SIS NMF recr dsd

November 23, 2014

1 Preparation

```
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)
##
## Attaching package: 'qplots'
##
## 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
setwd(data_dir)
load("image.rda")
```

2 Probe selection

```
table(cpss.sis$sel)

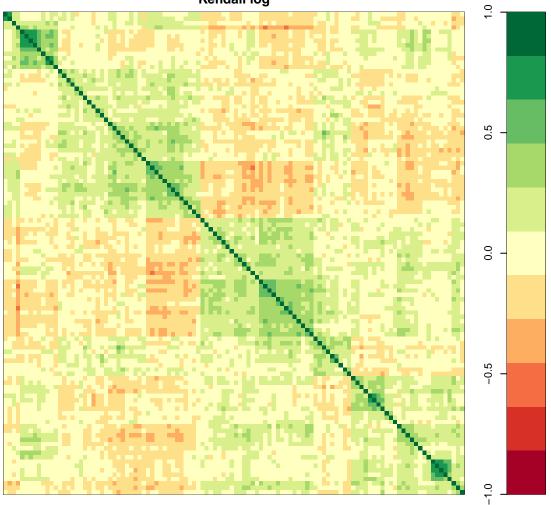
##
## FALSE TRUE
## 12890 110

mean(cpss.sis$sel)
## [1] 0.008462
```

3 Expression correlation

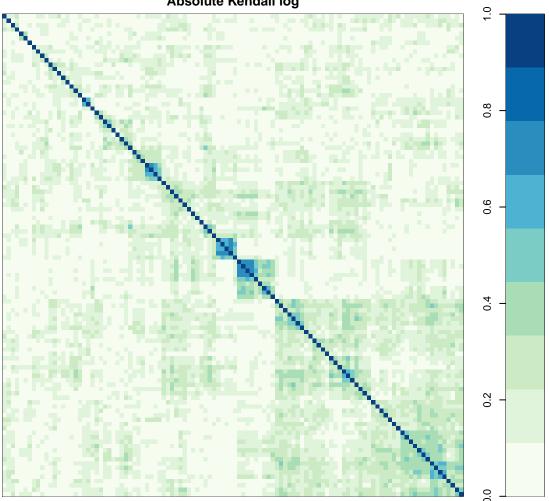
```
corPlot(x.sel.kcor, main = "Correlation Clusters of CPSS-SIS-FAST Probes\nKendall log",
    useRaster = FALSE)
```

Correlation Clusters of CPSS-SIS-FAST Probes Kendall log

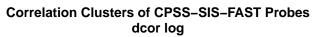


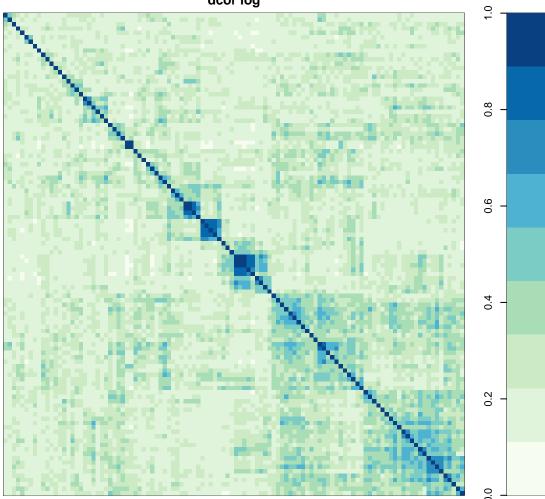
corPlot(abs(x.sel.kcor), zlim = c(0, 1), pal = "GnBu", main = "Correlation Clusters of CPSS-SIS-FAST Pro
 useRaster = FALSE)

Correlation Clusters of CPSS-SIS-FAST Probes Absolute Kendall log



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

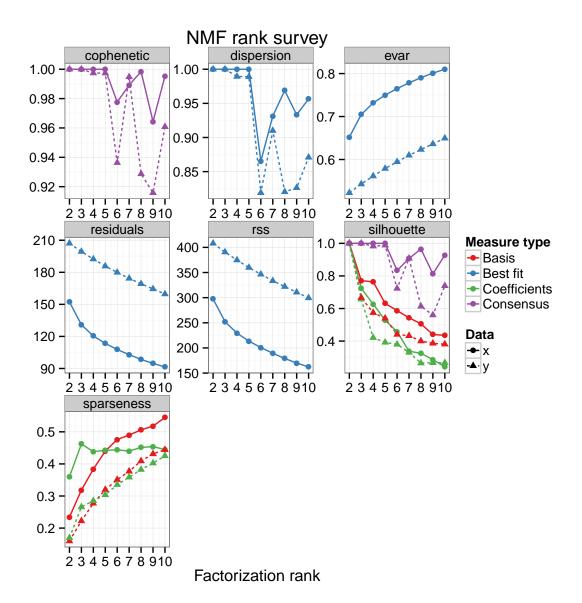




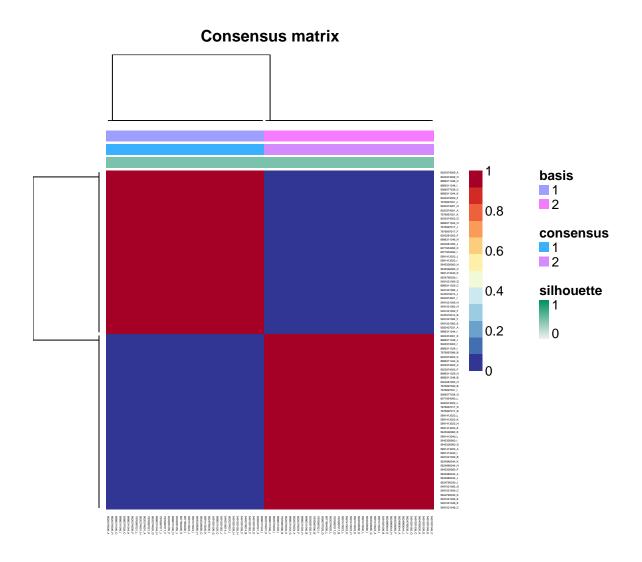
4 Factorization

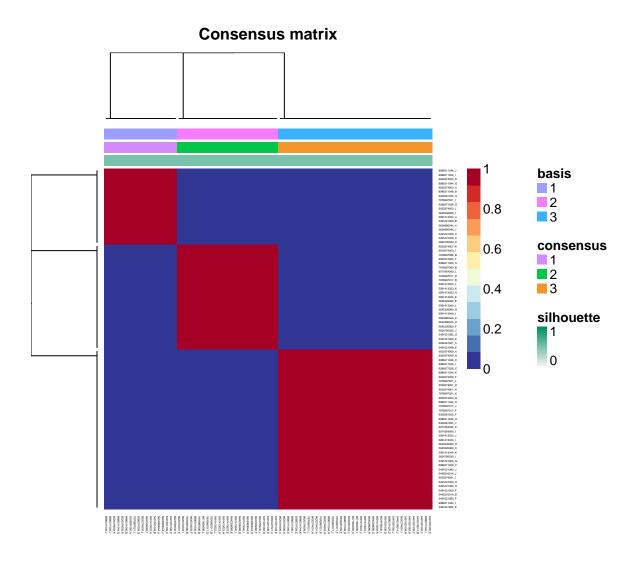
4.1 Rank estimation

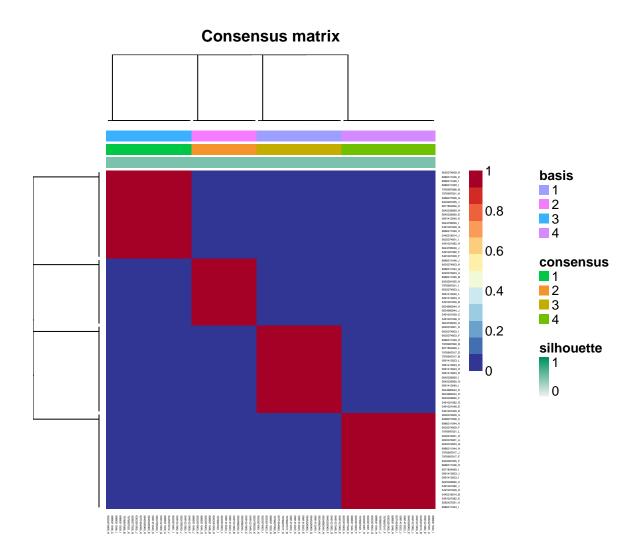
plot(temp.nmf.rank, temp.nmf.rank.random[[1]])

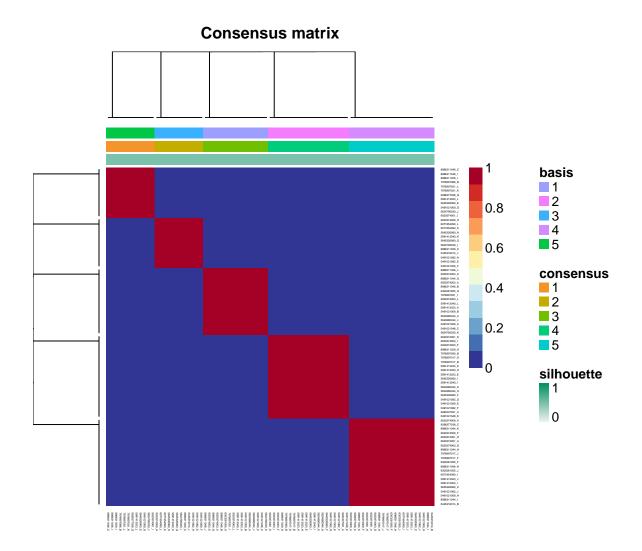


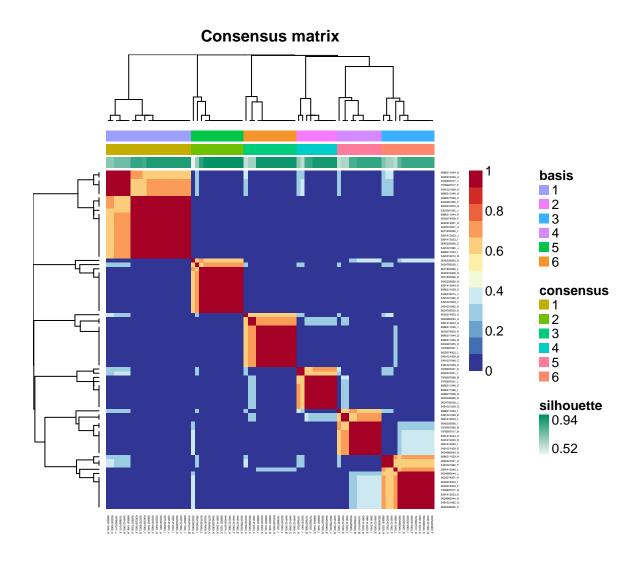
```
for (i in 1:length(temp.nmf.rank$fit)) {
    consensusmap(temp.nmf.rank$fit[[i]])
}
```

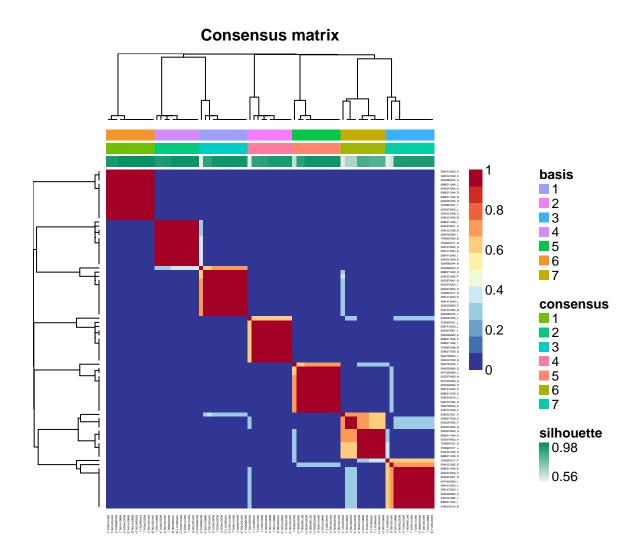


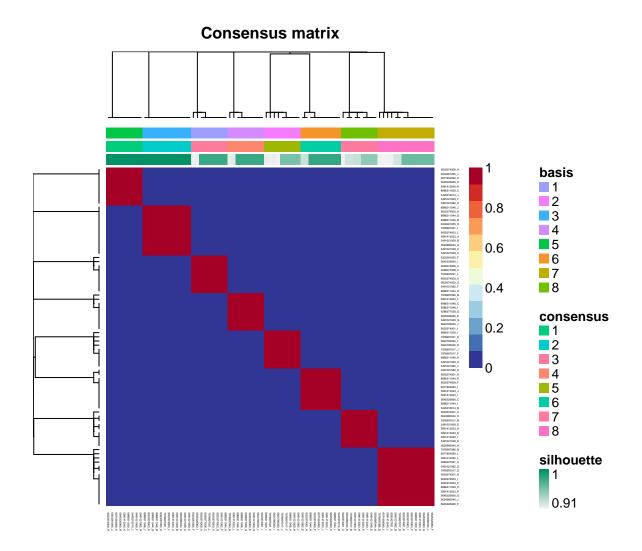


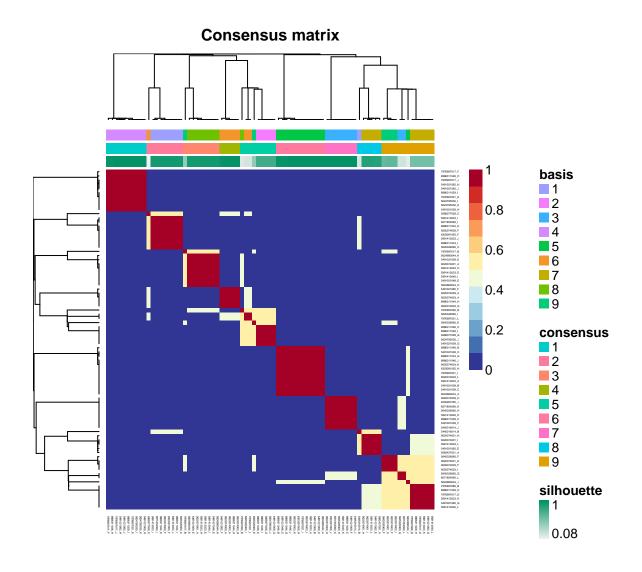


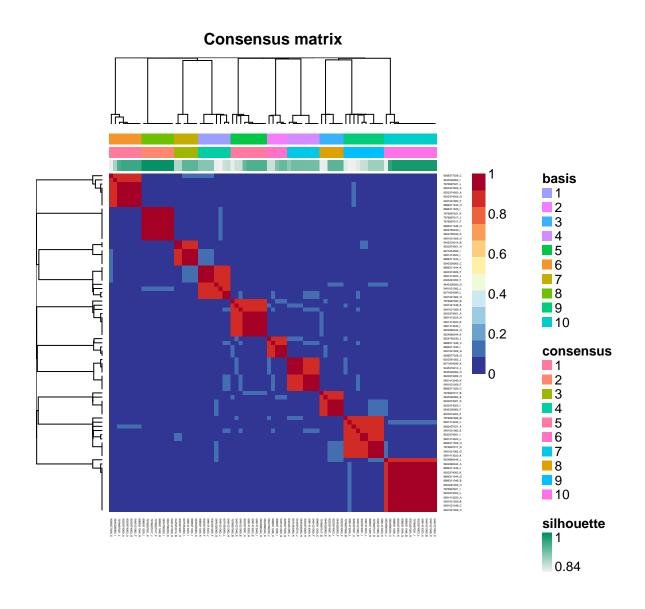






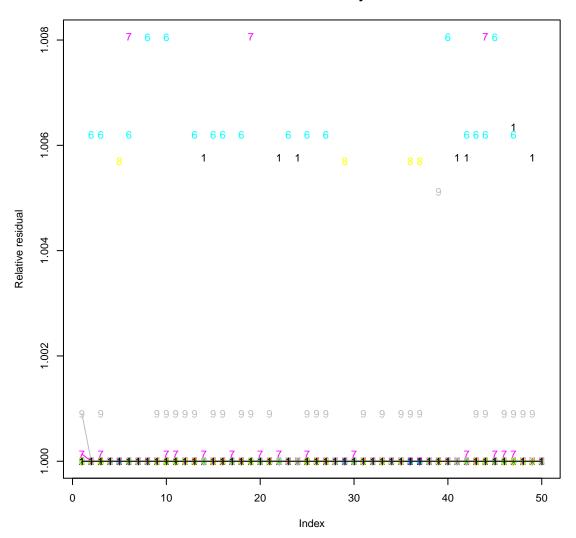






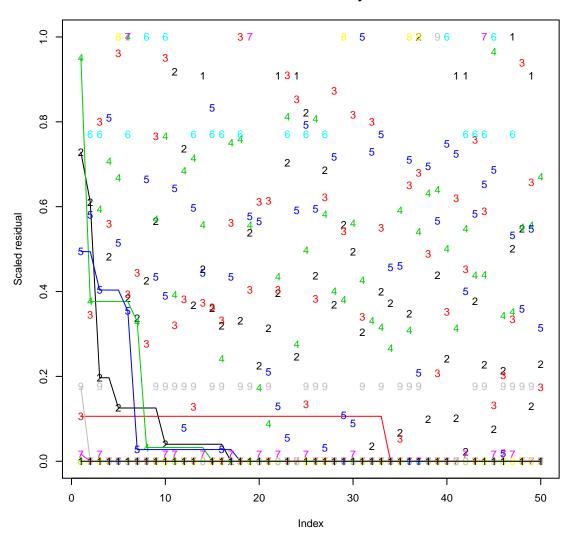
```
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_rel),
    ylab = "Relative residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_rel[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_rel[, i]), col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

Solution Stability

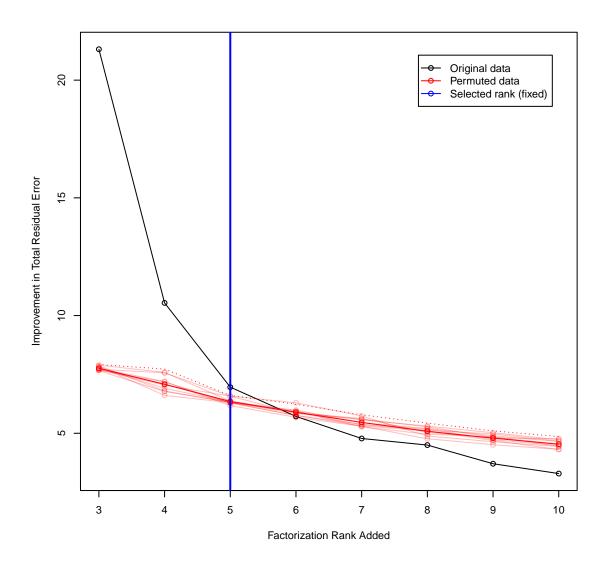


```
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_scaled),
    ylab = "Scaled residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_scaled[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_scaled[,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

Solution Stability

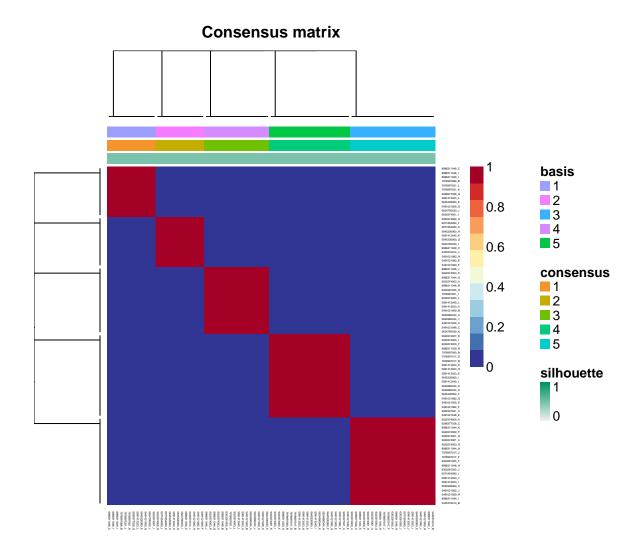


```
legend("topright", legend = c("Original data", "Permuted data", sprintf("Selected rank (%s)",
    ifelse(temp.col == "green", "auto", "fixed"))), col = c("black", "red",
    temp.col), lty = "solid", pch = 21, inset = 0.05)
```



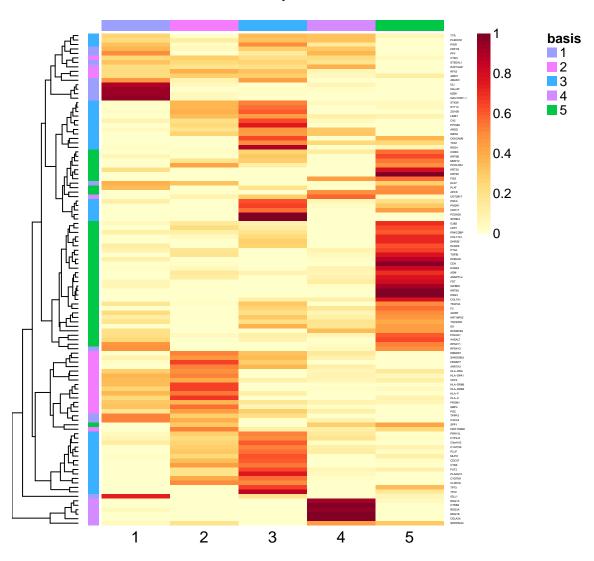
4.2 Fit

consensusmap(xlin.scaled.sel.nmf)

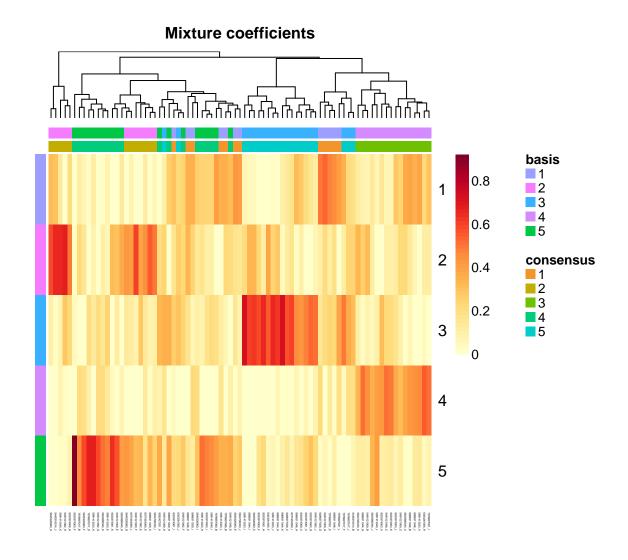


basismap(xlin.scaled.sel.nmf)

Basis components

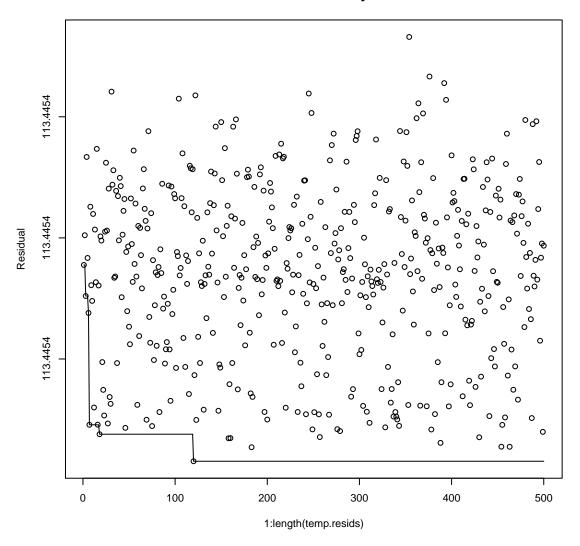


coefmap(xlin.scaled.sel.nmf)



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

Solution Stability



4.3 Component CPV associations

4.3.1 Survival: Diagnosis to disease-specific death

```
for (i in 1:nrow(coef(xlin.scaled.sel.nmf))) {
    print(summary(coxph(y ~ coef(xlin.scaled.sel.nmf)[i, ])))
## Call:
  coxph(formula = y ~ coef(xlin.scaled.sel.nmf)[i, ])
##
##
     n= 81, number of events= 64
##
##
                                                                  z Pr(>|z|)
##
                                      coef exp(coef) se(coef)
   coef(xlin.scaled.sel.nmf)[i, ] -4.8155
                                              0.0081
                                                       1.7671 -2.73
##
                                   exp(coef) exp(-coef) lower .95 upper .95
##
```

Table 1:

	Dependent variable:	
	У	
coef(xlin.scaled.sel.nmf)[i,]	-4.816***	
	(1.767)	
Observations	81	
\mathbb{R}^2	0.095	
Max. Possible \mathbb{R}^2	0.997	
Log Likelihood	-225.000	
Wald Test	$7.430^{***} (df = 1)$	
LR Test	$8.090^{***} (df = 1)$	
Score (Logrank) Test	$7.631^{***} (df = 1)$	
Vote	*n<0.1. **n<0.05. ***n	

Note:

*p<0.1; **p<0.05; ***p<0.01

Table 2:

	Dependent variable:
	у
$\overline{\operatorname{coef}(\operatorname{xlin.scaled.sel.nmf})[i,]}$	-6.078***
	(1.777)
Observations	81
\mathbb{R}^2	0.167
Max. Possible \mathbb{R}^2	0.997
Log Likelihood	-221.600
Wald Test	$11.710^{***} (df = 1)$
LR Test	$14.840^{***} (df = 1)$
Score (Logrank) Test	$12.230^{***} (df = 1)$
Note:	*p<0.1; **p<0.05; ***p<

Table 3:

	Dependent variable:
	y
coef(xlin.scaled.sel.nmf)[i,]	-1.093
	(1.398)
Observations	81
\mathbb{R}^2	0.008
Max. Possible \mathbb{R}^2	0.997
Log Likelihood	-228.700
Wald Test	0.610 (df = 1)
LR Test	0.632 (df = 1)
Score (Logrank) Test	0.612 (df = 1)
37 .	* 04 ** 00 ***

Note:

*p<0.1; **p<0.05; ***p<0.01

Table 4:

	Dependent variable.
	У
coef(xlin.scaled.sel.nmf)[i,]	0.709
	(1.349)
bservations	81
\mathbb{R}^2	0.003
Max. Possible R ²	0.997
log Likelihood	-228.900
Wald Test	0.280 (df = 1)
LR Test	0.269 (df = 1)
Score (Logrank) Test	0.277 (df = 1)
Note:	*** <0.1. *** <0.05. ****

Note:

*p<0.1; **p<0.05; ***p<0.01

Table 5:

_	Dependent variable:
	у
coef(xlin.scaled.sel.nmf)[i,]	7.827***
	(1.415)
Observations	81
\mathbb{R}^2	0.279
Max. Possible \mathbb{R}^2	0.997
Log Likelihood	-215.800
Wald Test	$30.590^{***} (df = 1)$
LR Test	$26.550^{***} (df = 1)$
Score (Logrank) Test	$32.970^{***} (df = 1)$

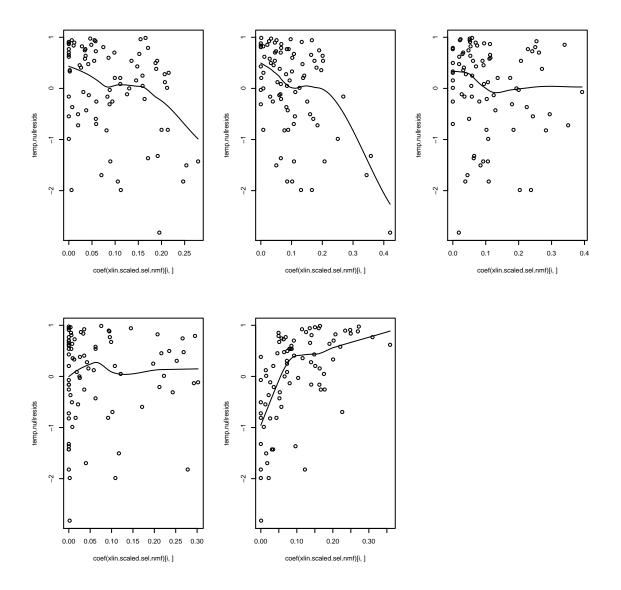
Note:

*p<0.1; **p<0.05; ***p<0.01

```
## coef(xlin.scaled.sel.nmf)[i, ] 0.0081 123 0.000254 0.259
##
## Concordance= 0.619 (se = 0.041)
## Rsquare= 0.095 (max possible= 0.997)
## Likelihood ratio test= 8.09 on 1 df, p=0.00445
## Wald test = 7.43 on 1 df, p=0.00643
## Score (logrank) test = 7.63 on 1 df, p=0.00574
## Call:
## coxph(formula = y ~ coef(xlin.scaled.sel.nmf)[i, ])
## n= 81, number of events= 64
##
                                coef exp(coef) se(coef) z Pr(>|z|)
## coef(xlin.scaled.sel.nmf)[i, ] -6.07811 0.00229 1.77655 -3.42 0.00062
                             exp(coef) exp(-coef) lower .95 upper .95
## coef(xlin.scaled.sel.nmf)[i, ] 0.00229 436 7.05e-05 0.0746
##
## Concordance= 0.636 (se = 0.041)
## Rsquare= 0.167 (max possible= 0.997)
## Likelihood ratio test= 14.8 on 1 df, p=0.000117
## Wald test = 11.7 on 1 df, p=0.000623
## Score (logrank) test = 12.2 on 1 df, p=0.00047
## Call:
## coxph(formula = y ~ coef(xlin.scaled.sel.nmf)[i, ])
## n= 81, number of events= 64
##
                              coef exp(coef) se(coef) z Pr(>|z|)
##
##
                            exp(coef) exp(-coef) lower .95 upper .95
## coef(xlin.scaled.sel.nmf)[i, ] 0.335 2.98 0.0216 5.2
##
## Concordance= 0.546 (se = 0.041)
## Rsquare= 0.008 (max possible= 0.997)
## Likelihood ratio test= 0.63 on 1 df, p=0.427
## Wald test = 0.61 on 1 df, p=0.435
## Score (logrank) test = 0.61 on 1 df, p=0.434
##
## coxph(formula = y ~ coef(xlin.scaled.sel.nmf)[i, ])
##
## n= 81, number of events= 64
##
                             coef exp(coef) se(coef) z Pr(>|z|)
##
## coef(xlin.scaled.sel.nmf)[i, ] 0.709 2.032 1.349 0.53 0.6
                             exp(coef) exp(-coef) lower .95 upper .95
##
## coef(xlin.scaled.sel.nmf)[i, ] 2.03 0.492 0.144 28.6
```

```
## Concordance= 0.503 (se = 0.041)
## Rsquare= 0.003 (max possible= 0.997)
## Likelihood ratio test= 0.27 on 1 df, p=0.604
## Wald test = 0.28 on 1 df, p=0.599
## Score (logrank) test = 0.28 on 1 df, p=0.599
## Call:
## coxph(formula = y ~ coef(xlin.scaled.sel.nmf)[i, ])
## n= 81, number of events= 64
##
                                  coef exp(coef) se(coef) z Pr(>|z|)
## coef(xlin.scaled.sel.nmf)[i, ]
                                 7.83 2507.09 1.42 5.53 3.2e-08
##
##
                              exp(coef) exp(-coef) lower .95 upper .95
## coef(xlin.scaled.sel.nmf)[i, ] 2507 0.000399 157 40152
## Concordance= 0.731 (se = 0.041)
## Rsquare= 0.279 (max possible= 0.997)
## Likelihood ratio test= 26.6 on 1 df, p=2.57e-07
## Wald test = 30.6 on 1 df, p=3.18e-08
## Score (logrank) test = 33 on 1 df, p=9.36e-09
temp.nullfit = coxph(y ~ 1)
temp.nullresids = residuals(temp.nullfit, type = "martingale")
par(mfrow = c(2, 3))
for (i in 1:nrow(coef(xlin.scaled.sel.nmf))) {
   scatter.smooth(temp.nullresids ~ coef(xlin.scaled.sel.nmf)[i, ])
```

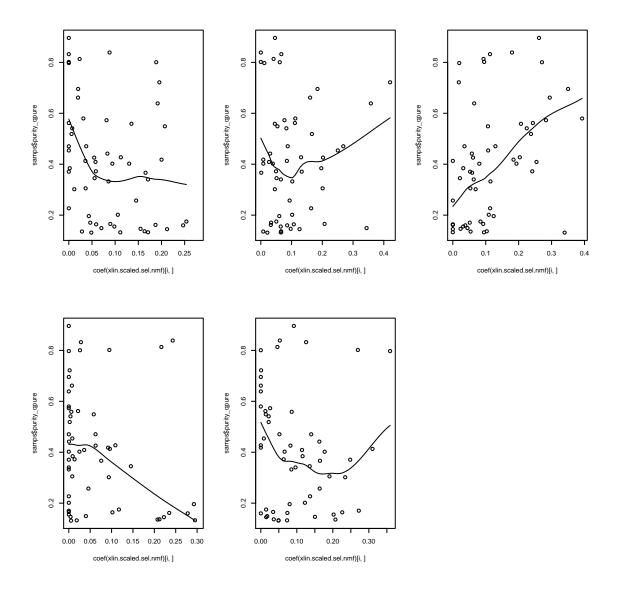
par(mfrow = c(1, 1))



4.4 Purity

```
apply(coef(xlin.scaled.sel.nmf), 1, function(xc) cor.test(samps$purity_qpure,
    xc, method = "kendall"))
##
   [[1]]
##
##
   Kendall's rank correlation tau
##
## data: samps$purity_qpure and xc
## z = -2.732, p-value = 0.006296
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##
      tau
## -0.251
##
```

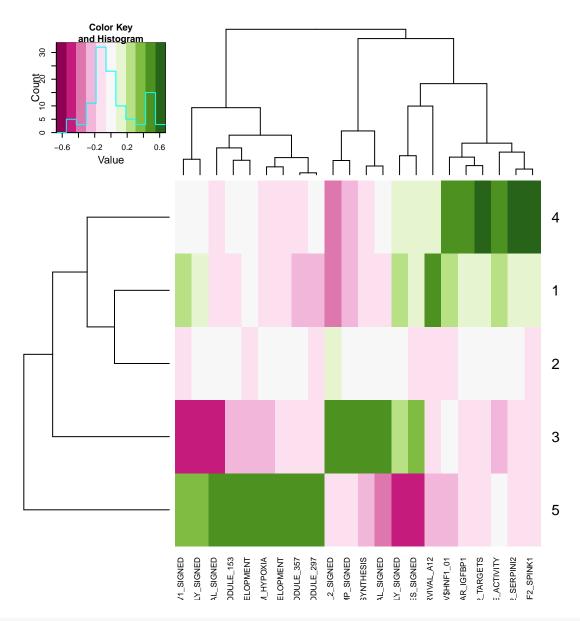
```
##
## [[2]]
## Kendall's rank correlation tau
##
## data: samps$purity_qpure and xc
## z = 0.0757, p-value = 0.9396
\#\# alternative hypothesis: true tau is not equal to 0
## sample estimates:
##
        tau
## 0.006894
##
##
## [[3]]
##
## Kendall's rank correlation tau
## data: samps$purity_qpure and xc
## z = 3.616, p-value = 0.0002987
\#\# alternative hypothesis: true tau is not equal to 0
## sample estimates:
    tau
##
## 0.3305
##
##
## [[4]]
##
## Kendall's rank correlation tau
##
## data: samps$purity_qpure and xc
## z = -1.741, p-value = 0.08161
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##
      tau
## -0.1629
##
##
## [[5]]
##
## Kendall's rank correlation tau
## data: samps$purity_qpure and xc
## z = -1.71, p-value = 0.08729
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##
      tau
## -0.1568
par(mfrow = c(2, 3))
for (i in 1:nrow(coef(xlin.scaled.sel.nmf))) {
    scatter.smooth(samps$purity_qpure ~ coef(xlin.scaled.sel.nmf)[i, ])
par(mfrow = c(1, 1))
```

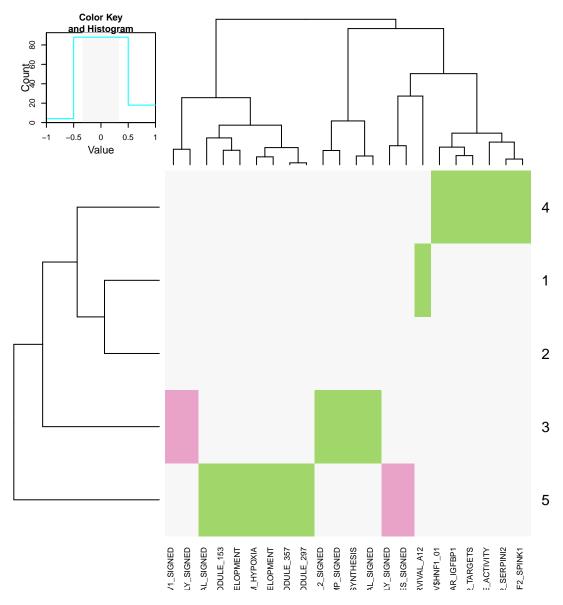


4.5 MTC P-values

```
xlin.scaled.sel.nmf.cpv.pvals$q.pure.BY = temp.qvals.BY[(length(temp.qvals.BY)/2 +
   1):length(temp.qvals.BY)]
xlin.scaled.sel.nmf.cpv.pvals
                p.pure d.surv d.pure p.surv.FWER p.pure.FWER q.surv.BY
       p.surv
## 1 4.450e-03 0.0062958 -1 -1 3.115e-02 0.037775 3.258e-02
                                1 1.052e-03 1.000000 1.712e-03
## 2 1.169e-04 0.9396386
                         -1
                                1 1.000e+00 0.002389 1.000e+00
## 3 4.265e-01 0.0002987
                          -1
## 4 6.037e-01 0.0816068
                        1
                                -1 1.000e+00 0.408034 1.000e+00
## 5 2.569e-07 0.0872948
                         1 -1 2.569e-06 0.408034 7.524e-06
## q.pure.BY
## 1 0.036880
## 2 1.000000
## 3 0.002916
## 4 0.365263
## 5 0.365263
```

4.6 MSigDB score correlation thresholding





```
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_report]
                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</pre>
        table
    }, simplify = FALSE)
})
tables
## [[1]]
## [[1]]$c1
## data frame with 0 columns and 0 rows
## [[1]]$c2
                                       GeneSet Correlation Metagenes
## 1 c2.SHEDDEN_LUNG_CANCER_GOOD_SURVIVAL_A12
                                                    0.5337
##
## [[1]]$c3
## data frame with 0 columns and 0 rows
## [[1]]$c4
## data frame with 0 columns and 0 rows
## [[1]]$c5
## data frame with 0 columns and 0 rows
##
## [[1]]$c6
## data frame with 0 columns and 0 rows
##
## [[1]]$c7
## data frame with 0 columns and 0 rows
##
## [[2]]
## [[2]]$c1
## data frame with 0 columns and 0 rows
##
## [[2]]$c2
## data frame with 0 columns and 0 rows
##
## [[2]]$c3
## data frame with 0 columns and 0 rows
##
## [[2]]$c4
## data frame with 0 columns and 0 rows
```

```
##
## [[2]]$c5
## data frame with 0 columns and 0 rows
## [[2]]$c6
## data frame with 0 columns and 0 rows
##
## [[2]]$c7
## data frame with 0 columns and 0 rows
##
##
## [[3]]
## [[3]]$c1
## data frame with 0 columns and 0 rows
## [[3]]$c2
                                                   GeneSet Correlation
##
## 1
                 c2.WAMUNYOKOLI_OVARIAN_CANCER_LMP_SIGNED
                                                              0.5388
## 2
             c2.REACTOME_GLYCEROPHOSPHOLIPID_BIOSYNTHESIS
                                                              0.5196
## 3 c2.CHARAFE_BREAST_CANCER_LUMINAL_VS_MESENCHYMAL_SIGNED
                                                              0.5189
          c2.WAMUNYOKOLI_OVARIAN_CANCER_GRADES_1_2_SIGNED
                                                              0.5115
## 5
         c2.VECCHI_GASTRIC_CANCER_ADVANCED_VS_EARLY_SIGNED
                                                              -0.5394
## Metagenes
## 1
         -3
## 2
           -3
           -3
## 3
## 4
           -3
## 5
##
## [[3]]$c3
## data frame with 0 columns and 0 rows
## [[3]]$c4
## data frame with 0 columns and 0 rows
##
## [[3]]$c5
## data frame with 0 columns and 0 rows
##
## [[3]]$c6
                 GeneSet Correlation Metagenes
## 1 c6.LEF1_UP.V1_SIGNED -0.5047
##
## [[3]]$c7
## data frame with 0 columns and 0 rows
##
##
## [[4]]
## [[4]]$c1
## data frame with 0 columns and 0 rows
## [[4]]$c2
##
                   GeneSet Correlation Metagenes
## 1 c2.BERGER_MBD2_TARGETS 0.5666
```

```
## [[4]]$c3
## GeneSet Correlation Metagenes
## 1 c3.V$HNF1_01 0.5236 4
##
## [[4]]$c4
     GeneSet Correlation Metagenes
##
## 1 c4.GNF2_SPINK1 0.6716
                       0.6383
## 2 c4.GNF2_SERPINI2
## 3 c4.CAR_IGFBP1
                       0.5006
## [[4]]$c5
                       GeneSet Correlation Metagenes
## 1 c5.CARBOXYPEPTIDASE_ACTIVITY 0.5403 4
##
## [[4]]$c6
## data frame with 0 columns and 0 rows
## [[4]]$c7
## data frame with 0 columns and 0 rows
##
## [[5]]
## [[5]]$c1
## data frame with 0 columns and 0 rows
## [[5]]$c2
##
                                              GeneSet Correlation
                                       c2.KIM_HYPOXIA 0.5226
                                                        0.5102
## 2 c2.LIEN_BREAST_CARCINOMA_METAPLASTIC_VS_DUCTAL_SIGNED
                  c2.DOANE_BREAST_CANCER_CLASSES_SIGNED
                                                       -0.5065
## 4 c2.RICKMAN_TUMOR_DIFFERENTIATED_WELL_VS_POORLY_SIGNED
                                                       -0.5412
## Metagenes
## 1 5
          5
## 2
## 3
          -5
## 4
          -5
##
## [[5]]$c3
## data frame with 0 columns and 0 rows
##
## [[5]]$c4
## GeneSet Correlation Metagenes
## 1 c4.MODULE_153 0.5369 5
## 2 c4.MODULE_297
                    0.5047
                                   5
## 3 c4.MODULE_357
                    0.5003
##
## [[5]]$c5
                 GeneSet Correlation Metagenes
## 1 c5.TISSUE_DEVELOPMENT 0.5369 5
## 2 c5.ECTODERM_DEVELOPMENT
                              0.5121
##
## [[5]]$c6
## data frame with 0 columns and 0 rows
```

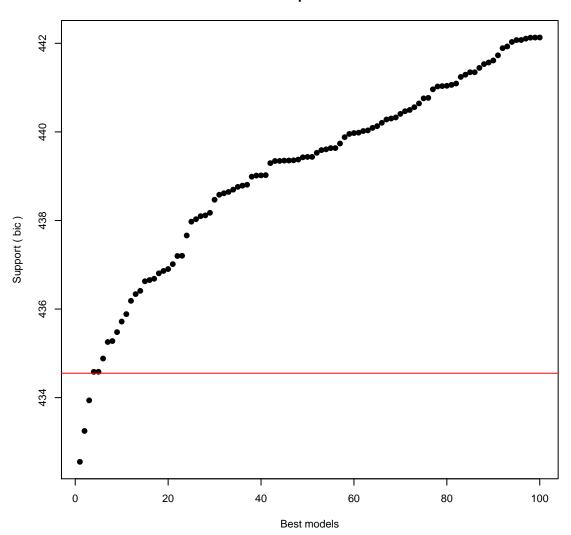
```
## [[5]]$c7
## data frame with 0 columns and 0 rows
print(asreg.result)
## glmulti.analysis
## Method: h / Fitting: coxph / IC used: bic
## Level: 2 / Marginality: TRUE
## From 100 models:
## Best IC: 432.551249683703
## Best model:
## [1] "Surv(time, event) ~ 1 + mg.1 + mg.2 + mg.5"
## Evidence weight: 0.125293531872459
## Worst IC: 442.13132559522
## 3 models within 2 IC units.
## 68 models to reach 95% of evidence weight.
coef(asreg.result)
            Estimate Uncond. variance Nb models Importance +/- (alpha=0.05)
                     0.10270 4 0.008333
## mg.2:mg.4 0.098835
                                                                0.6381
## mg.1:mg.4 0.008364
                           0.06559
                                           4
                                              0.011946
                                                                0.5100
                                          8 0.028848
## mg.4:mg.5 -0.029790
                           0.46834
                                                               1.3627
## mg.1:mg.3 -0.773195
                           4.32555
                                         11 0.035520
                                                               4.1413
                           5.64241
                                         11 0.037161
## mg.2:mg.3 -1.004421
                                                                4.7298
## mg.1:mg.5 0.086515
                           1.30755
                                         11 0.042344
                                                                2.2769
## mg.1:mg.2 0.434806
                                         10 0.043311
                                                               3.1760
                           2.54417
## mg.2:mg.5 0.727347
                           4.51337
                                         13 0.058518
                                                               4.2302
                                         15 0.104823
## mg.3:mg.4 -4.109188
                          65.59200
                                                              16.1264
                                        54 0.377634
                           4.69184
## mg.4 1.447956
                                                               4.3131
## mg.3:mg.5 21.751186
                                         42 0.382604
                                                              58.4463
                         861.56514
                          10.75460
                                        71 0.585706
## mg.1 -3.029032
                                                               6.5300
                                       74 0.590001
70 0.634246
          -0.023278
                           4.25102
## mg.3
                                                               4.1054
## mg.2
                           11.21244
          -3.453980
                                                               6.6675
## mg.5
           5.716940
                           9.24799
                                         88 0.906838
                                                               6.0553
summary(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 -4.48079 0.01132 2.08425 -2.15 0.03157
## mg.2 -5.55809 0.00386 1.97613 -2.81 0.00491
## mg.5 5.25086 190.73055 1.57321 3.34 0.00084
##
       exp(coef) exp(-coef) lower .95 upper .95
## mg.1 1.13e-02 8.83e+01 1.91e-04 0.673
## mg.2 3.86e-03 2.59e+02 8.02e-05 0.185
## mg.5 1.91e+02 5.24e-03 8.74e+00 4164.293
```

##

```
## Concordance= 0.741 (se = 0.041 )
## Rsquare= 0.375 (max possible= 0.997 )
## Likelihood ratio test= 38 on 3 df, p=2.78e-08
## Wald test = 36.5 on 3 df, p=6.03e-08
## Score (logrank) test = 39.2 on 3 df, p=1.58e-08
```

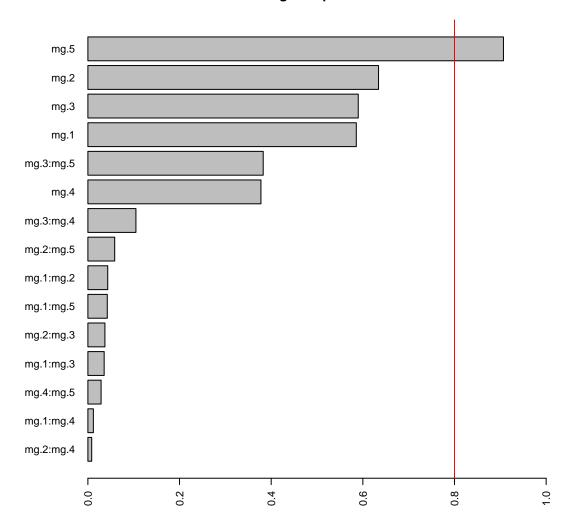
```
plot(asreg.result, type = "p")
```

IC profile



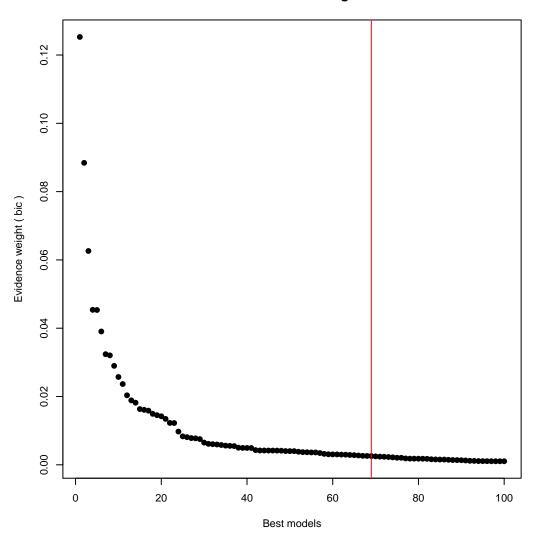
```
plot(asreg.result, type = "s")
```

Model-averaged importance of terms



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

Profile of model weights



```
glmnet.coef.1se

## 5 x 1 sparse Matrix of class "dgCMatrix"

## mg.1 .

## mg.2 -0.6197

## mg.3 .

## mg.5 3.1918

glmnet.coef.min

## 5 x 1 sparse Matrix of class "dgCMatrix"

## mg.1 -2.236

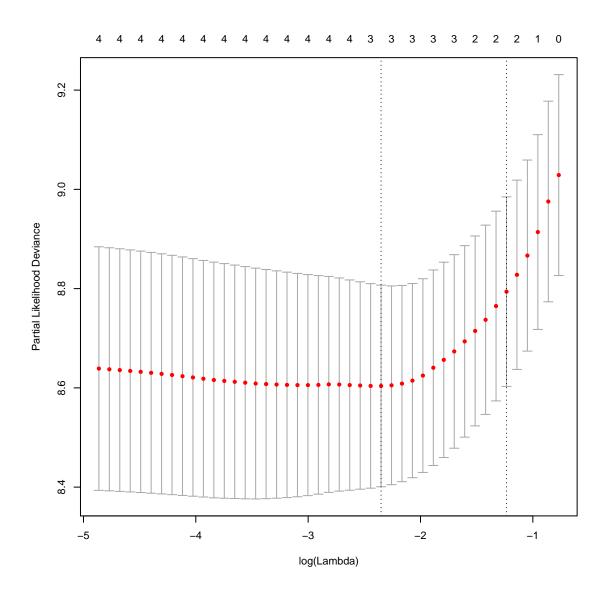
## mg.2 -3.351

## mg.3 .

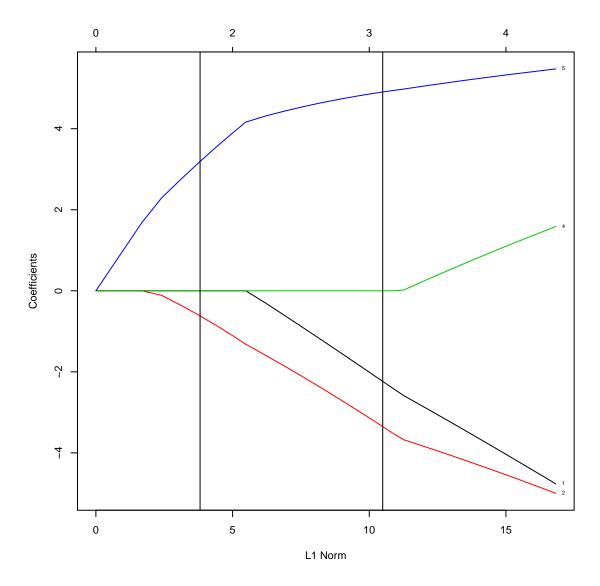
## mg.4 .
```

```
## mg.5 4.906
```

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



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



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

## 5 x 1 sparse Matrix of class "dgCMatrix"

## mg.1 .

## mg.2 -24.277313

## mg.3 -0.002698

## mg.4 .

## mg.5 93.332569

adaglmnet.coef.min/adaglmnet.weights

## 5 x 1 sparse Matrix of class "dgCMatrix"

## mg.1 -97.698

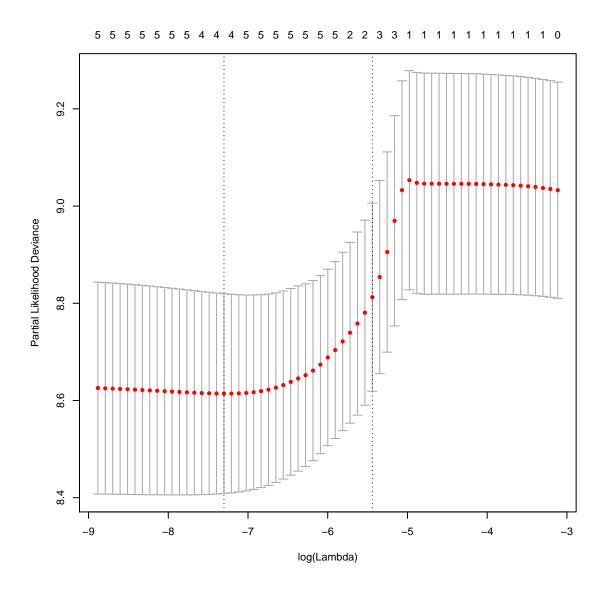
## mg.2 -115.856

## mg.3 .

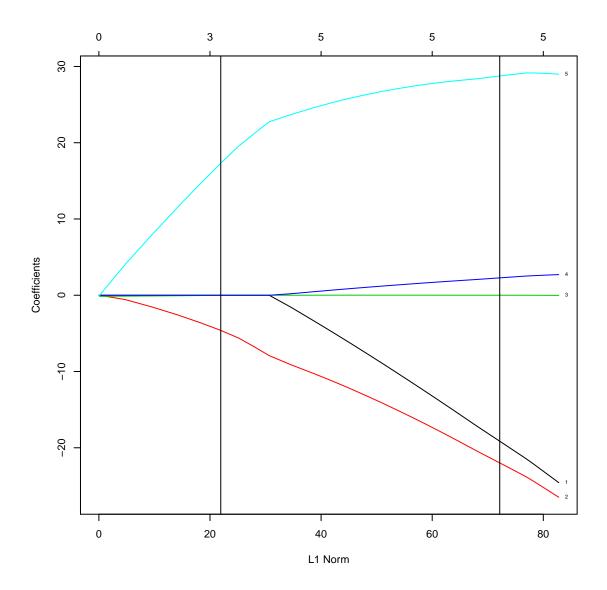
## mg.4 3.818
```

```
## mg.5 154.967
```

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



```
plot(adaglmnet.fit.cv$glmnet.fit, label = TRUE)
abline(v = sum(abs(adaglmnet.coef.1se)))
abline(v = sum(abs(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.iso885915
##
                                          LC_NUMERIC=C
   [3] LC_TIME=en_US.iso885915
                                          LC_COLLATE=en_US.iso885915
##
   [5] LC_MONETARY=en_US.iso885915
                                          LC_MESSAGES=en_US.iso885915
##
    [7] LC_PAPER=en_US.iso885915
                                          LC_NAME=en_US.iso885915
   [9] LC_ADDRESS=en_US.iso885915
                                          LC_TELEPHONE=en_US.iso885915
                                          LC_IDENTIFICATION=en_US.iso885915
   [11] LC_MEASUREMENT=en_US.iso885915
##
## attached base packages:
```

```
## [1] splines
                 parallel methods
                                     stats
                                               graphics grDevices utils
## [8] datasets
                base
## other attached packages:
                                                foreach 1.4.2
## [1] doParallel 1.0.8
                           iterators 1.0.7
## [4] ahaz_1.14
                            survival_2.37-7
                                                stargazer_5.1
## [7] gplots_2.14.2
                            NMF_0.20.4
                                                synchronicity_1.1.4
## [10] bigmemory_4.4.6
                            BH_1.54.0-5
                                                bigmemory.sri_0.1.3
## [13] Biobase_2.26.0
                            BiocGenerics_0.12.1 cluster_1.15.3
## [16] rngtools_1.2.4
                            pkgmaker_0.22
                                                registry_0.2
## [19] RColorBrewer_1.0-5 energy_1.6.2
                                                glmnet_1.9-8
## [22] Matrix_1.1-4
                            glmulti_1.0.7
                                                rJava_0.9-6
## [25] xtable_1.7-4
##
## loaded via a namespace (and not attached):
## [1] bitops_1.0-6
                           boot_1.3-13
                                              caTools_1.17.1
## [4] codetools_0.2-9
                           colorspace_1.2-4
                                              compiler_3.1.1
## [7] digest_0.6.4
                           gdata_2.13.3
                                              ggplot2_1.0.0
## [10] grid_3.1.1
                           gridBase_0.4-7
                                              gtable_0.1.2
                           KernSmooth_2.23-13 lattice_0.20-29
## [13] gtools_3.4.1
## [16] MASS_7.3-35
                           munsell_0.4.2
                                              plyr_1.8.1
## [19] proto_0.3-10
                           Rcpp_0.11.3
                                              reshape2_1.4
## [22] scales_0.2.4
                           stringr_0.6.2
                                              tools_3.1.1
sessionInfo()
## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
## [1] LC_CTYPE=en_US.iso885915
                                          LC_NUMERIC=C
## [3] LC_TIME=en_US.iso885915
                                          LC_COLLATE=en_US.iso885915
## [5] LC_MONETARY=en_US.iso885915
                                          LC_MESSAGES=en_US.iso885915
## [7] LC_PAPER=en_US.iso885915
                                          LC_NAME=en_US.iso885915
## [9] LC_ADDRESS=en_US.iso885915
                                          LC_TELEPHONE=en_US.iso885915
## [11] LC_MEASUREMENT=en_US.iso885915
                                         LC_IDENTIFICATION=en_US.iso885915
## attached base packages:
## [1] parallel methods
                          splines
                                               graphics grDevices utils
                                     stats
## [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_1.15.3
                            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_1.3-13
                                              caTools_1.17.1
## [4] codetools_0.2-9
                        colorspace_1.2-4 digest_0.6.4
```

```
## [7] doParallel_1.0.8
                          evaluate_0.5.5
                                             foreach_1.4.2
## [10] formatR_1.0
                          gdata_2.13.3
                                             ggplot2_1.0.0
## [13] grid_3.1.1
                          gridBase_0.4-7
                                             gtable_0.1.2
## [16] gtools_3.4.1
                          highr_0.4
                                             iterators_1.0.7
## [19] KernSmooth_2.23-13 labeling_0.3
                                             lattice_0.20-29
## [22] MASS_7.3-35
                          munsell_0.4.2
                                             plyr_1.8.1
## [25] proto_0.3-10
                          Rcpp_0.11.3
                                             reshape2_1.4
## [28] scales_0.2.4
                          stringr_0.6.2
                                             tools_3.1.1
```