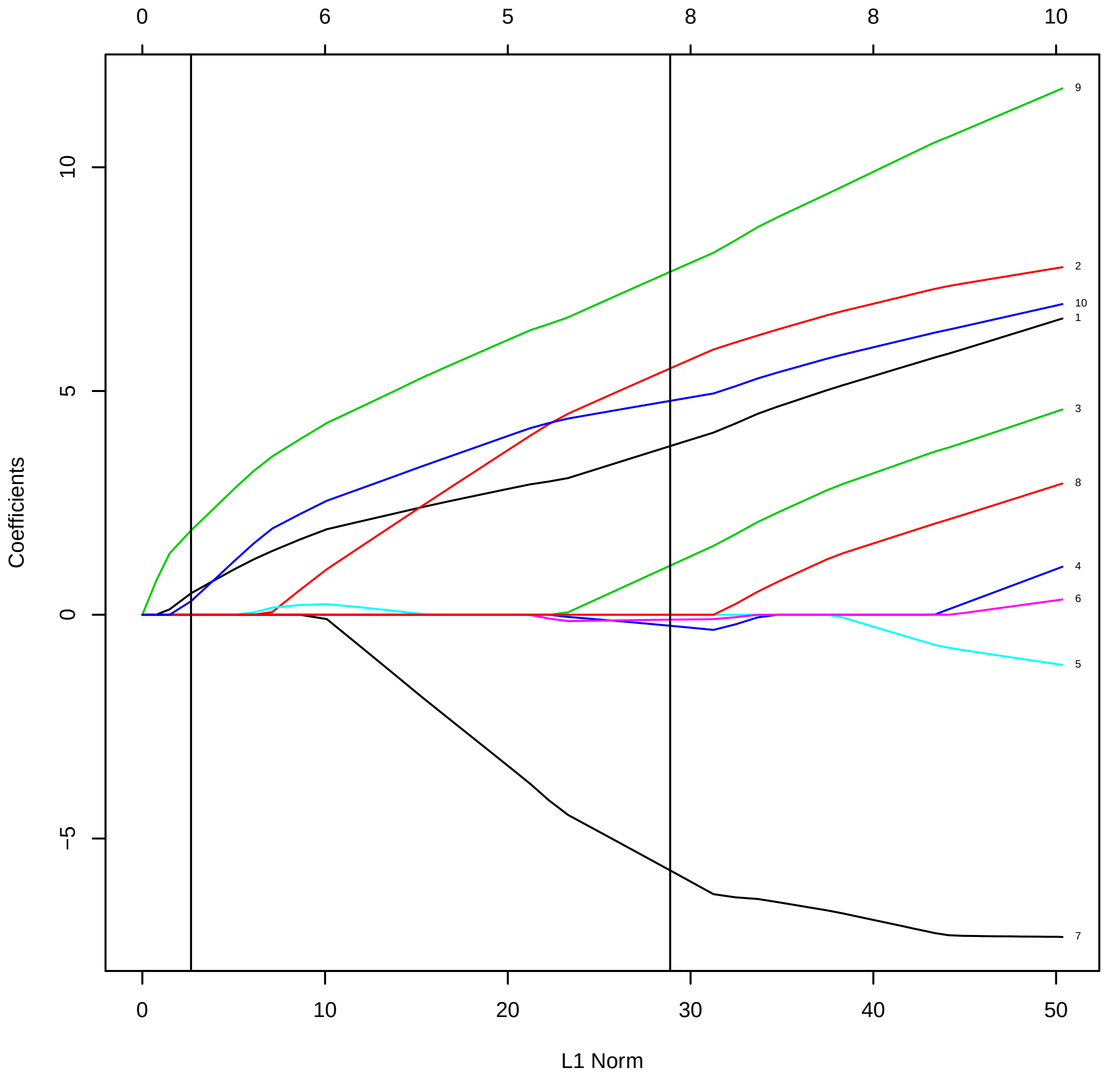
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1   3.7687
## mg.2   5.5044
## mg.3   1.0976
## mg.4  -0.2457
## mg.5   .
## mg.6  -0.1094
## mg.7  -5.7123
## mg.8   .
## mg.9   7.6634
## mg.10  4.7798
```

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

LASSO




```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)))
```



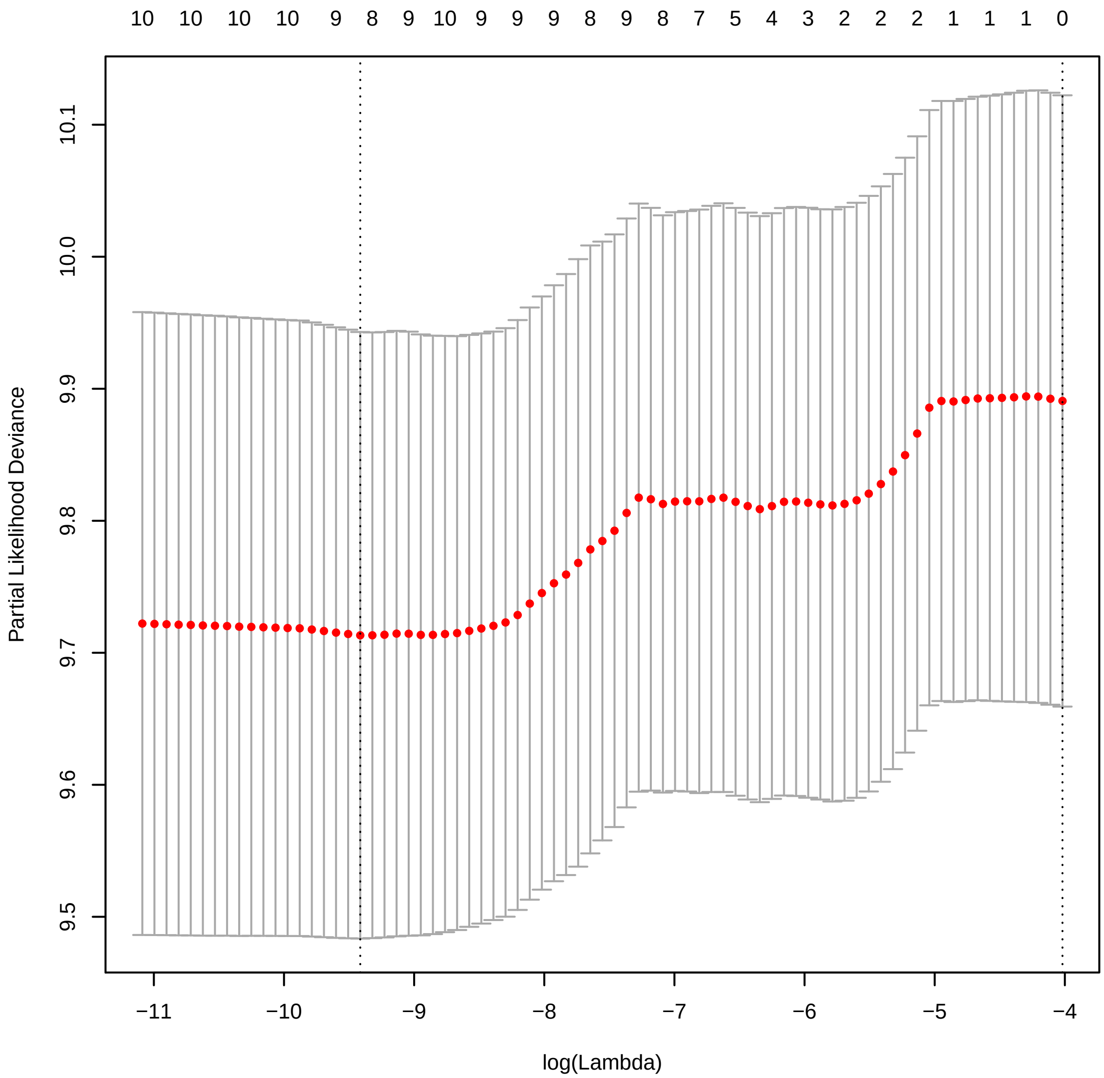

```
diag_rec.adaglmnet.coef.1se/diag_rec.adaglmnet.weights

## 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
```

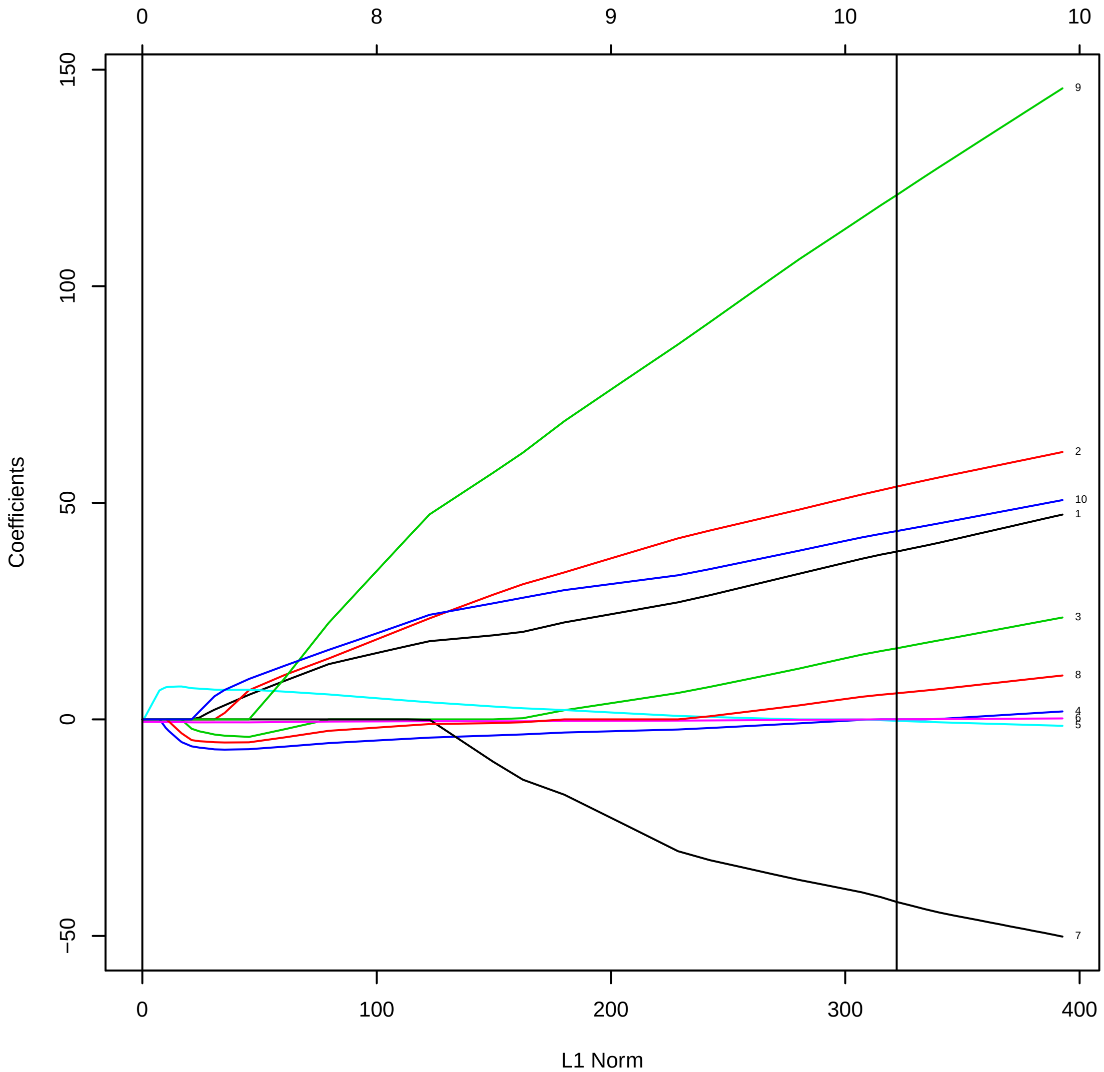
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                1
## mg.1   2.734e+02
## mg.2   4.292e+02
## mg.3   8.279e+01
## mg.4   .
## mg.5  -4.366e-01
## mg.6   4.932e-03
## mg.7  -3.039e+02
## mg.8   2.019e+01
## mg.9   1.493e+03
## mg.10  3.153e+02
```

```
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)))
```



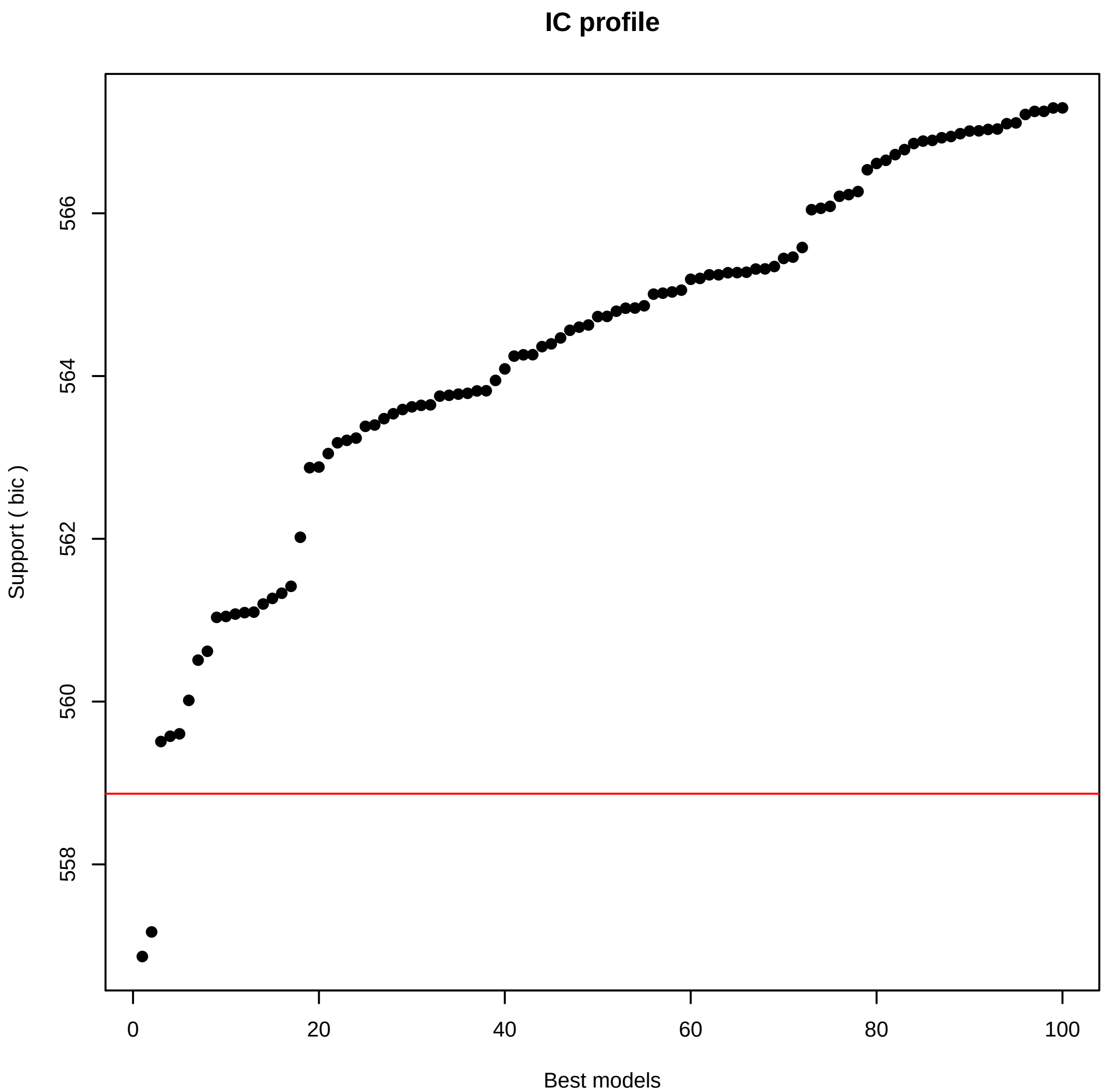
4.5.2 Outcome: Diagnosis to disease-specific death

```
print(diag_dsd.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 100 models:
## Best IC: 556.867702288718
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.1 + mg.9"
## Evidence weight: 0.170536282369992
## Worst IC: 567.295023763913
## 2 models within 2 IC units.
## 65 models to reach 95% of evidence weight.
```

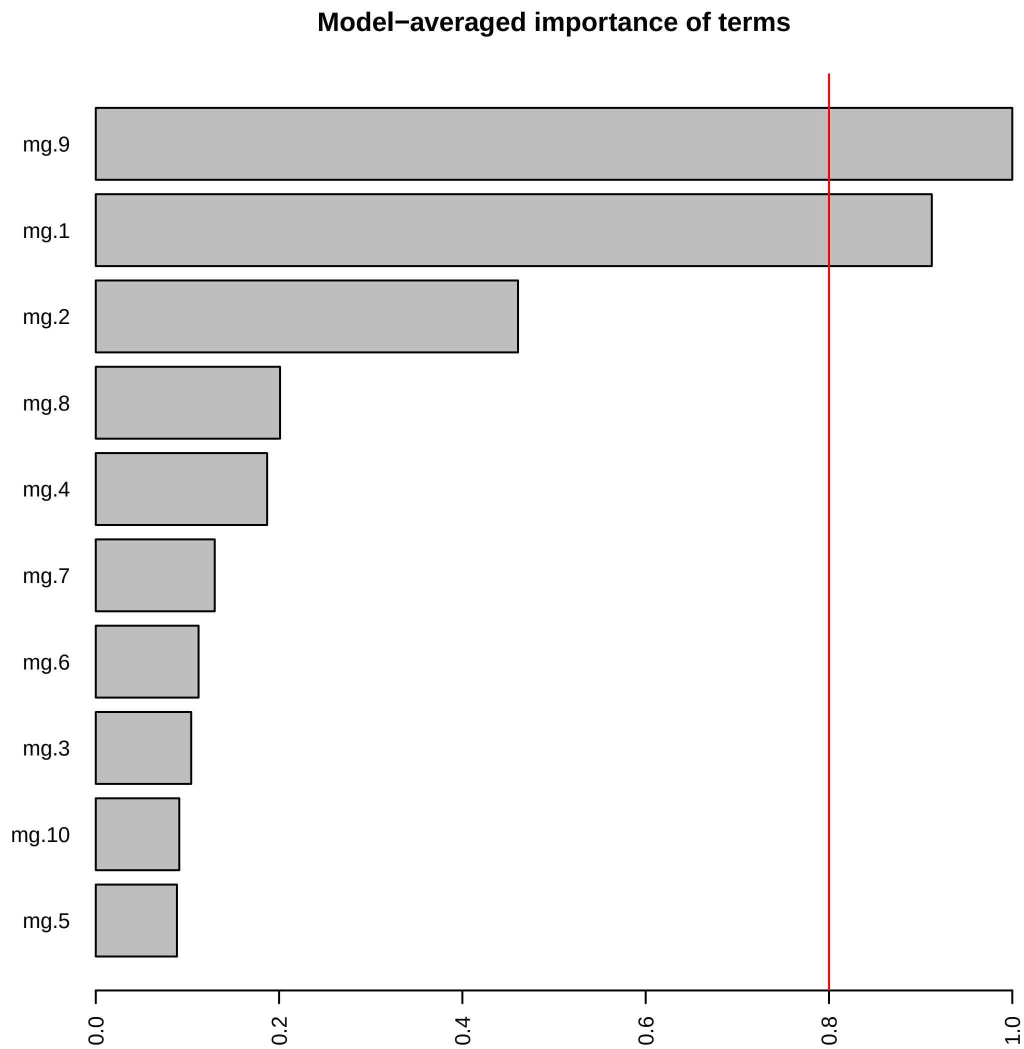
```r
coef(diag_dsd.asreg.result)

##        Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.5  -0.008612          0.07855        20    0.08856           0.5556
## mg.10  0.044179          0.05892        21    0.09113           0.4812
## mg.3   0.049442          0.09597        24    0.10413           0.6142
## mg.6   0.123275          0.10152        24    0.11216           0.6316
## mg.7  -0.554651          1.86670        30    0.12979           2.7086
## mg.4   0.969869          3.53822        30    0.18697           3.7290
## mg.8  -0.636590          1.49661        37    0.20105           2.4252
## mg.2   2.122754          7.28306        48    0.46073           5.3501
## mg.1   5.137280          4.80906        77    0.91217           4.3474
## mg.9   8.989159          4.49096       100    1.00000           4.2012

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    5.32    203.54     1.71 3.12   0.0018
## mg.9    8.60   5415.06     1.91 4.51  6.5e-06
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.1       204   0.004913       7.2      5755
## mg.9      5415   0.000185     129.0    227339
##
## Concordance= 0.691  (se = 0.038 )
## Rsquare= 0.229   (max possible= 0.995 )
## Likelihood ratio test= 28.6  on 2 df,   p=6.11e-07
## Wald test            = 31.5  on 2 df,   p=1.45e-07
## Score (logrank) test = 33.5  on 2 df,   p=5.17e-08
```

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

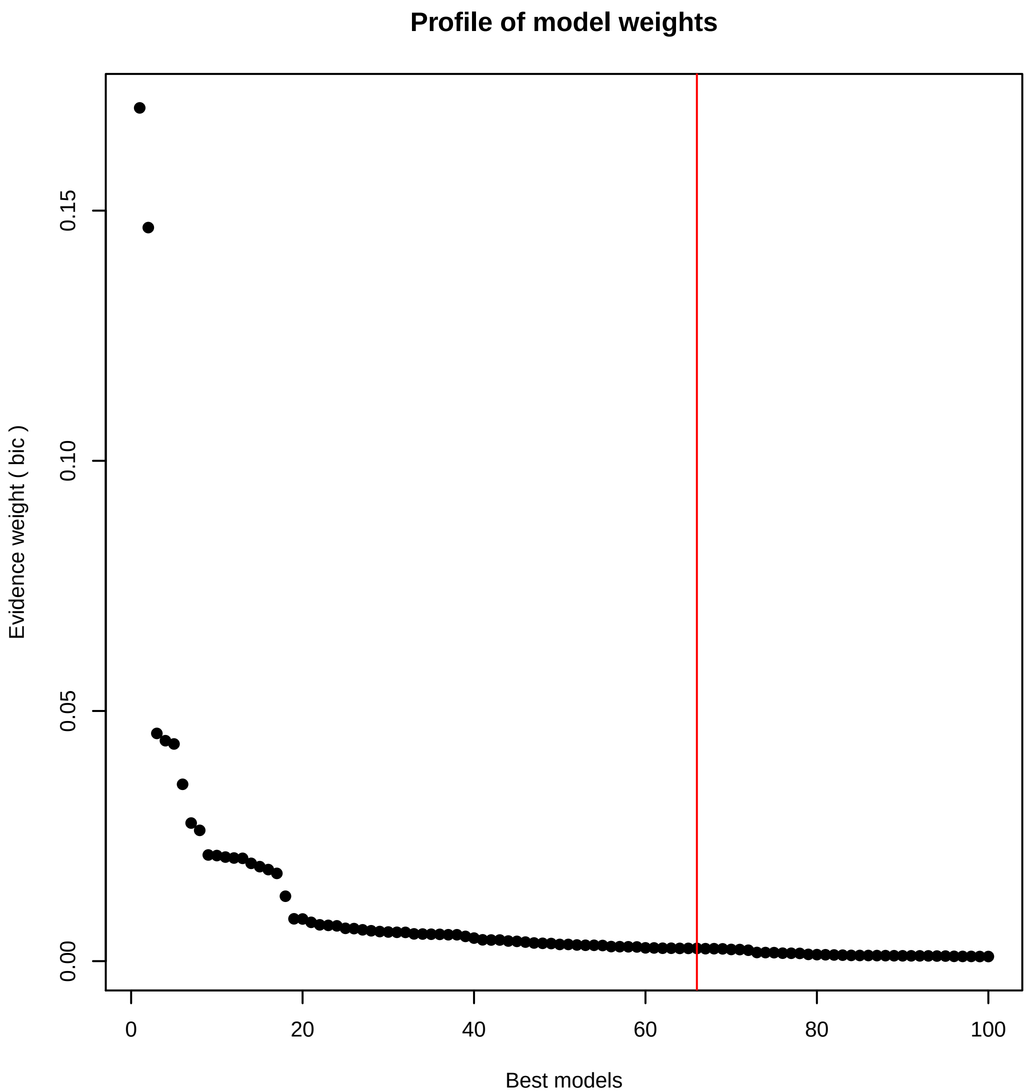
All-subsets regression




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




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



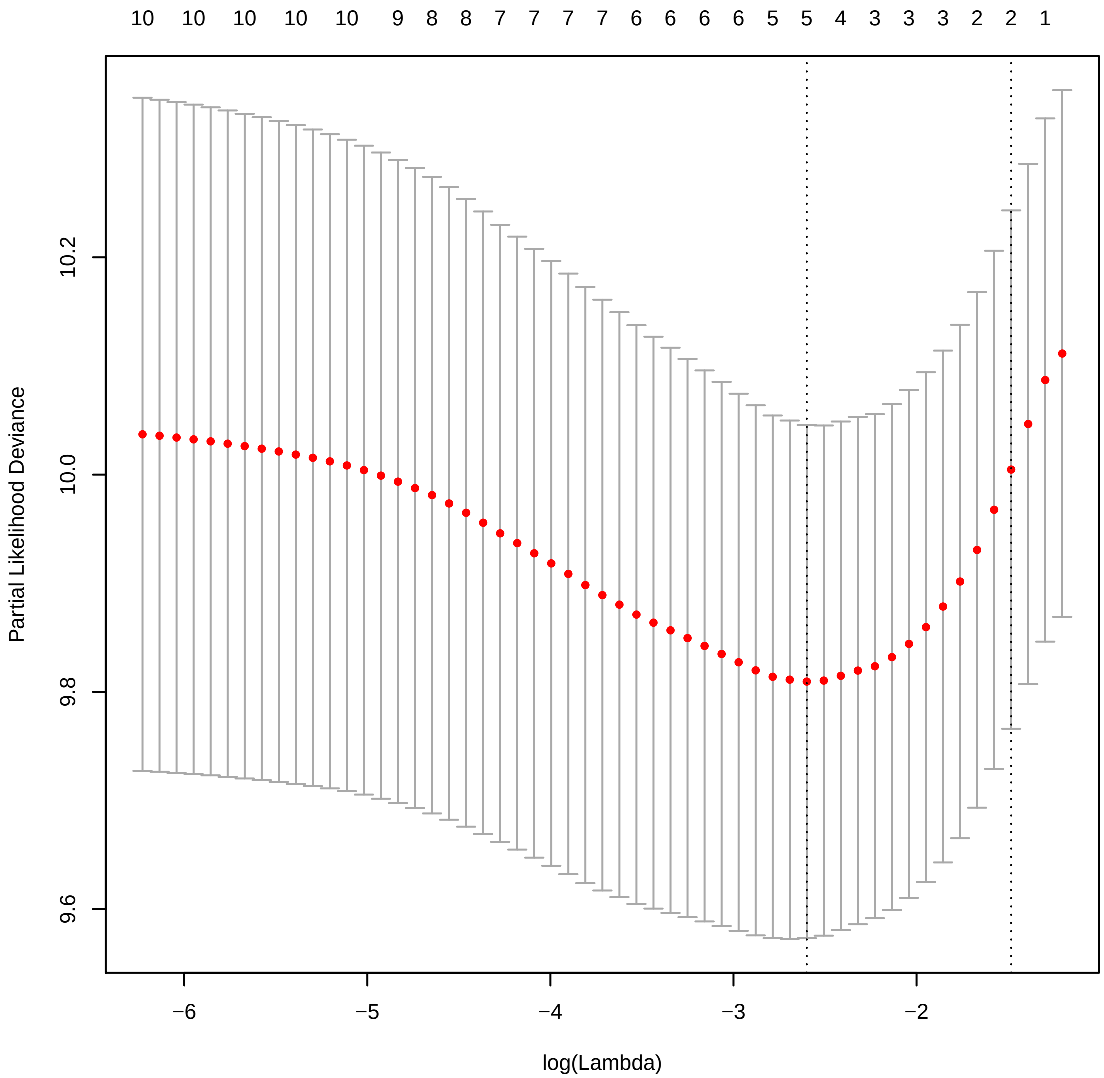

```
diag_dsd.glmnet.coef.1se

## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1  0.3598
## mg.2  .
## mg.3  .
## mg.4  .
## mg.5  .
## mg.6  .
## mg.7  .
## mg.8  .
## mg.9  2.7511
## mg.10 .
```

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

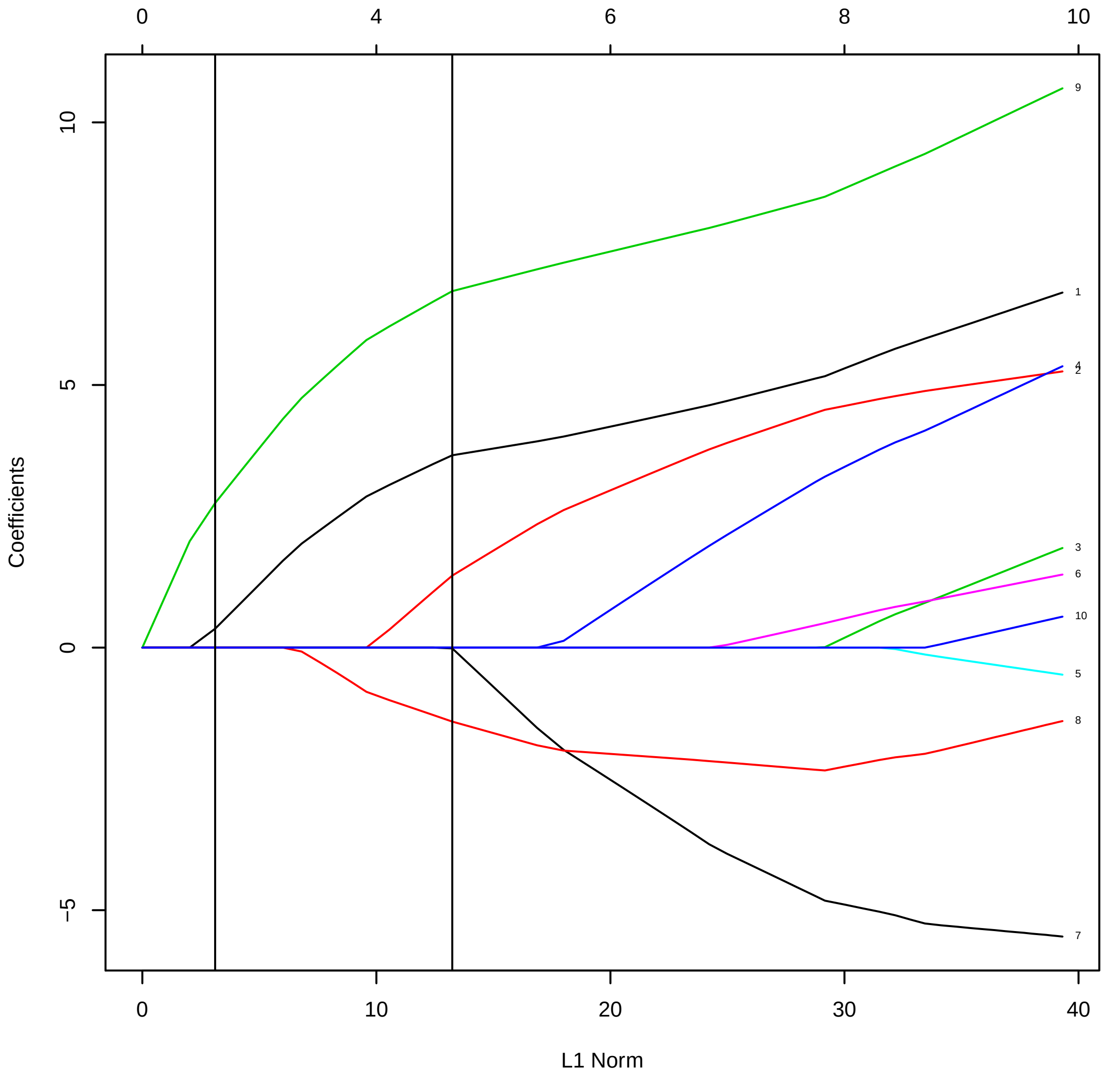
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##              1
## mg.1   3.66155
## mg.2   1.37061
## mg.3   .
## mg.4   .
## mg.5   .
## mg.6   .
## mg.7  -0.01541
## mg.8  -1.40923
## mg.9   6.78666
## mg.10  .
```

```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)))
```



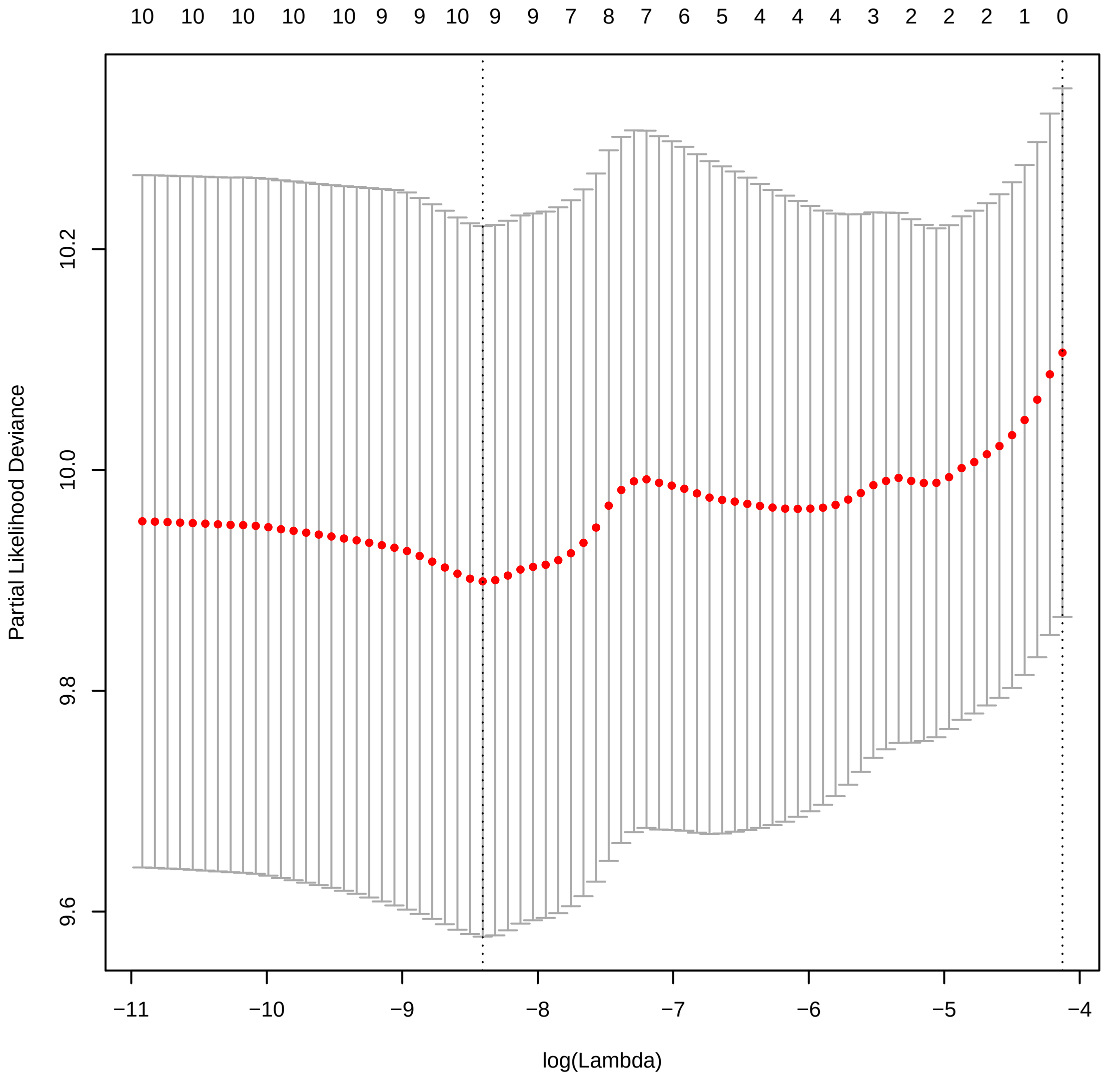

```
diag_dsd.adaglmnet.coef.1se/diag_dsd.adaglmnet.weights

## 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  .
```

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

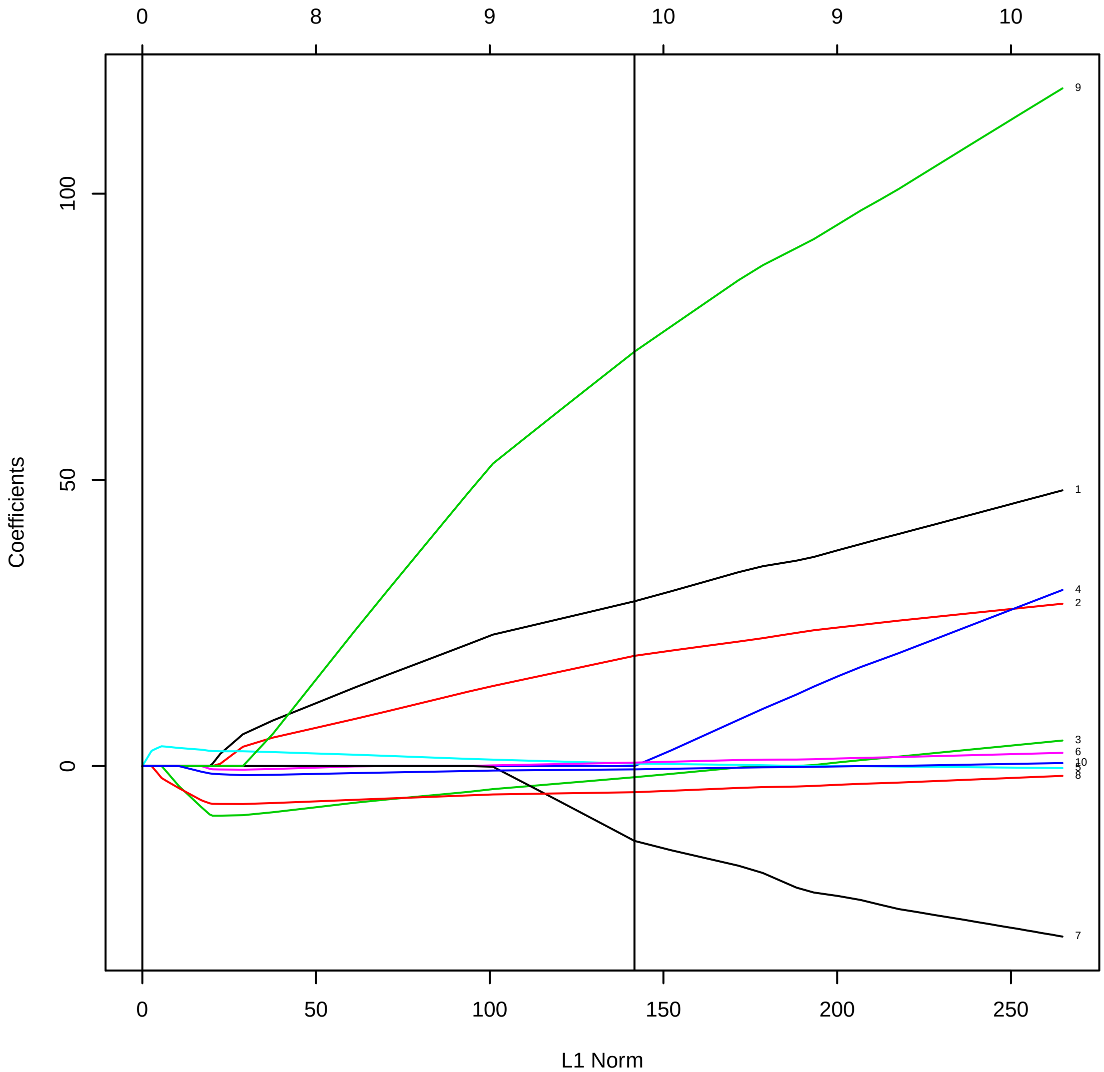
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                1
## mg.1  204.8068
## mg.2  104.2665
## mg.3   -4.5232
## mg.4     .
## mg.5    0.2736
## mg.6    0.9583
## mg.7  -72.9761
## mg.8   -5.3641
## mg.9  804.8149
## mg.10  -0.4809
```

```r
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)))
```



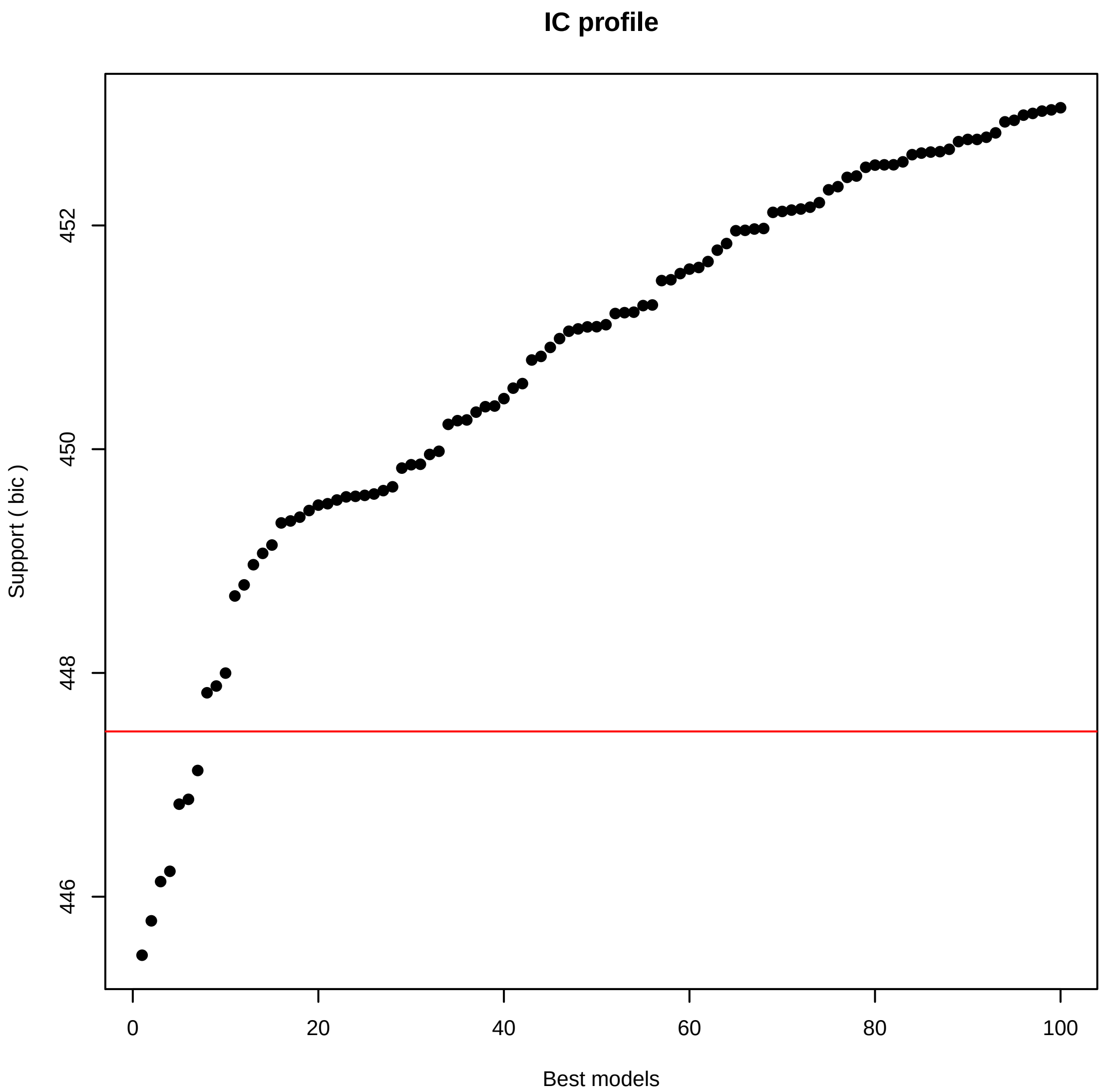
4.5.3 Outcome: Recurrence to disease-specific death

```
print(recr_dsd.asreg.result)

## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 1 / Marginality: TRUE
## From 100 models:
## Best IC: 445.476980260502
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.1 + mg.3 + mg.4 + mg.6 + mg.9"
## Evidence weight: 0.0844730316616126
## Worst IC: 453.052069476891
## 7 models within 2 IC units.
## 77 models to reach 95% of evidence weight.
```

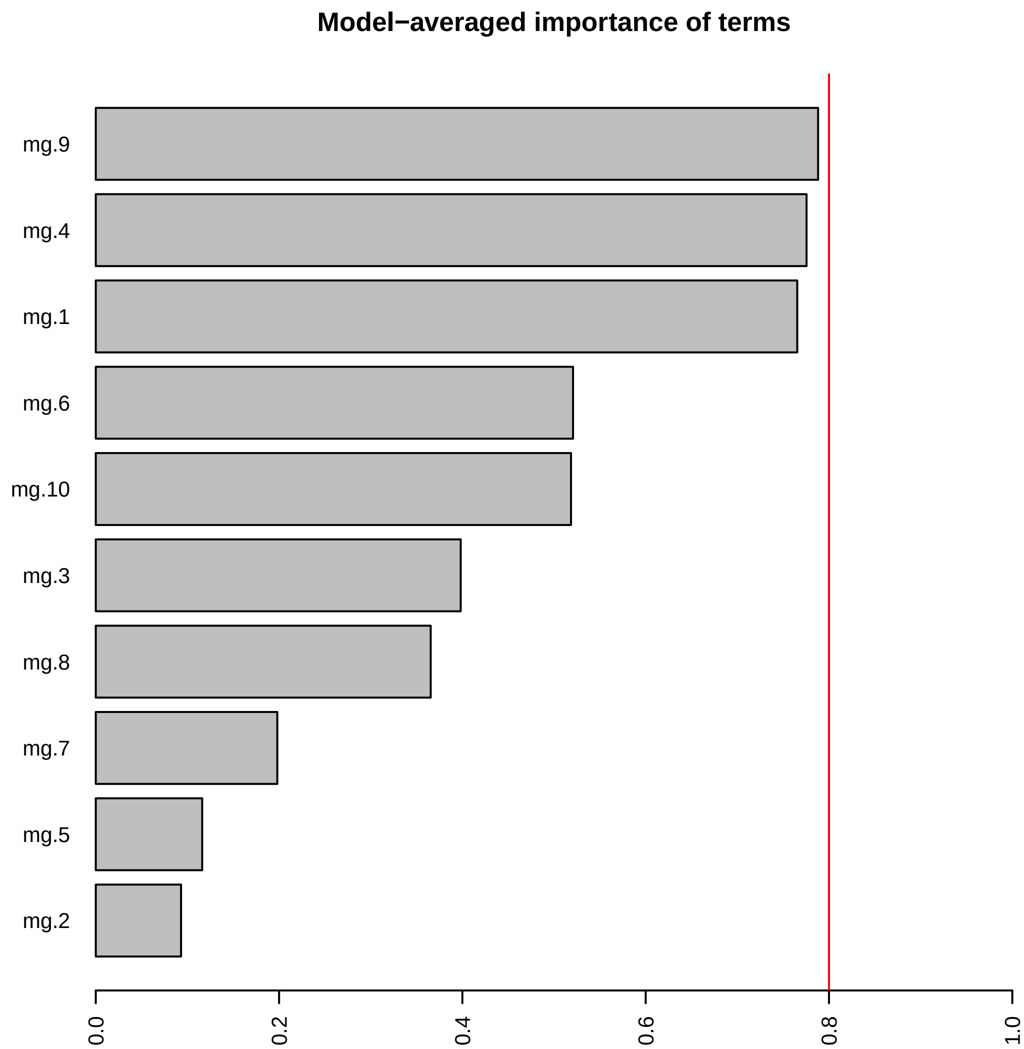
```r
coef(recr_dsd.asreg.result)

##       Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
## mg.2   -0.1664           0.1938        17    0.09306           0.8766
## mg.5   -0.3148           0.5772        24    0.11613           1.5130
## mg.7    2.0062          13.7165        23    0.19807           7.3756
## mg.8   -2.8070          16.5987        53    0.36545           8.1136
## mg.3    0.2056           6.1002        37    0.39820           4.9187
## mg.10  -3.1715          13.4338        62    0.51854           7.2992
## mg.6    2.4337           7.6675        48    0.52075           5.5144
## mg.1    5.3336          12.8263        71    0.76538           7.1322
## mg.4   12.3462          61.2298        75    0.77551          15.5832
## mg.9    6.4599          16.6082        75    0.78812           8.1159

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 9.65e+00  1.56e+04 2.45e+00 3.94  8.2e-05
## mg.3 6.30e+00  5.45e+02 2.83e+00 2.22   0.0261
## mg.4 1.94e+01  2.72e+08 4.68e+00 4.15  3.4e-05
## mg.6 6.63e+00  7.57e+02 2.19e+00 3.03   0.0024
## mg.9 1.13e+01  7.94e+04 2.63e+00 4.29  1.8e-05
##
##      exp(coef) exp(-coef) lower .95 upper .95
## mg.1  1.56e+04   6.41e-05    127.74  1.90e+06
## mg.3  5.45e+02   1.83e-03      2.11  1.41e+05
## mg.4  2.72e+08   3.68e-09  28060.66  2.63e+12
## mg.6  7.57e+02   1.32e-03     10.46  5.49e+04
## mg.9  7.94e+04   1.26e-05    459.87  1.37e+07
##
## Concordance= 0.726  (se = 0.041 )
## Rsquare= 0.338   (max possible= 0.997 )
## Likelihood ratio test= 33.4  on 5 df,   p=3.1e-06
## Wald test            = 34  on 5 df,   p=2.41e-06
## Score (logrank) test = 35.5  on 5 df,   p=1.19e-06
```

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

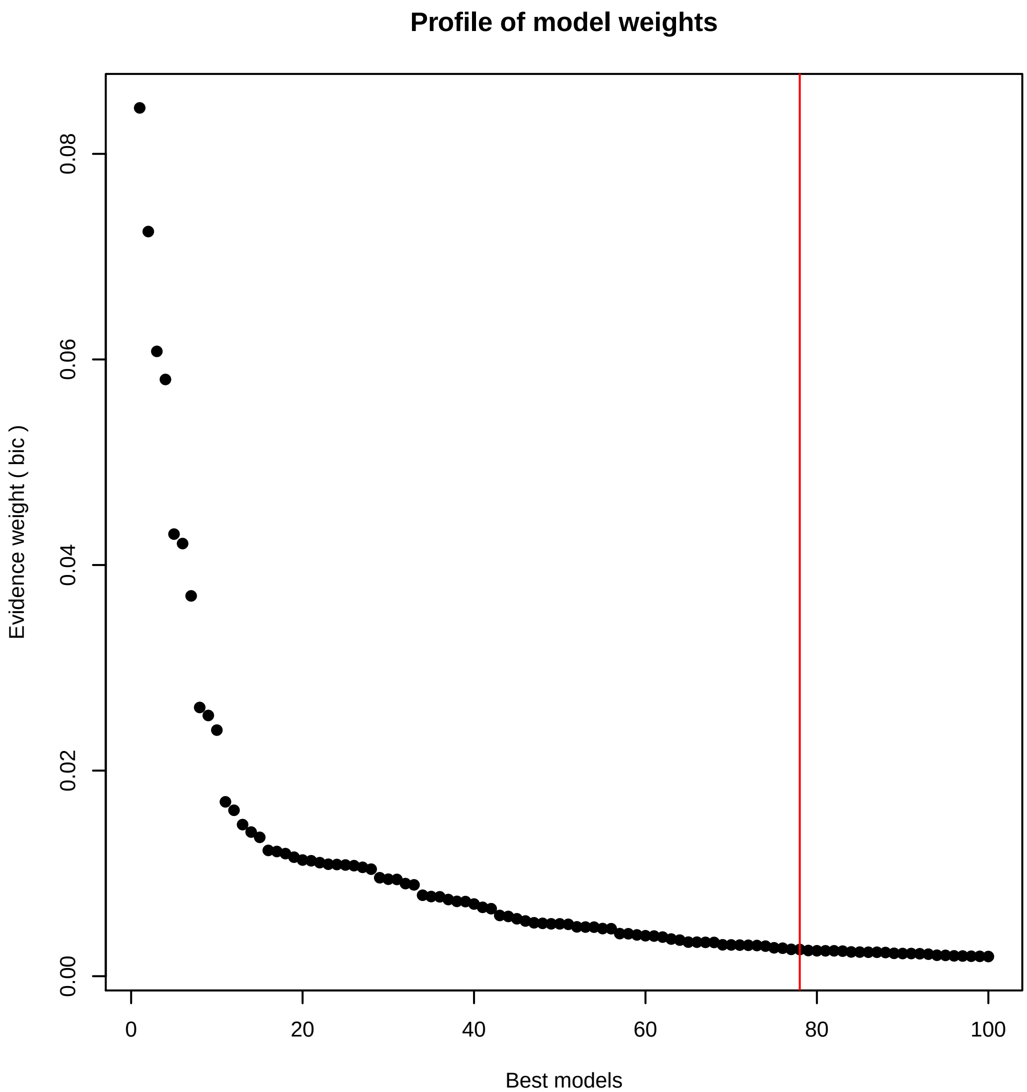
All-subsets regression




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




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



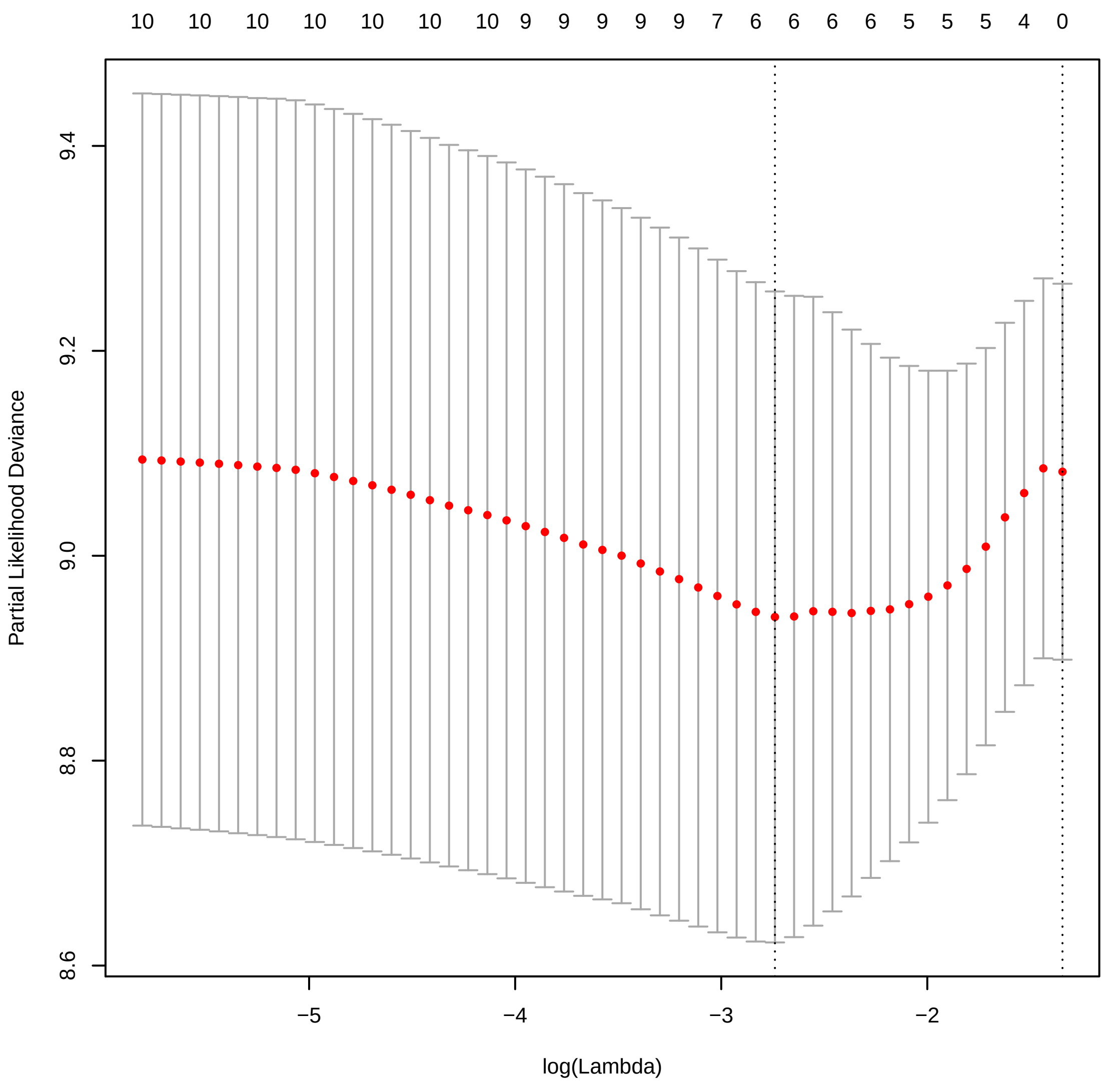

```
recr_dsd.glmnet.coef.1se

## 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  .
```

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

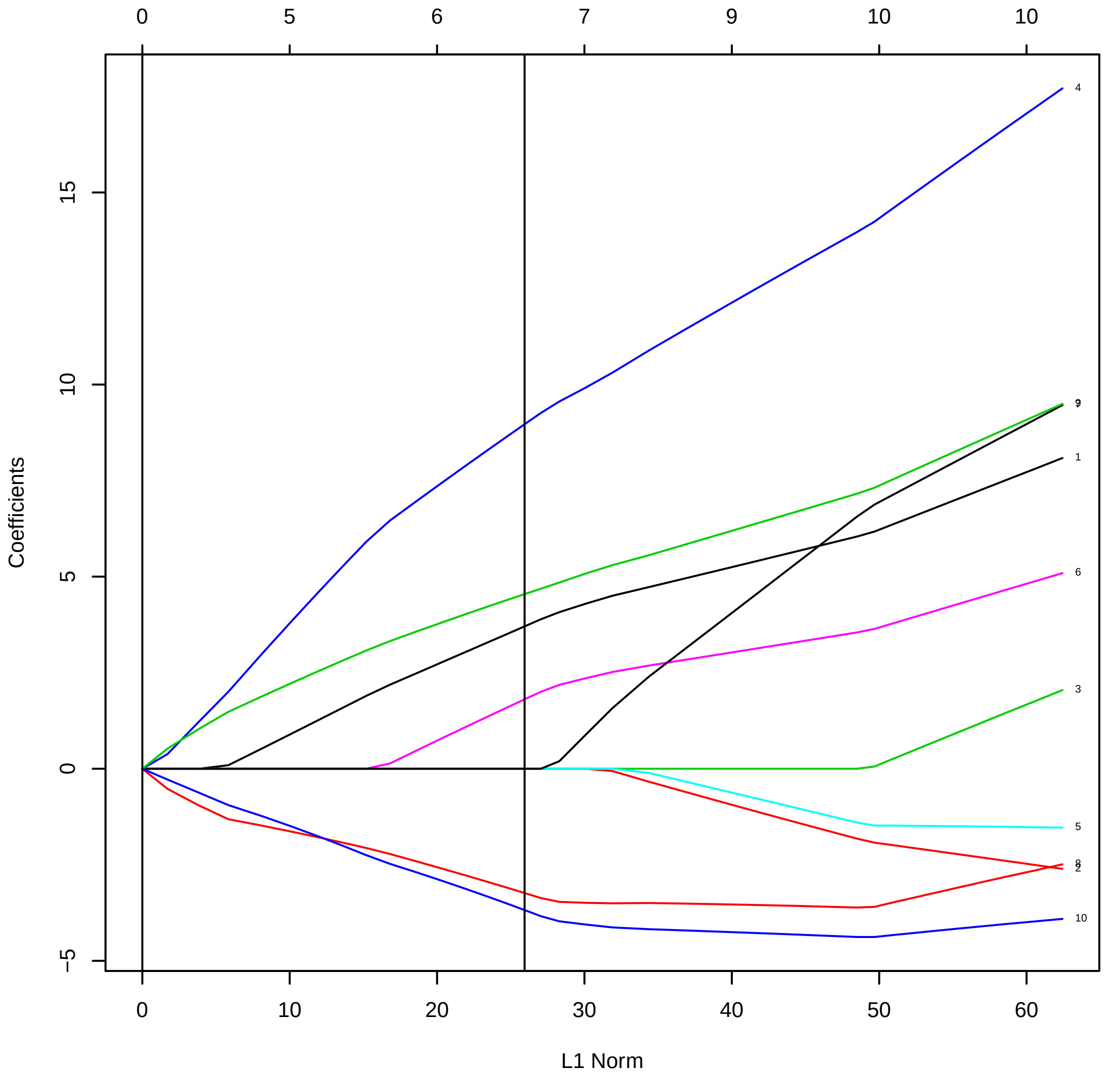
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1   3.705
## mg.2   .
## mg.3   .
## mg.4   8.967
## mg.5   .
## mg.6   1.806
## mg.7   .
## mg.8  -3.235
## mg.9   4.545
## mg.10 -3.678
```

```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)))
```



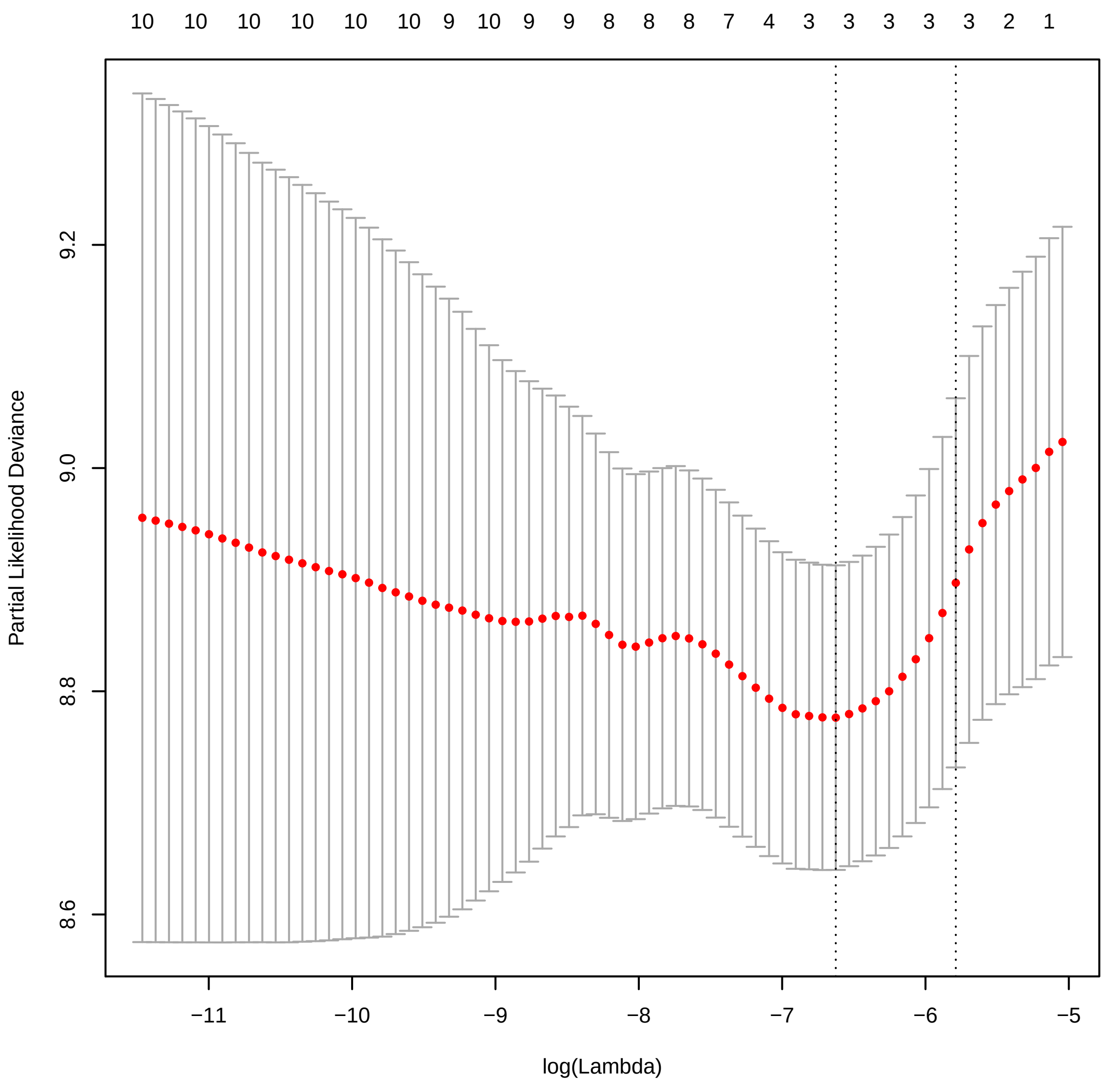

```
recr_dsd.adaglmnet.coef.1se/recr_dsd.adaglmnet.weights

## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1    .
## mg.2    .
## mg.3  -10.39
## mg.4    .
## mg.5    .
## mg.6    .
## mg.7    .
## mg.8  -21.56
## mg.9    .
## mg.10 -25.24
```

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

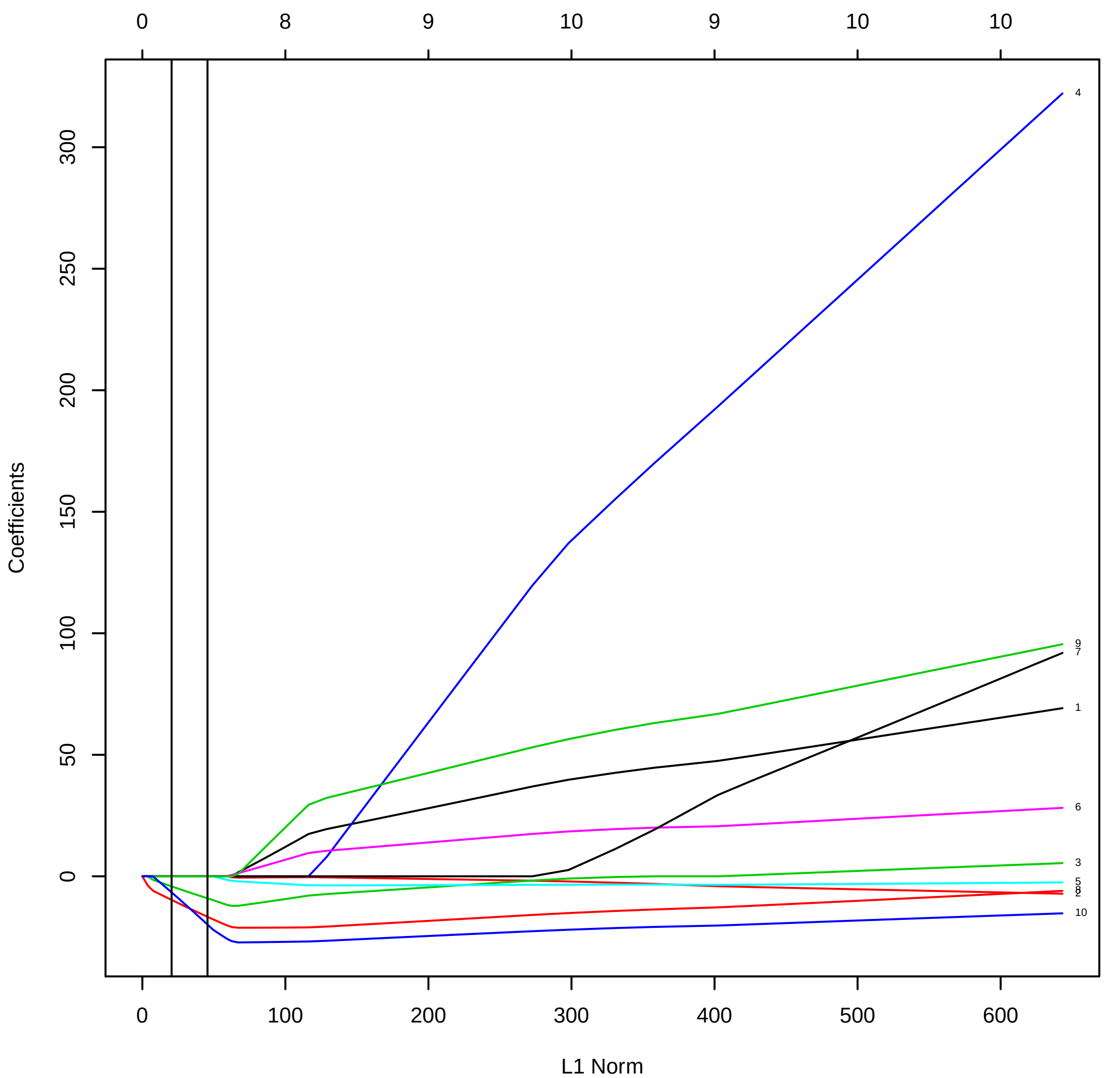
```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             1
## mg.1    .
## mg.2    .
## mg.3  -22.70
## mg.4    .
## mg.5    .
## mg.6    .
## mg.7    .
## mg.8  -37.01
## mg.9    .
## mg.10 -75.77
```

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

Adaptive LASSO




```r
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   stats     graphics  grDevices utils     datasets
## [8] methods   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
```