SIS NMF: Diagnosis to Disease-specific Death

November 28, 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 7/8

library(glmulti)

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

library(glmnet)

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

library(RColorBrewer)
library(gplots)

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

library(xtable)
library(stargazer)
```

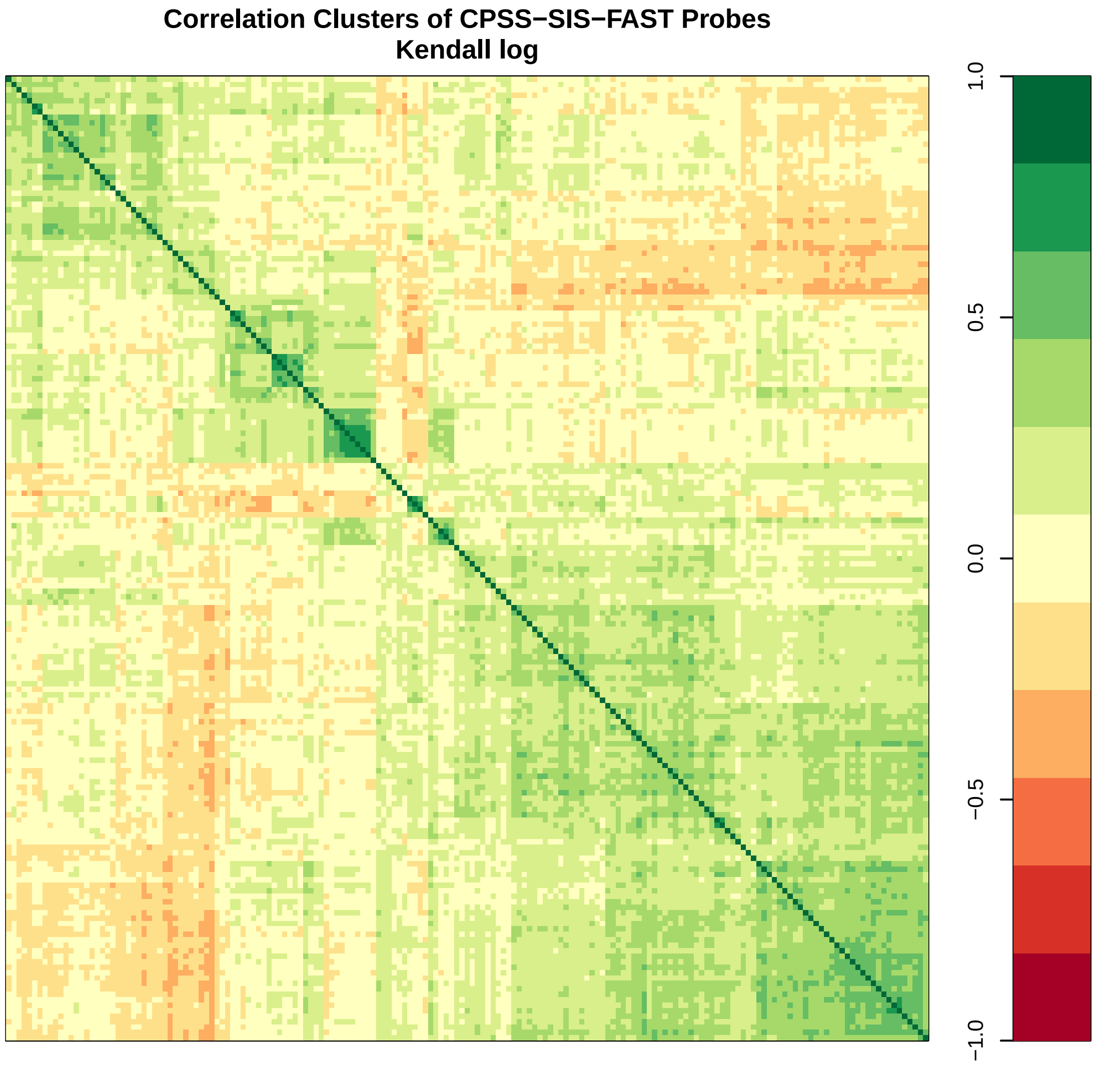
```
##
## Please cite as:
##
##  Hlavac, Marek (2014).  stargazer:  LaTeX code and ASCII text for well-formatted regression
and summary statistics tables.
##  R package version 5.1.  http://CRAN.R-project.org/package=stargazer

setwd(data_dir)

## Error in setwd(data_dir):  object 'data_dir' not found

load("image.rda")
```

2 Probe selection

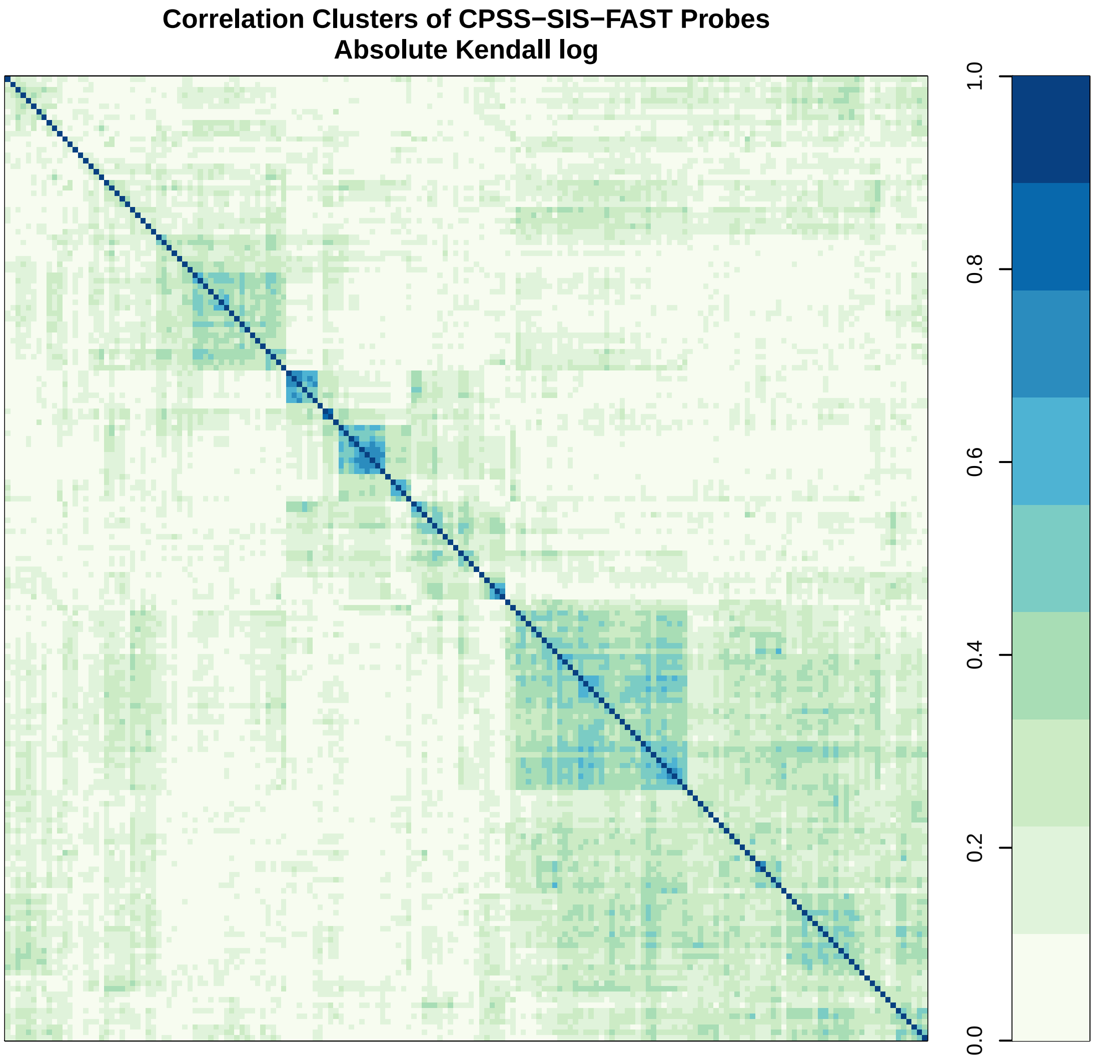
```
table(cpss.sis$sel)

##
## FALSE  TRUE
## 12823   177

mean(cpss.sis$sel)

## [1] 0.01362
```

3 Expression correlation

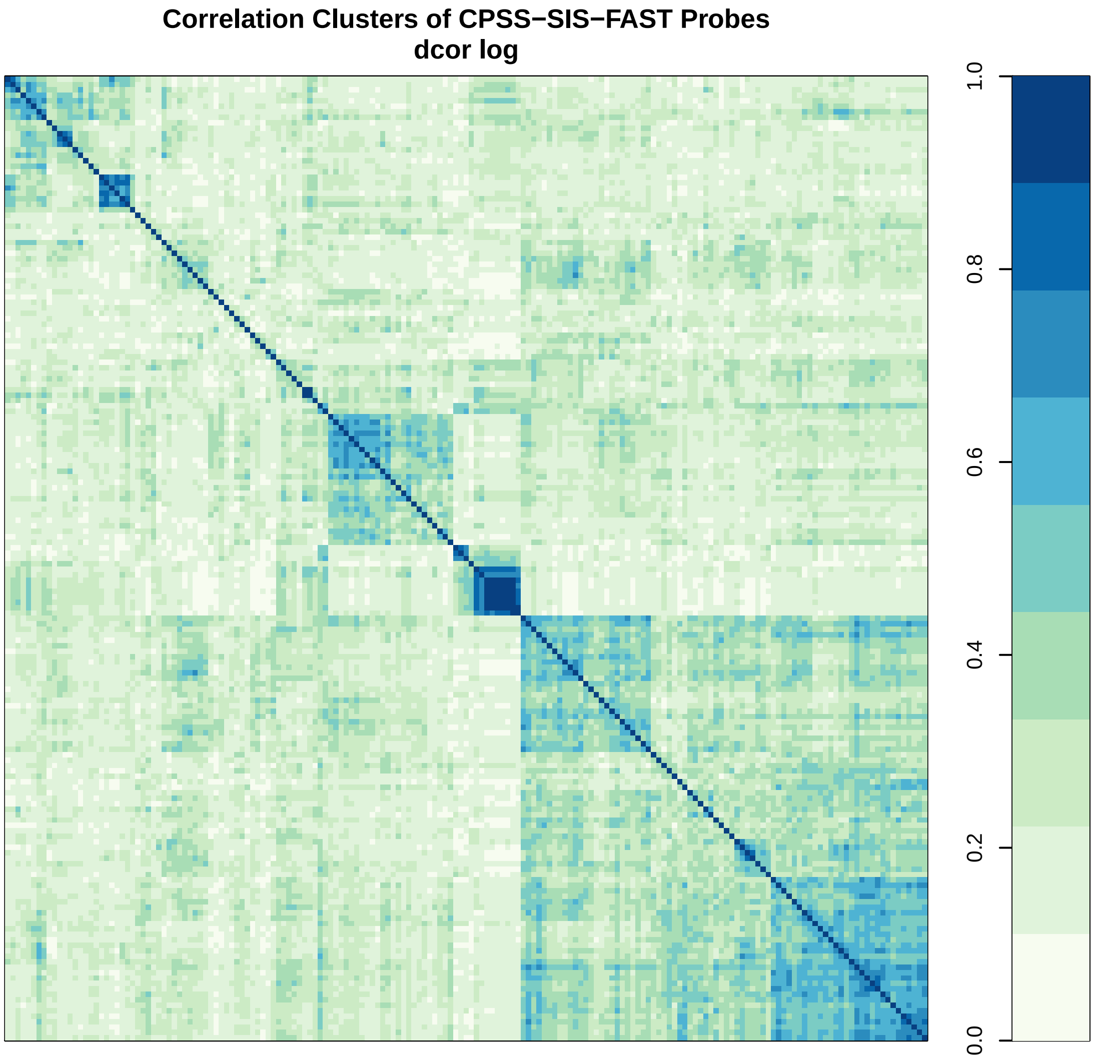
```
corPlot(x.sel.kcor, main = "Correlation Clusters of CPSS-SIS-FAST Probes\nKendall log",
    useRaster = FALSE)
```



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



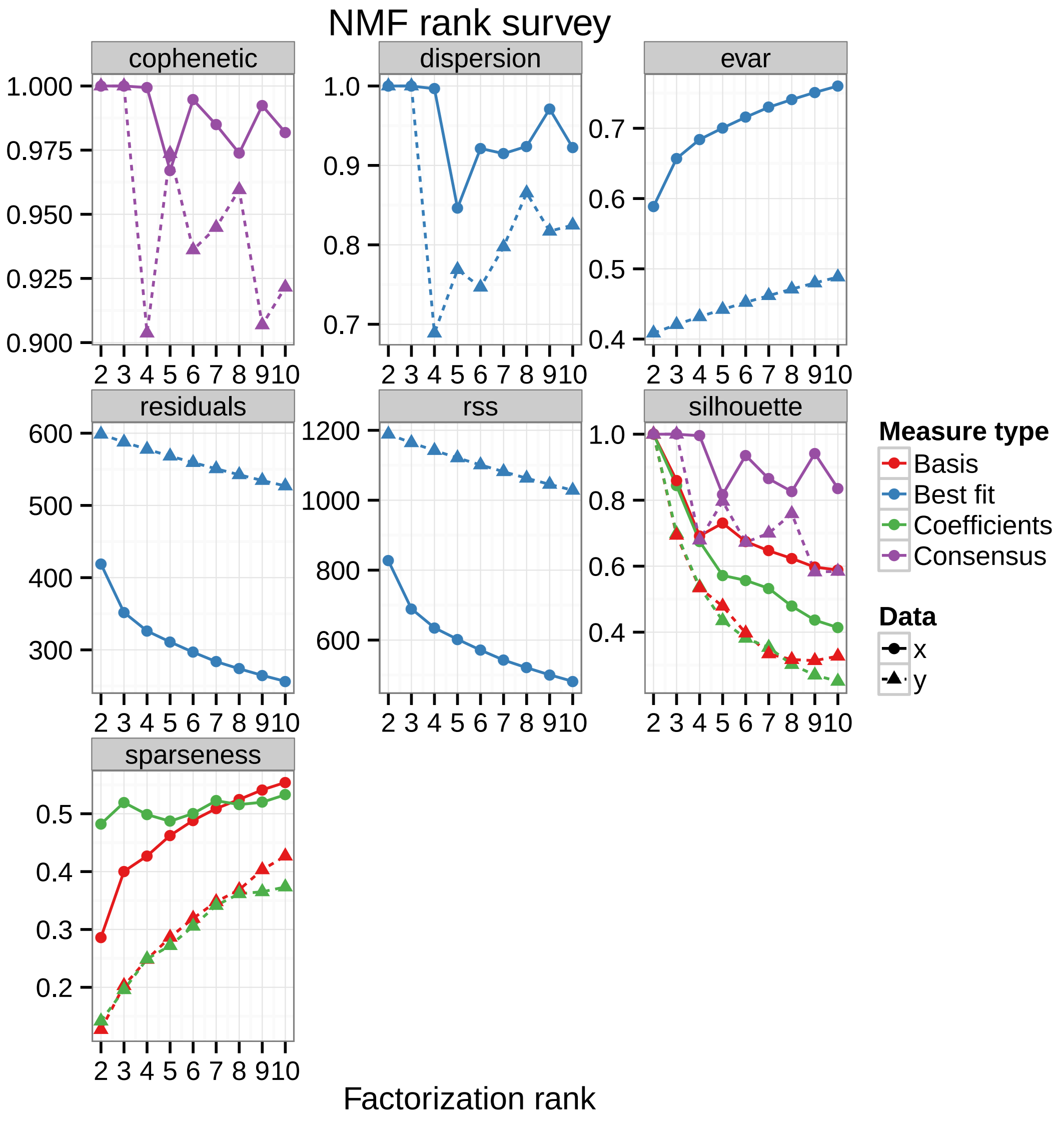
```
corPlot(x.sel.dcor, zlim = c(0, 1), pal = "GnBu", main = "Correlation Clusters of CPSS-SIS-FAST Probes\r
    useRaster = FALSE)
```



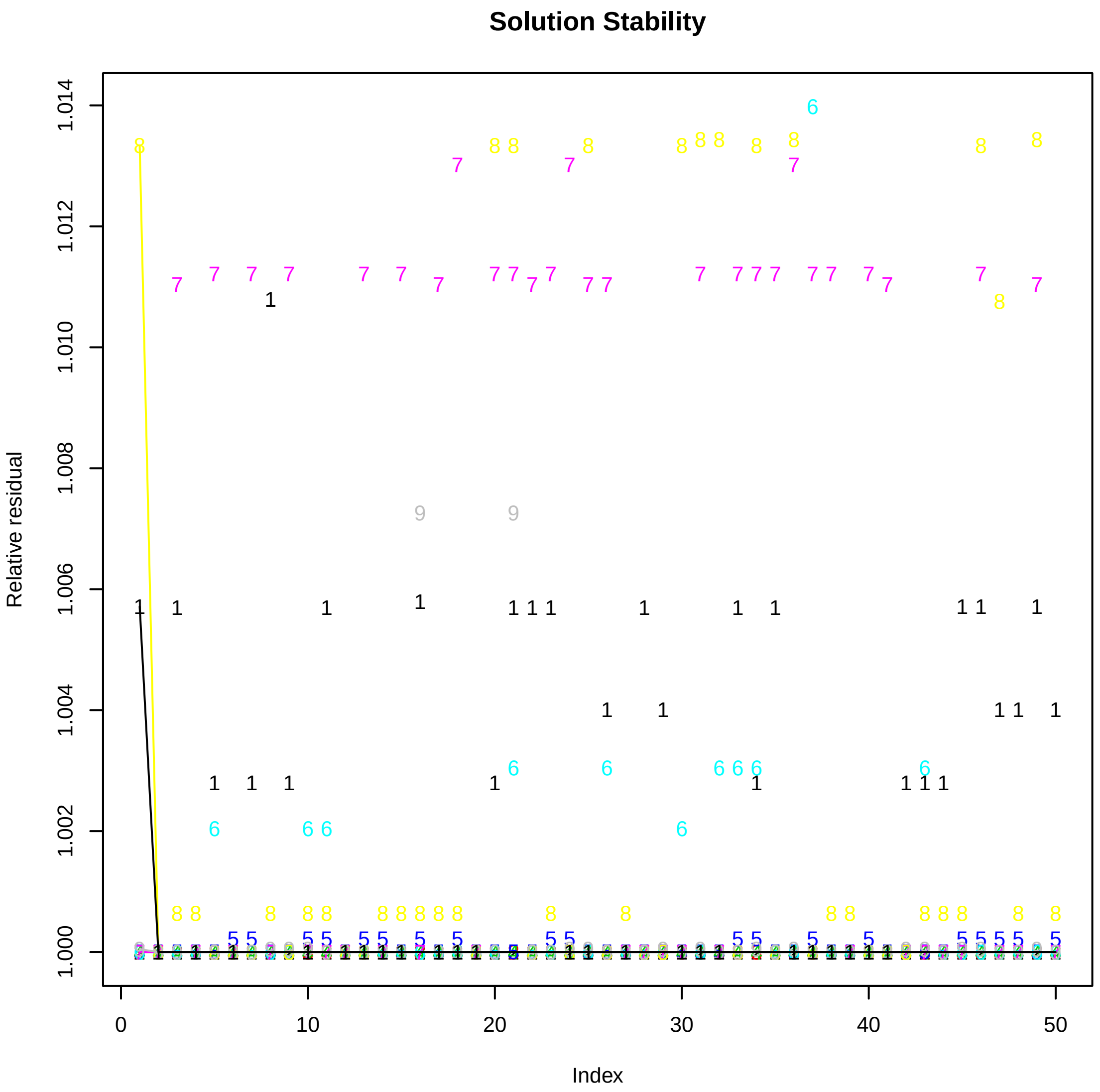
4 Factorization

4.1 Rank estimation

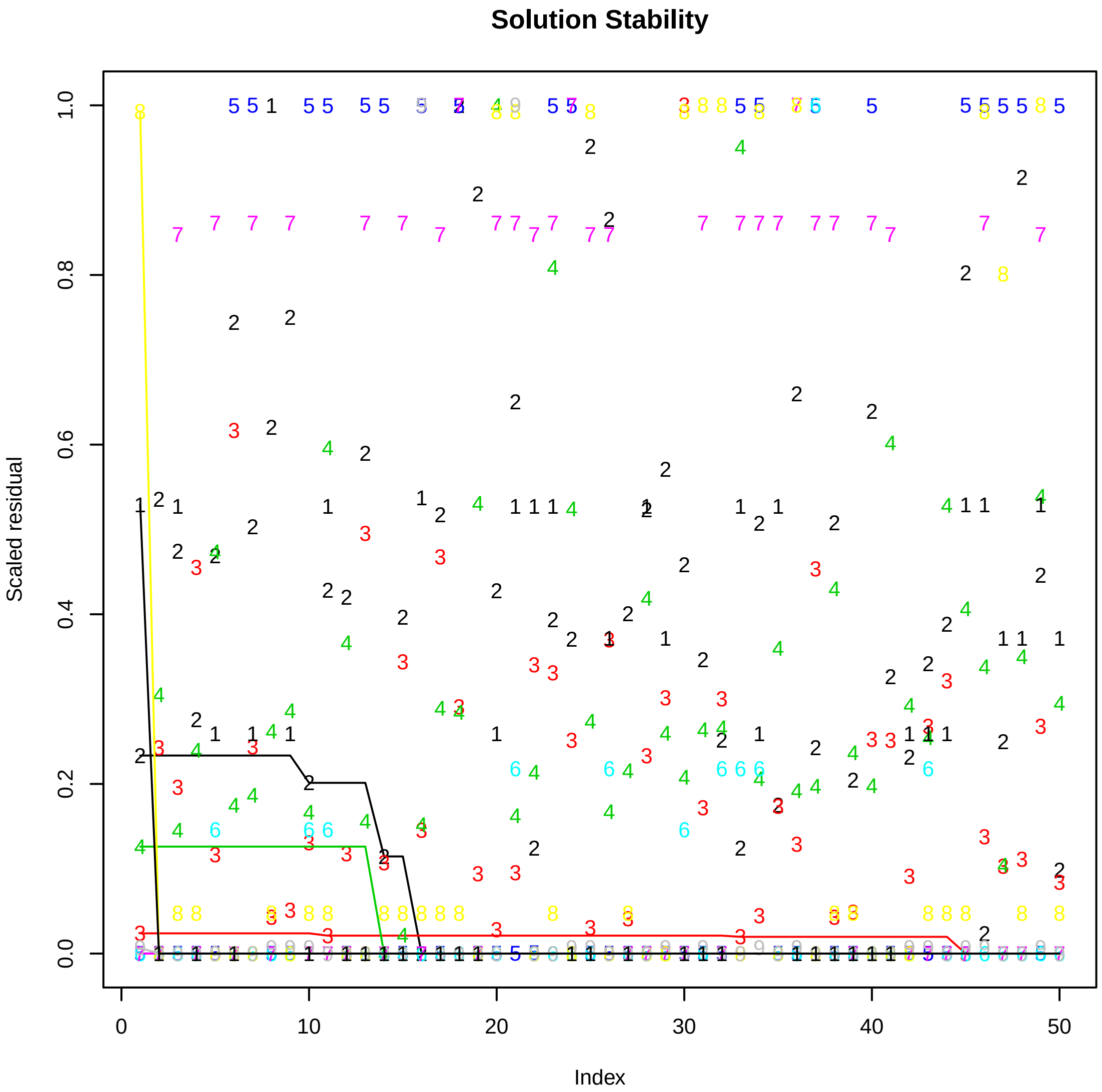
```
plot(temp.nmf.rank, temp.nmf.rank.random[[1]])
```



```
# for (i in 1:length(temp.nmf.rank£fit)) { if
# (as.numeric(names(temp.nmf.rank£fit)[i]) <= 10) {
# 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)
}
```

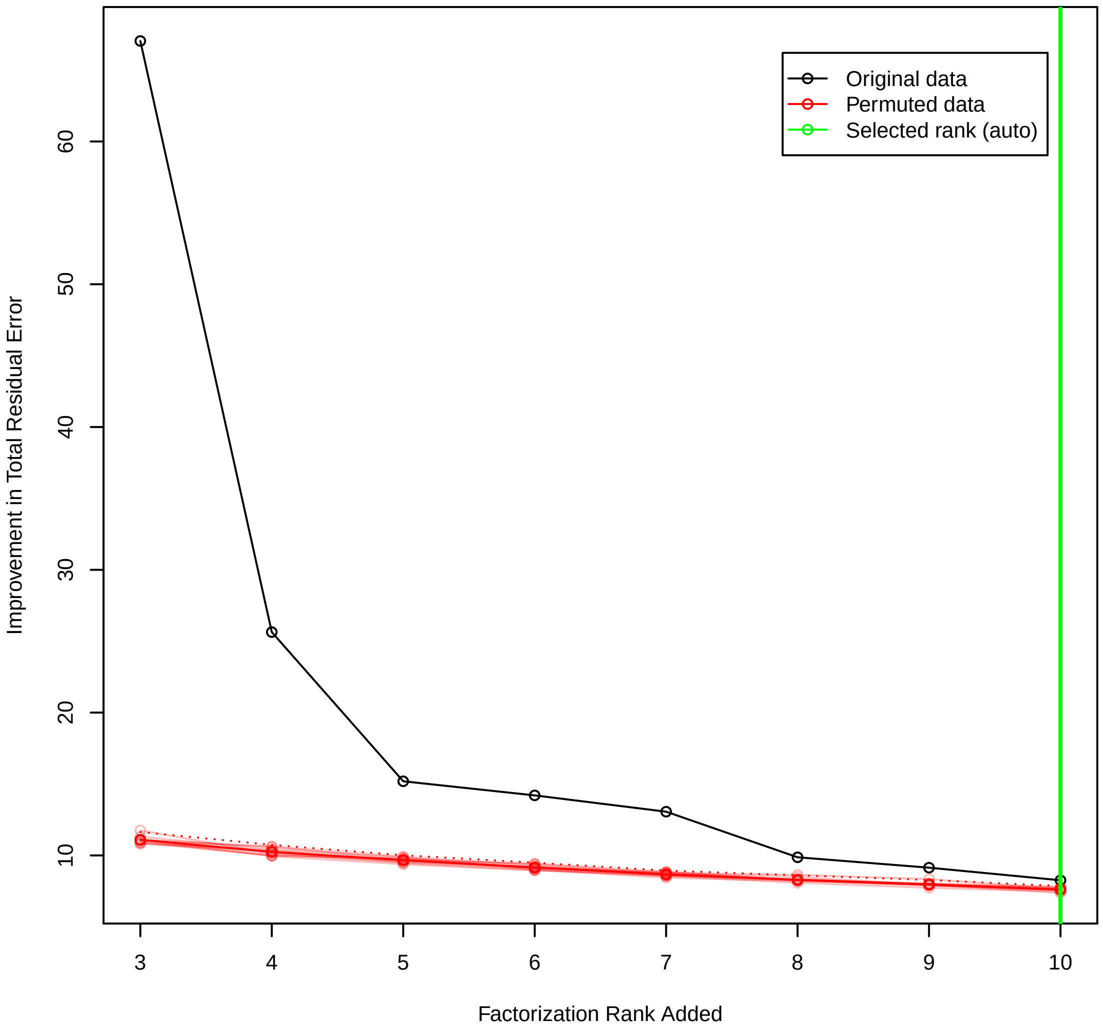


```
plot(0 ~ 0, type = "n", xlim = c(1, nrow(temp.resids)), ylim = range(temp.resids_scaled),
    ylab = "Scaled residual", main = "Solution Stability")
for (i in 1:ncol(temp.resids)) {
    points(temp.resids_scaled[, i], col = i, pch = colnames(temp.resids)[i])
    lines(cummin(temp.resids_scaled[,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



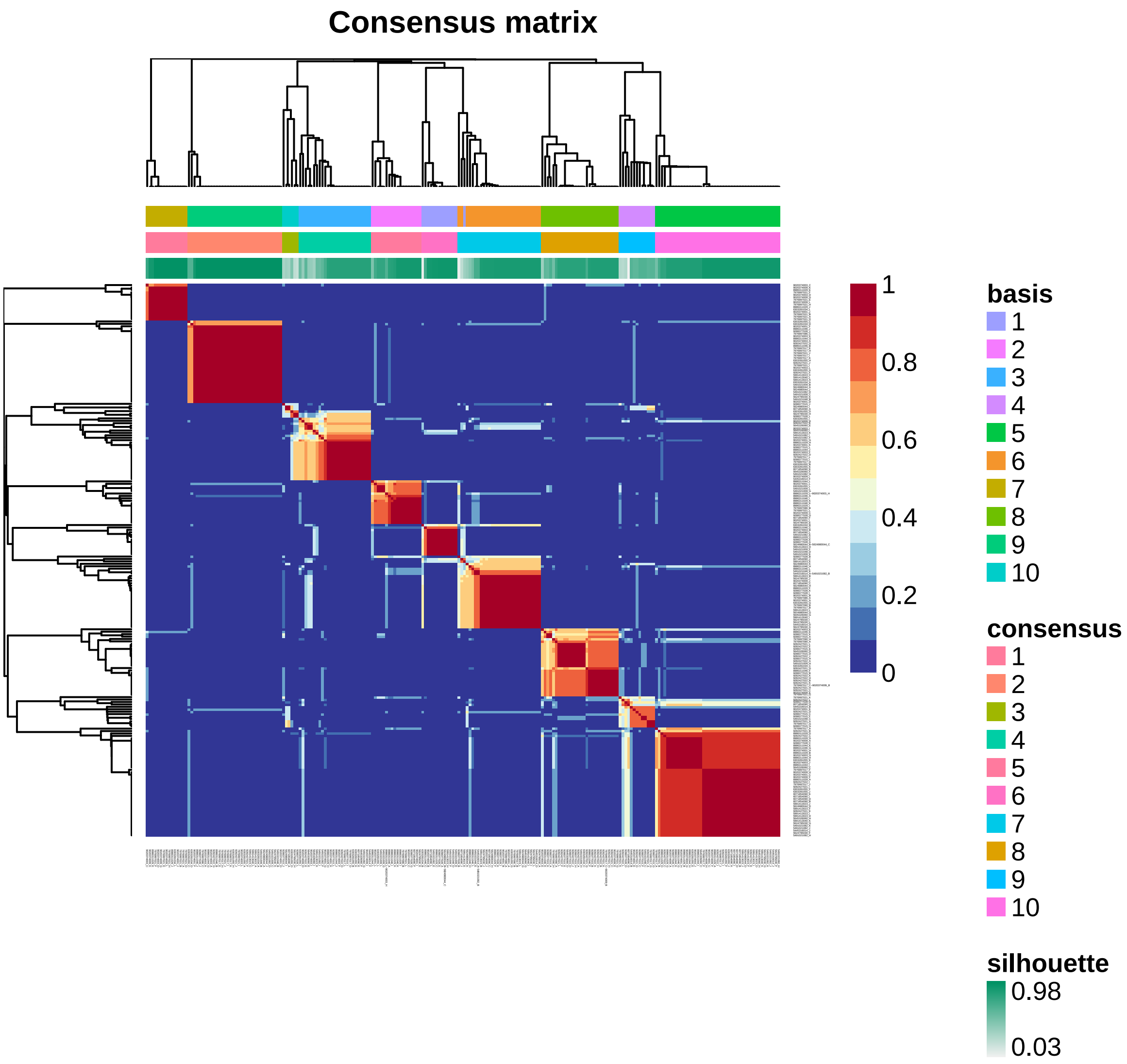
```
plot(nmf.rankrange[-1], -temp.orig_resids.delta, type = "o", col = "black",
    pch = 21, ylim = range(-c(temp.orig_resids.delta, temp.perm_resids.delta.mean)),
    xlab = "Factorization Rank Added", ylab = "Improvement in Total Residual Error")
lines(nmf.rankrange[-1], -temp.perm_resids.delta.mean, 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"
    nmf.rank = nmf.rank.auto
} else {
    temp.col = "blue"
}
abline(v = nmf.rank, col = temp.col, lwd = 2)
```

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

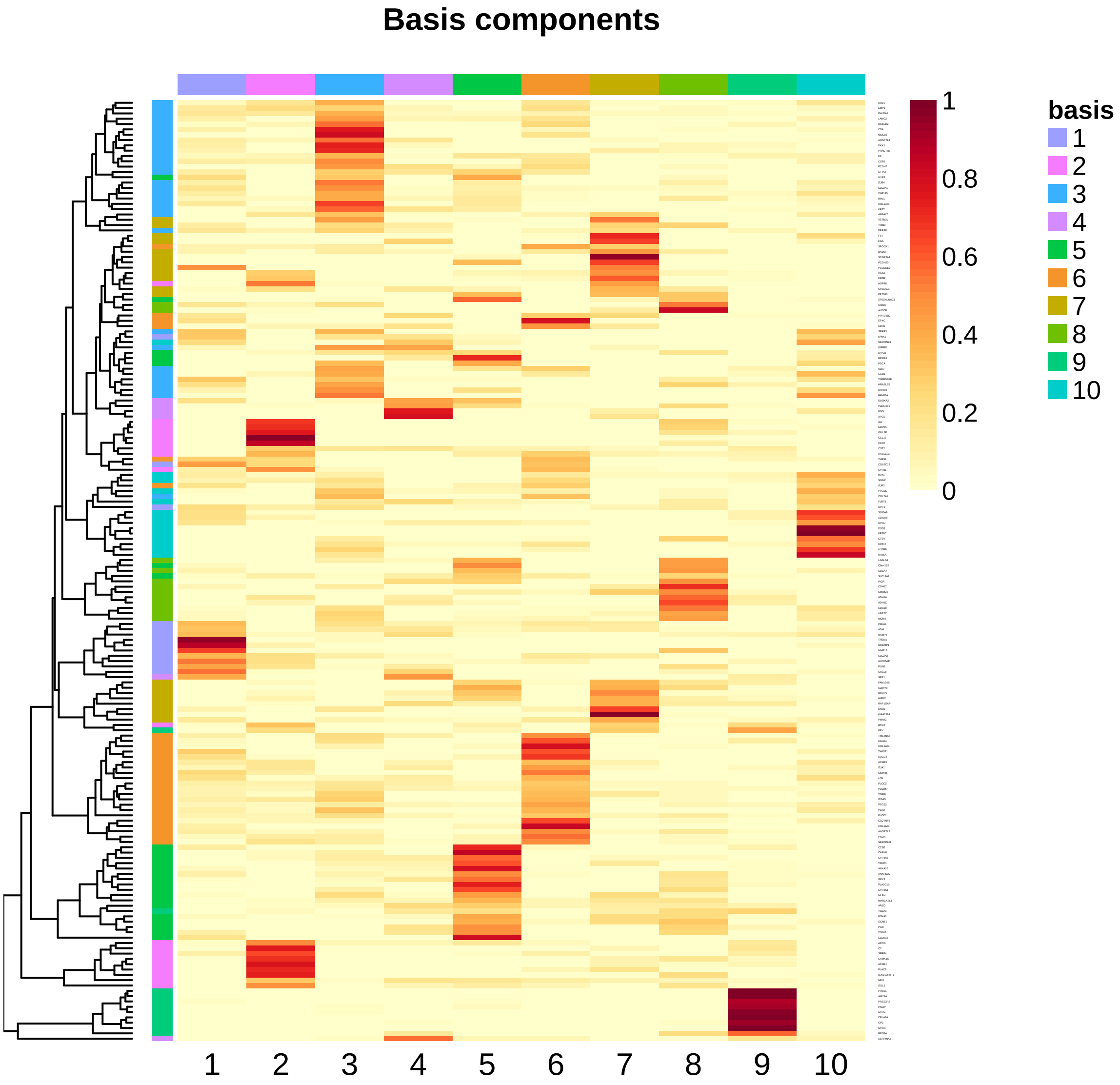


4.2 Fit

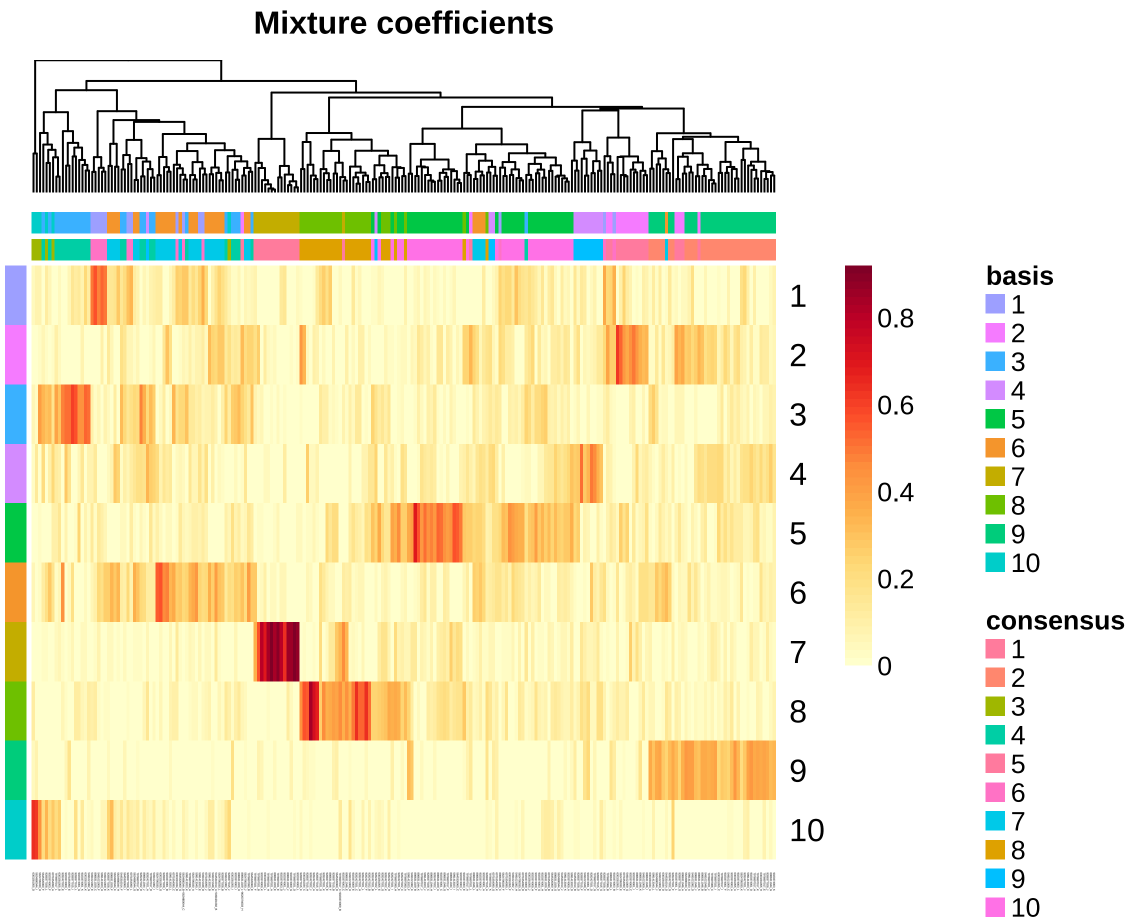
```
consensusmap(xlin.scaled.sel.nmf)
```



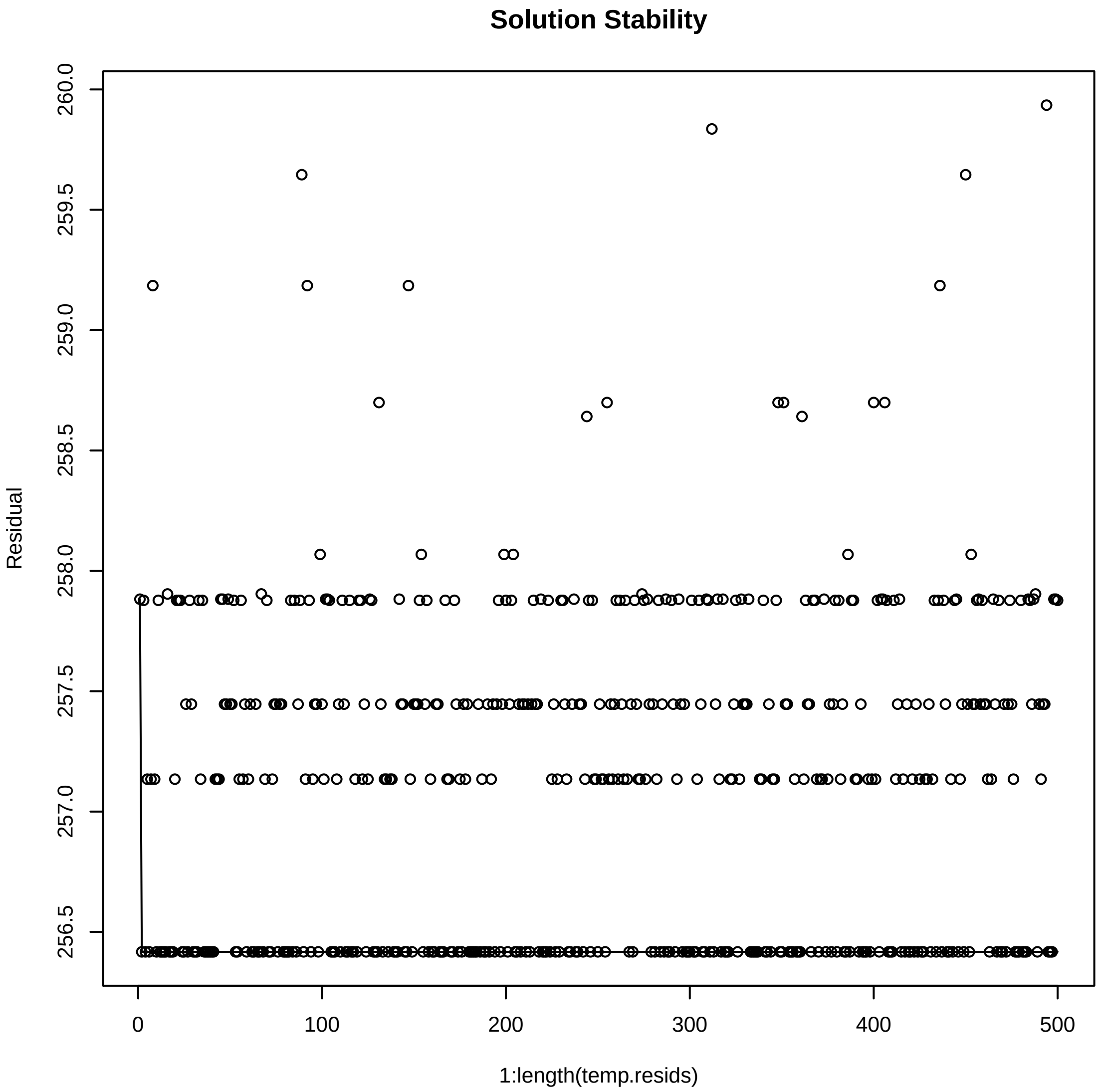
```
basismap(xlin.scaled.sel.nmf)
```



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



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



4.3 Component CPV associations

4.3.1 Survival: Diagnosis to disease-specific death

```
Error in eval(expr, envir, enclos):  object ’y’ not found
```

```
for (i in 1:nrow(coef(xlin.scaled.sel.nmf))) {
    print(summary(coxph(y ~ coef(xlin.scaled.sel.nmf)[i, ])))
}

## Error in print(summary(coxph(y ~ coef(xlin.scaled.sel.nmf)[i, ]))):  error in evaluating
the argument ’x’ in selecting a method for function ’print’:  Error in summary(coxph(y ~ coef(xlin.scale
])) :
##  error in evaluating the argument ’object’ in selecting a method for function ’summary’:
Error in eval(expr, envir, enclos) :  object ’y’ not found
## Calls:  coxph ...  eval -> model.frame -> model.frame.default -> eval -> eval
```

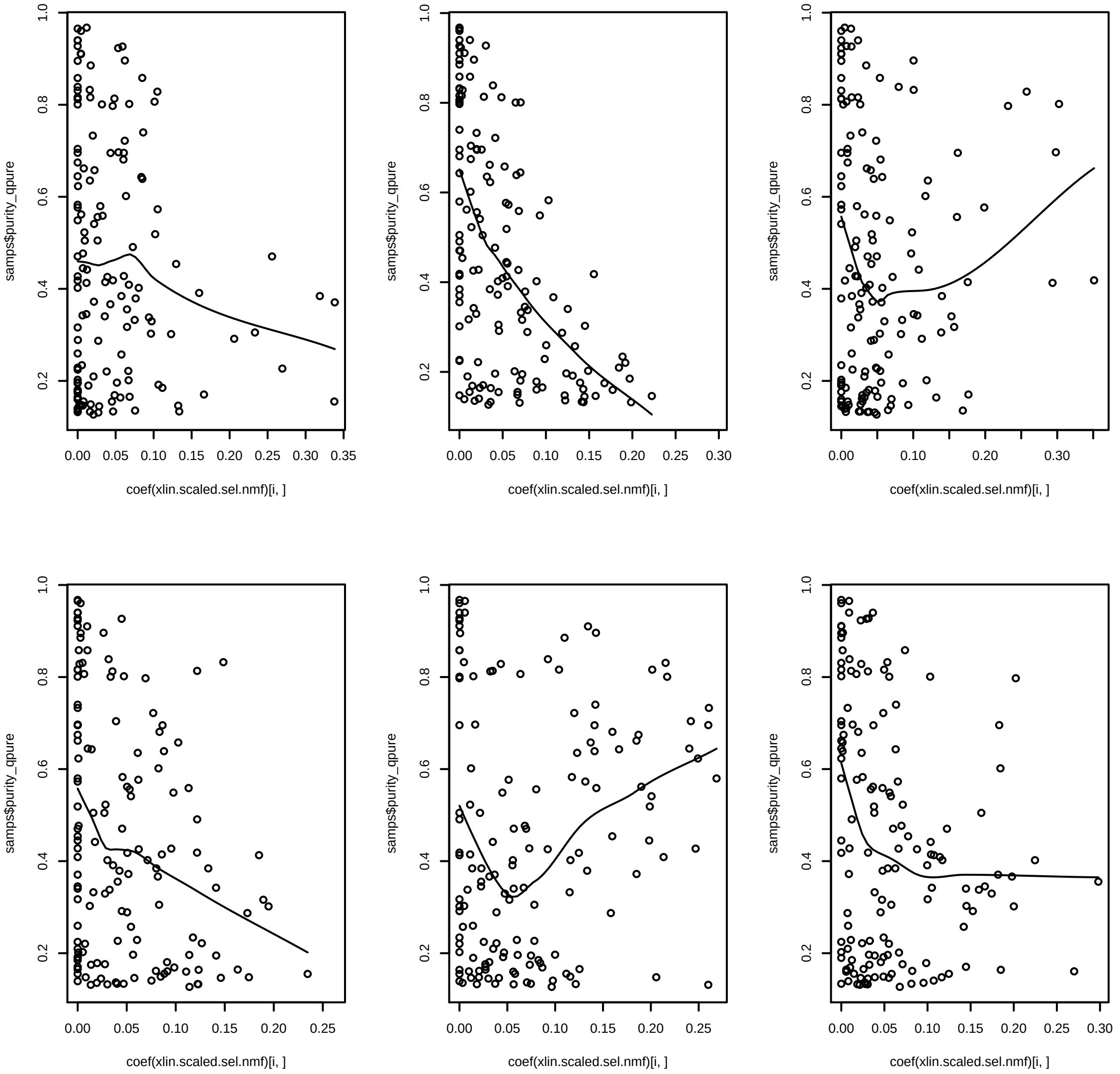
```
temp.nullfit = coxph(y ~ 1)

## Error in eval(expr, envir, enclos):  object ’y’ not found

temp.nullresids = residuals(temp.nullfit, type = "martingale")

## Error in residuals(temp.nullfit, type = "martingale"):  error in evaluating the argument
’object’ in selecting a method for function ’residuals’:  Error:  object ’temp.nullfit’ not
found

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, ])
}

## Error in eval(expr, envir, enclos):  object ’temp.nullresids’ not found

par(mfrow = c(1, 1))
```

4.4 Purity

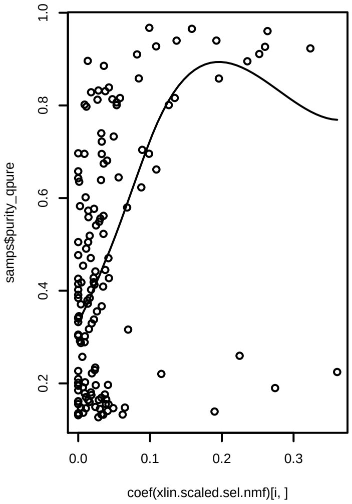
```
apply(coef(xlin.scaled.sel.nmf), 1, function(xc) cor.test(samps$purity_qpure,
    xc, method = "kendall"))

## [[1]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -0.6882, p-value = 0.4913
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.03945
##
##
## [[2]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -7.107, p-value = 1.182e-12
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.4028
##
##
## [[3]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -1.199, p-value = 0.2304
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.06757
##
##
## [[4]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.754, p-value = 0.0001739
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.2144
##
##
## [[5]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 0.6498, p-value = 0.5158
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## 0.03658
##
##
## [[6]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.246, p-value = 0.001171
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.1821
##
##
## [[7]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 4.747, p-value = 2.067e-06
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.2675
##
##
```

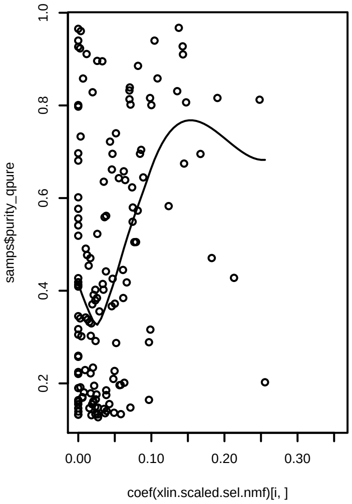
```
## [[8]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 3.22, p-value = 0.001283
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##    tau
## 0.1821
##
##
## [[9]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = -3.951, p-value = 7.784e-05
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## -0.2325
##
##
## [[10]]
##
##  Kendall's rank correlation tau
##
## data:  samps$purity_qpure and xc
## z = 0.2922, p-value = 0.7702
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##     tau
## 0.01727
```

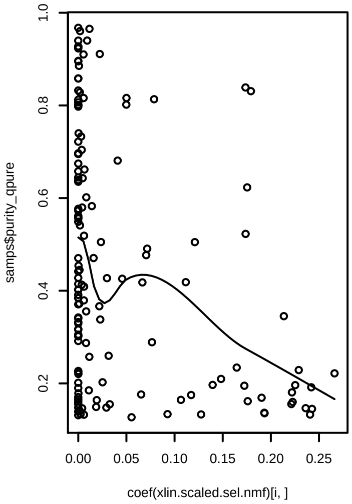
```
par(mfrow = c(2, 3))
for (i in 1:nrow(coef(xlin.scaled.sel.nmf))) {
    scatter.smooth(samps$purity_qpure ~ coef(xlin.scaled.sel.nmf)[i, ])
}
```

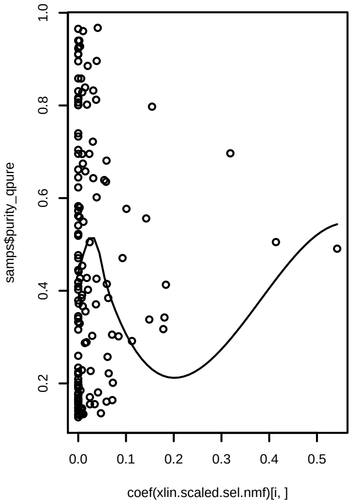


```r
par(mfrow = c(1, 1))
```









4.5 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
```

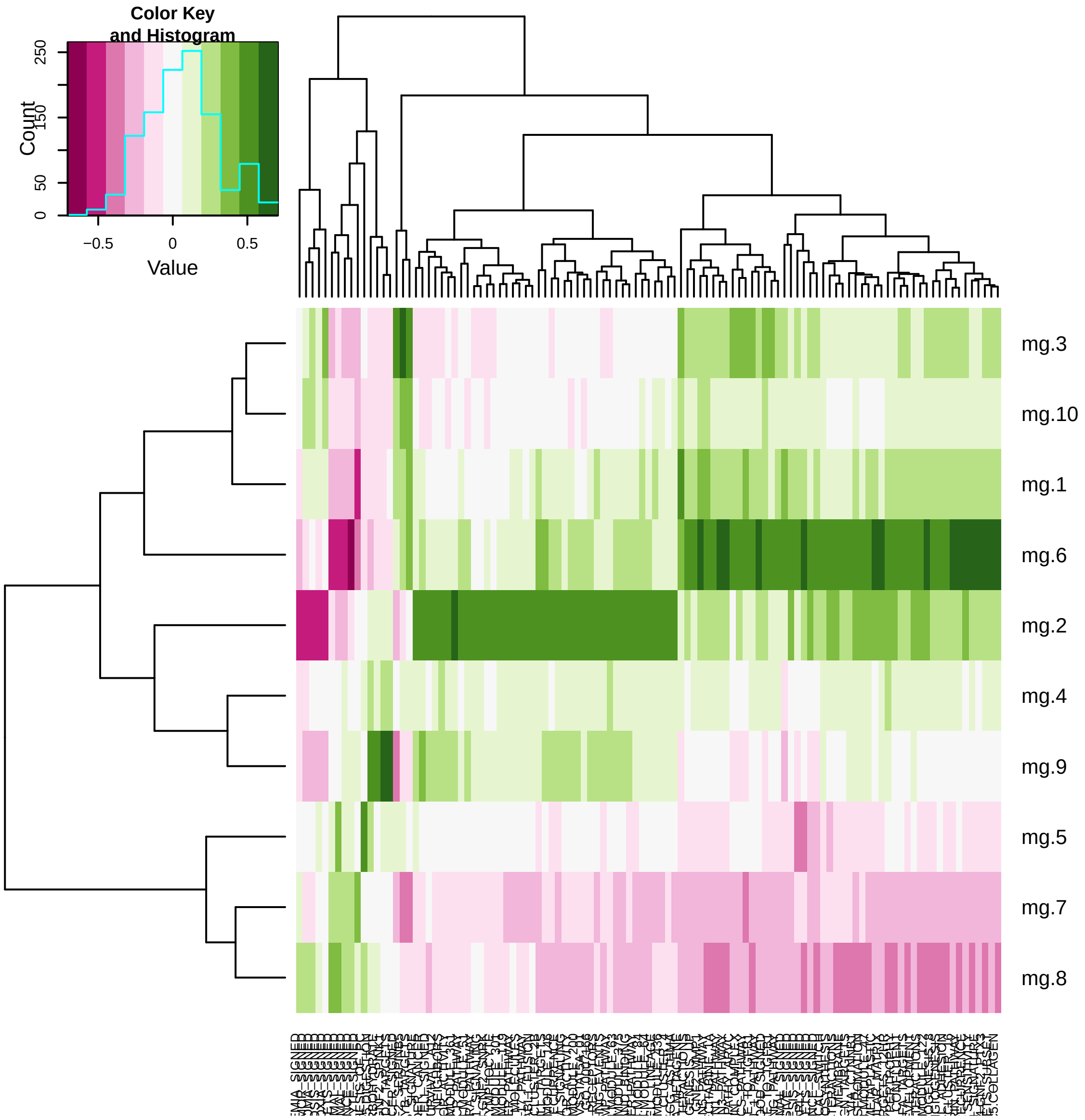
```
##          pure.p pure.p.BY pure.p.Holm  pure.s surv.diag_dsd.c
## mg.1  4.913e-01 1.000e+00   1.000e+00 -0.6882           4.835
## mg.2  1.182e-12 2.024e-10   4.730e-11 -7.1074          -5.296
## mg.3  2.304e-01 1.000e+00   1.000e+00 -1.1993           8.991
## mg.4  1.739e-04 4.959e-03   6.085e-03 -3.7542           4.661
## mg.5  5.158e-01 1.000e+00   1.000e+00  0.6498          -7.270
## mg.6  1.171e-03 1.996e-02   3.748e-02 -3.2458           5.390
## mg.7  2.067e-06 1.769e-04   8.062e-05  4.7467         -15.638
## mg.8  1.283e-03 1.996e-02   3.848e-02  3.2198          -4.273
## mg.9  7.784e-05 3.330e-03   2.880e-03 -3.9510          -1.440
## mg.10 7.702e-01 1.000e+00   1.000e+00  0.2922          10.853
##       surv.diag_dsd.p surv.diag_dsd.p.BY surv.diag_dsd.p.Holm
## mg.1        1.067e-03          0.0199573            0.0352207
## mg.2        1.244e-02          0.1182876            0.2861433
## mg.3        4.512e-06          0.0002574            0.0001715
## mg.4        4.783e-02          0.3274037            0.8026363
## mg.5        1.083e-04          0.0037073            0.0038992
## mg.6        3.259e-03          0.0371818            0.0847304
## mg.7        2.725e-03          0.0333167            0.0735865
## mg.8        2.484e-01          1.0000000            1.0000000
## mg.9        3.395e-01          1.0000000            1.0000000
## mg.10       1.195e-03          0.0199573            0.0374837
##       surv.diag_rec.c surv.diag_rec.p surv.diag_rec.p.BY
## mg.1            4.386       0.0017274            0.02464
## mg.2           -3.735       0.0472139            0.32740
## mg.3            6.614       0.0004398            0.01075
## mg.4            5.247       0.0218747            0.18718
## mg.5           -3.431       0.0340566            0.26493
## mg.6            3.829       0.0313128            0.25519
## mg.7          -11.050       0.0145725            0.13126
## mg.8           -3.745       0.3121647            1.00000
## mg.9           -2.315       0.1121060            0.66159
## mg.10          10.069       0.0021595            0.02843
##       surv.diag_rec.p.Holm surv.recr_dsd.c surv.recr_dsd.p
## mg.1               0.05009           3.049        0.064311
## mg.2               0.80264          -4.410        0.073554
## mg.3               0.01495           5.845        0.004374
## mg.4               0.45937           2.707        0.276281
## mg.5               0.64707          -5.666        0.005030
## mg.6               0.62626           3.941        0.042584
## mg.7               0.32059         -10.769        0.093131
## mg.8               1.00000          -4.601        0.239518
## mg.9               1.00000           0.552        0.738863
## mg.10              0.06047           4.358        0.258079
##       surv.recr_dsd.p.BY surv.recr_dsd.p.Holm
## mg.1             0.42332               0.9647
## mg.2             0.46623               1.0000
```

	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	1.0000	-0.6882	4.835	0.0352	4.386	0.0501	3.049	0.9647
mg.2	0.0000	-7.1074	-5.296	0.2861	-3.735	0.8026	-4.410	1.0000
mg.3	1.0000	-1.1993	8.991	0.0002	6.614	0.0150	5.845	0.1094
mg.4	0.0061	-3.7542	4.661	0.8026	5.247	0.4594	2.707	1.0000
mg.5	1.0000	0.6498	-7.270	0.0039	-3.431	0.6471	-5.666	0.1207
mg.6	0.0375	-3.2458	5.390	0.0847	3.829	0.6263	3.941	0.7665
mg.7	0.0001	4.7467	-15.638	0.0736	-11.050	0.3206	-10.769	1.0000
mg.8	0.0385	3.2198	-4.273	1.0000	-3.745	1.0000	-4.601	1.0000
mg.9	0.0029	-3.9510	-1.440	1.0000	-2.315	1.0000	0.552	1.0000
mg.10	1.0000	0.2922	10.853	0.0375	10.069	0.0605	4.358	1.0000

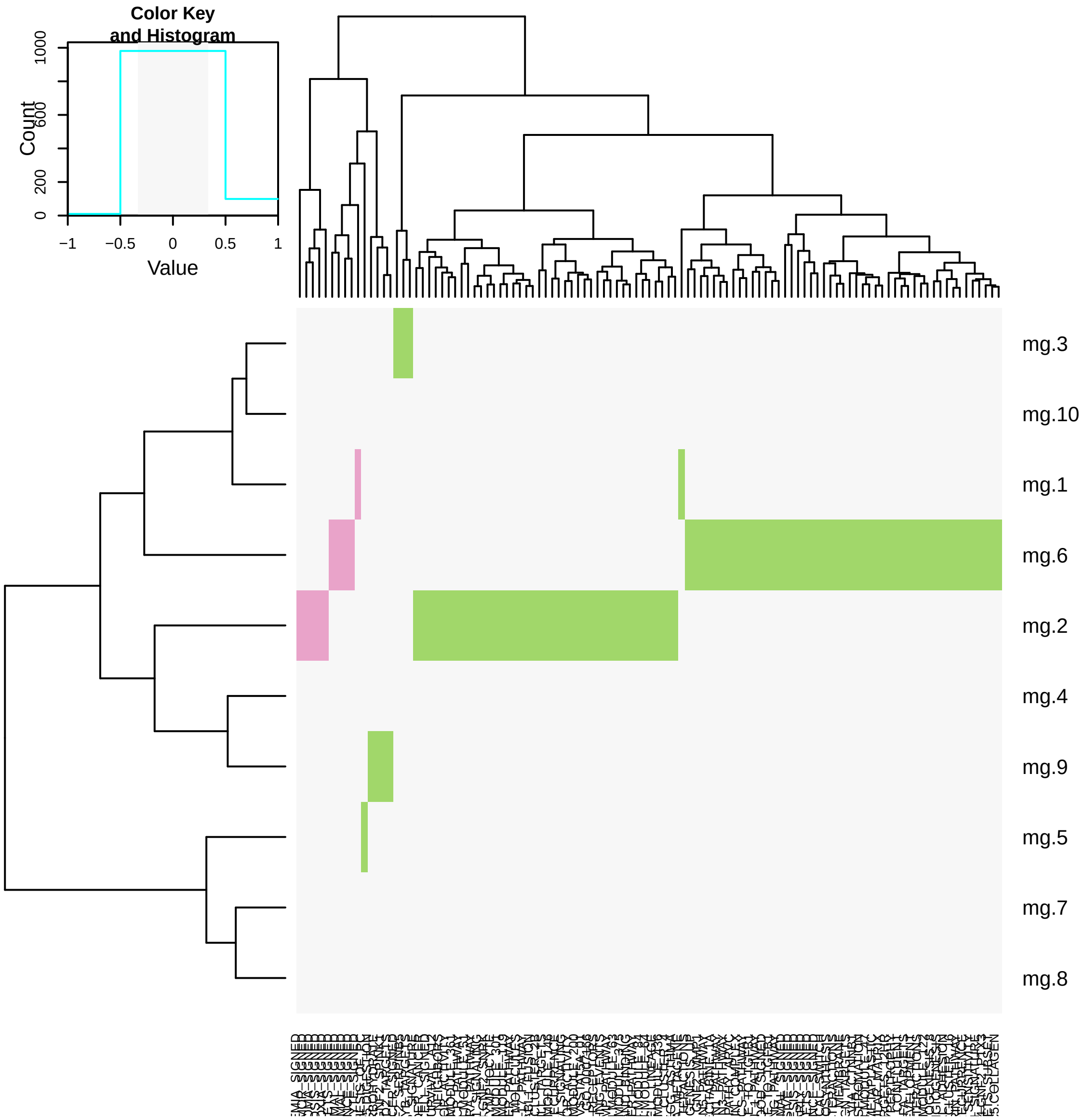
```
## mg.3                 0.04679                 0.1094
## mg.4                 1.00000                 1.0000
## mg.5                 0.05064                 0.1207
## mg.6                 0.31686                 0.7665
## mg.7                 0.56924                 1.0000
## mg.8                 1.00000                 1.0000
## mg.9                 1.00000                 1.0000
## mg.10                1.00000                 1.0000
```

4.6 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
```

```
print(asreg.result)

## Error in print(asreg.result):  error in evaluating the argument ’x’ in selecting a method
for function ’print’:  Error:  object ’asreg.result’ not found

coef(asreg.result)

## Error in coef(asreg.result):  error in evaluating the argument ’object’ in selecting a method
for function ’coef’:  Error:  object ’asreg.result’ not found

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

## Error in summary(asreg.result@objects[[1]]):  error in evaluating the argument ’object’
in selecting a method for function ’summary’:  Error:  object ’asreg.result’ not found
```

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

## Error in plot(asreg.result, type = "p"):  error in evaluating the argument ’x’ in selecting
a method for function ’plot’:  Error:  object ’asreg.result’ not found

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

## Error in plot(asreg.result, type = "s"):  error in evaluating the argument ’x’ in selecting
a method for function ’plot’:  Error:  object ’asreg.result’ not found

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

## Error in plot(asreg.result, type = "w"):  error in evaluating the argument ’x’ in selecting
a method for function ’plot’:  Error:  object ’asreg.result’ not found
```

```
glmnet.coef.1se

## Error in eval(expr, envir, enclos):  object ’glmnet.coef.1se’ not found

glmnet.coef.min

## Error in eval(expr, envir, enclos):  object ’glmnet.coef.min’ not found
```

```
plot(glmnet.fit.cv)

## Error in plot(glmnet.fit.cv):  error in evaluating the argument ’x’ in selecting a method
for function ’plot’:  Error:  object ’glmnet.fit.cv’ not found

plot(glmnet.fit.cv$glmnet.fit, label = TRUE)

## Error in plot(glmnet.fit.cv$glmnet.fit, label = TRUE): error in evaluating the argument
’x’ in selecting a method for function ’plot’:  Error:  object ’glmnet.fit.cv’ not found

abline(v = sum(abs(glmnet.coef.1se)))

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  object ’glmnet.coef.1se’
not found

abline(v = sum(abs(glmnet.coef.min)))

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  object ’glmnet.coef.min’
not found
```

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

## Error in eval(expr, envir, enclos):  object ’adaglmnet.coef.1se’ not found

adaglmnet.coef.min/adaglmnet.weights

## Error in eval(expr, envir, enclos):  object ’adaglmnet.coef.min’ not found
```

```
plot(adaglmnet.fit.cv)

## Error in plot(adaglmnet.fit.cv):  error in evaluating the argument ’x’ in selecting a method
for function ’plot’:  Error:  object ’adaglmnet.fit.cv’ not found

plot(adaglmnet.fit.cv$glmnet.fit, label = TRUE)

## Error in plot(adaglmnet.fit.cv$glmnet.fit, label = TRUE): error in evaluating the argument
’x’ in selecting a method for function ’plot’:  Error:  object ’adaglmnet.fit.cv’ not found

abline(v = sum(abs(adaglmnet.coef.1se)))

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  object ’adaglmnet.coef.1se’
not found

abline(v = sum(abs(adaglmnet.coef.min)))

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  object ’adaglmnet.coef.min’
not found
```

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

sessionInfo()

## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_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_0.20.5
## [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
```