This vignette must be run after the viz14 vignette.

Exercises 1

1. Obtain a list of names of putatively bound genes for the transcription factors ACE2, SWI5, SWI6, SWI4, MBP1, FKH1, FKH2, NDD1, MCM1.

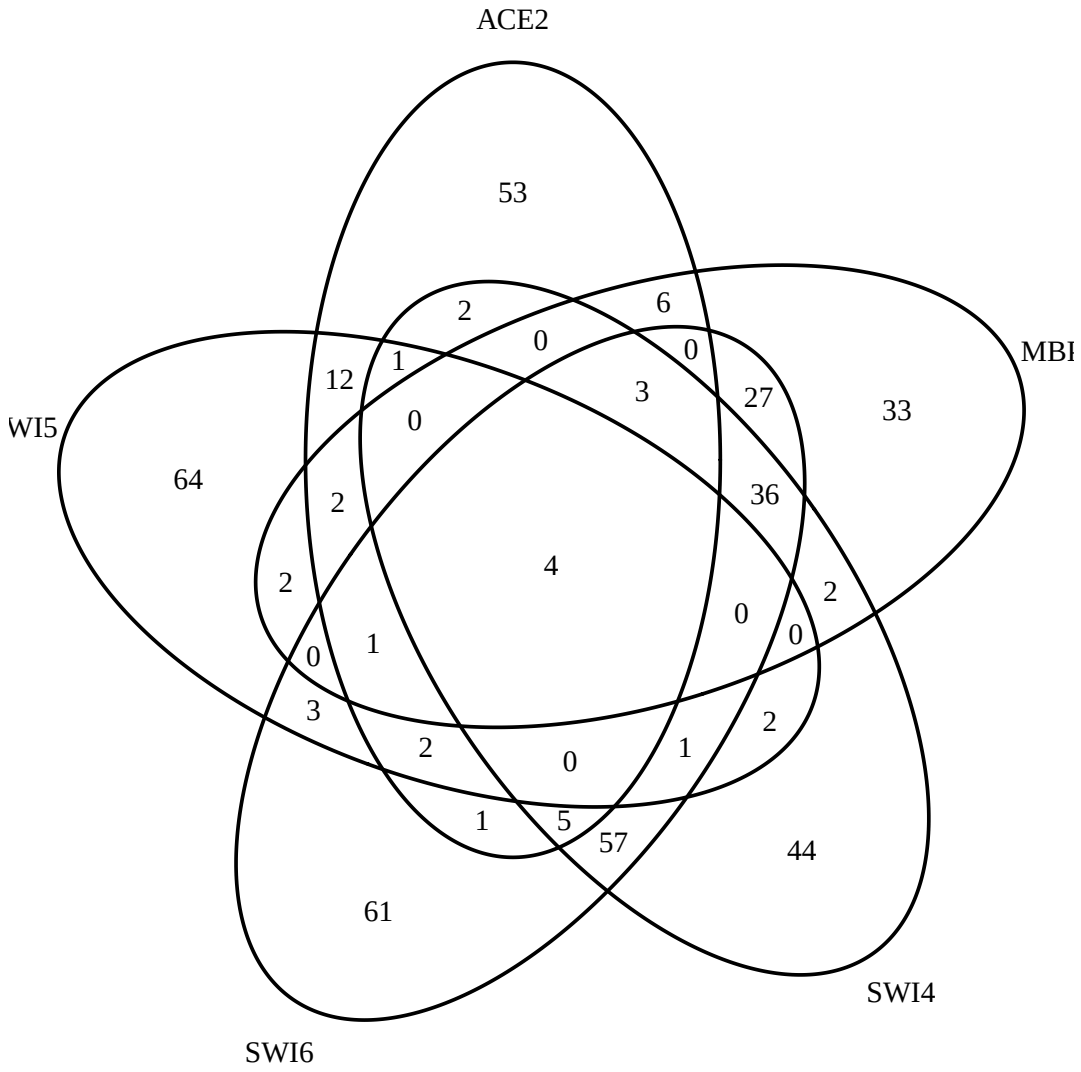
```
> library(viz14)
> facs = c("ACE2", "SWI5", "SWI6", "SWI4", "MBP1", "FKH1", "FKH2", "NDD1",
+     "MCM1")
> bg = lapply(facs, function(x) boundGenes(makebs(x)))
```

Selections from the lists of putatively bound genes:

```
> names(bg) = facs
> sapply(bg, length)
ACE2 SWI5 SWI6 SWI4 MBP1 FKH1 FKH2 NDD1 MCM1
  92   94  201  157  116  157  129  124  104
> sapply(bg, head)
     ACE2      SWI5      SWI6      SWI4        MBP1      FKH1      FKH2      NDD1
[1,] "YDR230W" "YLR011W" "YCR065W" "YNR044W"   "YDL101C" "YOR346W" "YGL116W" "YIL158W"
[2,] "YHR143W" "YLR013W" "YCR064C" "YMR307W"   "YPR075C" "YDR130C" "YIL158W" "YGL116W"
[3,] "YBR158W" "YLR012C" "YNL289W" "YMR306C-A" "YCR065W" "YKR055W" "YLR131C" "YPR119W"
[4,] "YFR017C" "YKL164C" "YNR044W" "YCR065W"   "YCR064C" "YKR054C" "YKR074W" "YML053C"
[5,] "YER125W" "YPL158C" "YBR265W" "YCR064C"   "YGR109C" "YFR004W" "YBR139W" "YML052W"
[6,] "YER124C" "YPL157W" "YBR264C" "YNL289W"   "YLR112W" "YFR003C" "YBR138C" "YJR092W"
     MCM1
[1,] "YIL158W"
[2,] "YLL031C"
[3,] "YGL116W"
[4,] "YLR131C"
[5,] "YHR152W"
[6,] "YHR151C"
```

2. Using the VennDiagram package, assess overlap between five of these bound sets.

```
> library(VennDiagram)
> v1 = venn.diagram(bg[1:5], filename = NULL)
> grid.draw(v1)
```



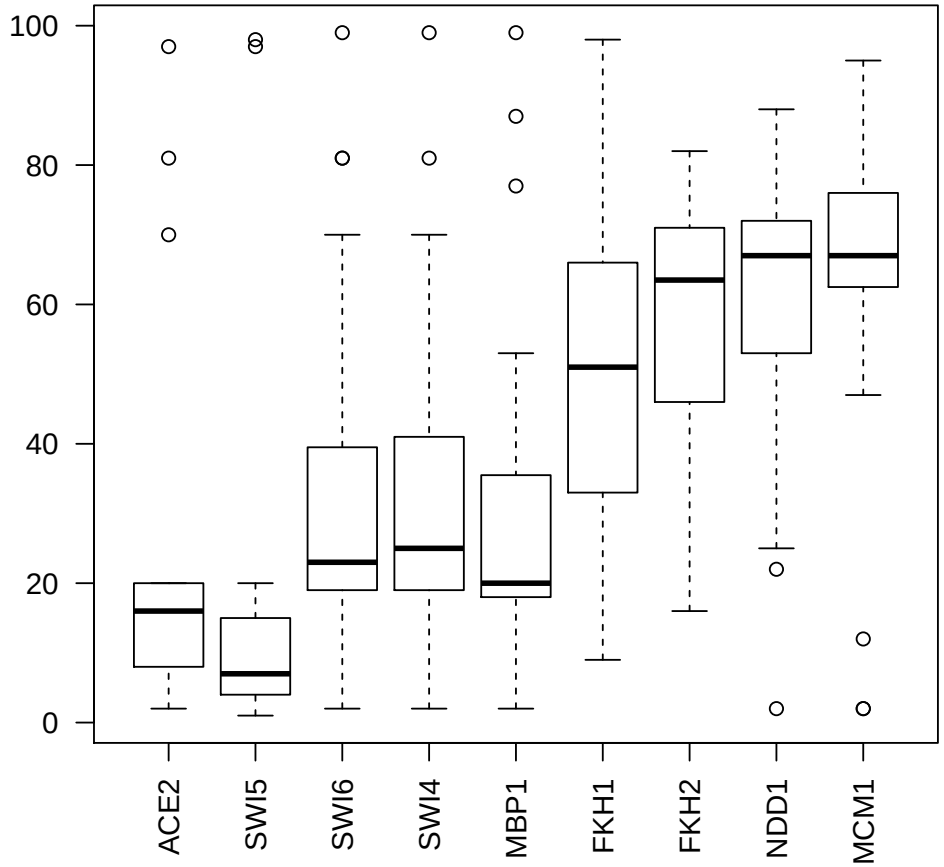
3. Check the documentation for `calout.detect`. How would you apply the standard boxplot outlier rules to define binding events?

```
> ac2s = makebs("ACE2")
> bac2s = boundGenes(ac2s, method = "boxplot", scale = function(...) 1.5)
> length(bac2s)
[1] 119
> bac2g = boundGenes(ac2s)
> length(bac2g)
[1] 92
```

Exercises 2

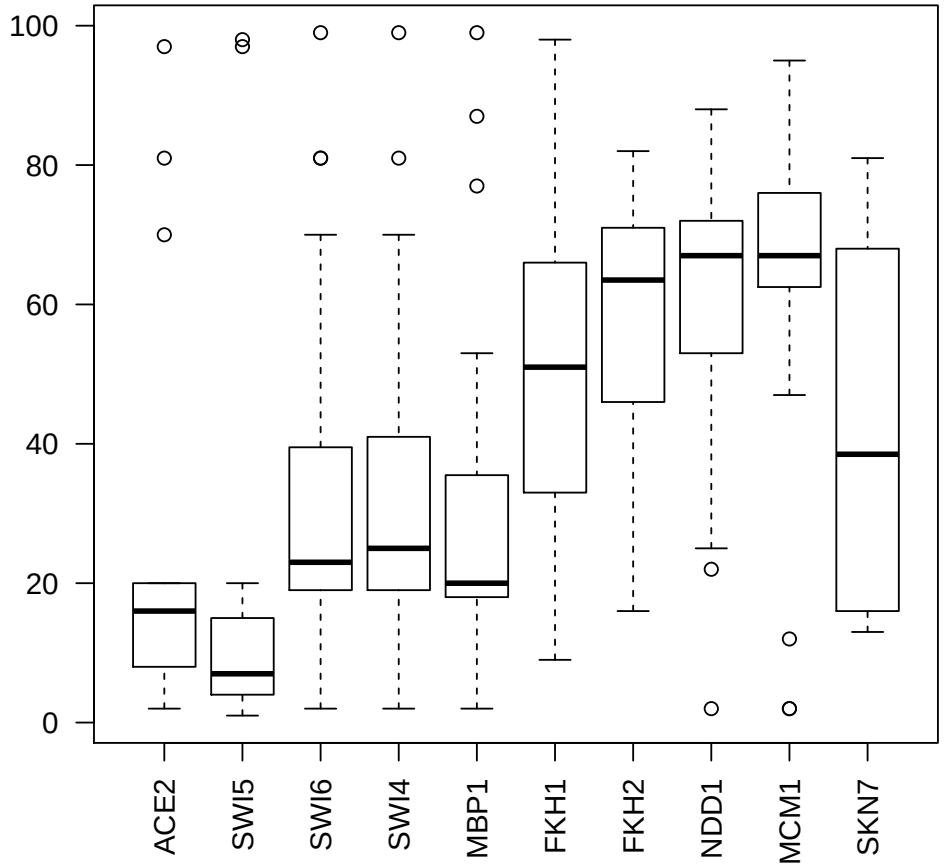
1. Interpret the following plot:

```
> names(bgrps) = facs
> facs
[1] "ACE2" "SWI5" "SWI6" "SWI4" "MBP1" "FKH1" "FKH2" "NDD1" "MCM1"
> sapply(bgrps, function(x) length(na.omit(x)))
ACE2 SWI5 SWI6 SWI4 MBP1 FKH1 FKH2 NDD1 MCM1
  13   12   64   54   40   26   40   41   23
> boxplot(bgrps, las = 2)
```



2. The TF SKN7 was omitted from `facs`. Introduce the timings for gene group associated with SKN7 into the boxplot. Interpret the new display in terms of potential combinatorial relations among TFs? How do you reorder the plot for clearer ingestion?

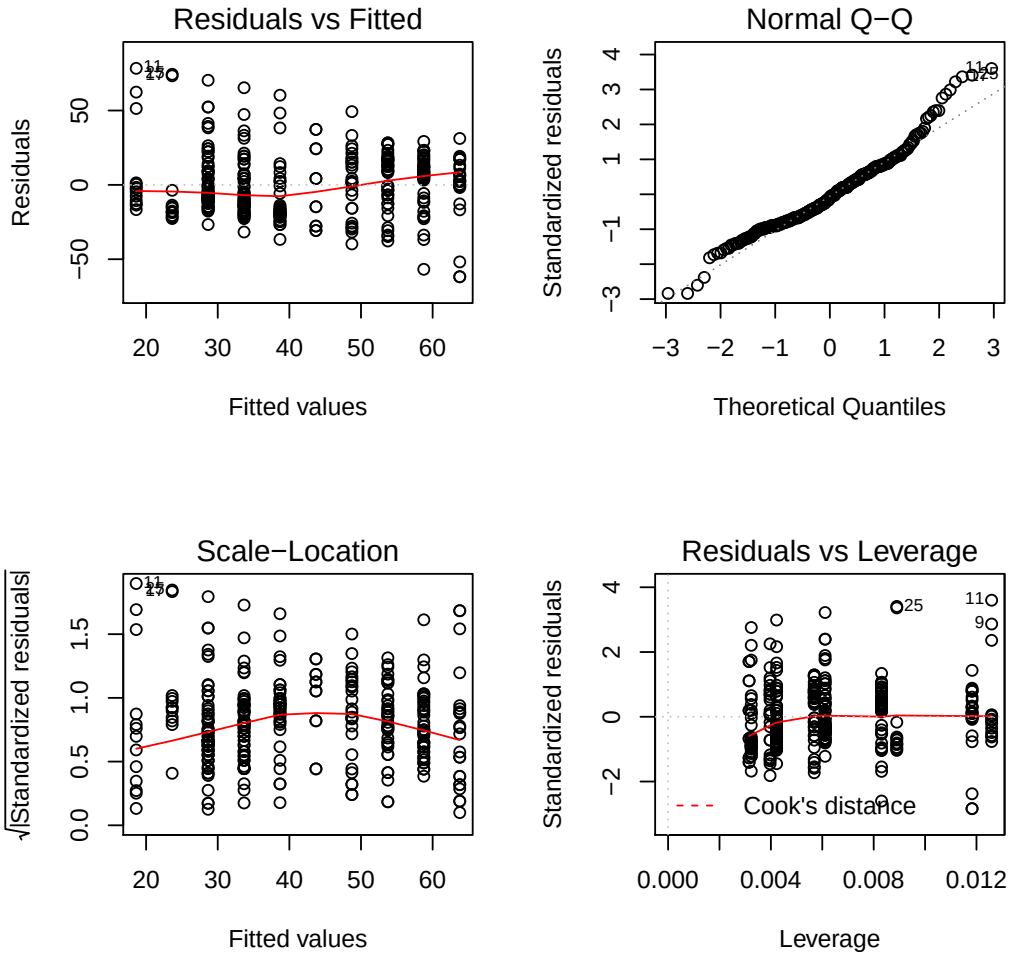
```
> bsk = boundGenes(makebs("SKN7"))
> bgrps[[length(bgrps) + 1]] = trigFits[intersect(bsk, rownames(trigFits)),
+     "dtf"]
> names(bgrps)[length(bgrps)] = "SKN7"
> boxplot(bgrps, las = 2)
```



3. Your paper on this finding has been rejected with the comment that the figure does not include any appraisal of statistical significance. How do you respond?
   The following code reshapes the percent cell-cycle to peak data and uses nonparametric test of common location followed by parametric test for trend among a speculatively ordered sequence of TFs. Both provide an approach to statistical appraisal, but neither is completely satisfactory.

```
> bgn = lapply(bgrps, na.omit)
> nb = names(bgn)
> alln = rep(nb, sapply(bgn, length))
> mydf = data.frame(ppkt = unlist(bgn), fac = factor(alln, levels = c(facs[1:5],
+     "SKN7", facs[-c(1:5)])))
> kruskal.test(ppkt ~ fac, data = mydf)

        Kruskal-Wallis rank sum test

data:  ppkt by fac
Kruskal-Wallis chi-squared = 99.2456, df = 9, p-value < 2.2e-16
> summary(tlm <- lm(ppkt ~ as.numeric(fac), data = mydf))
Call:
lm(formula = ppkt ~ as.numeric(fac), data = mydf)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-61.782 -15.636  -1.782  13.236  78.383

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      13.5992     2.8717   4.736 3.28e-06 ***
as.numeric(fac)   5.0183     0.4651  10.789  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.91 on 323 degrees of freedom
Multiple R-squared:  0.2649,        Adjusted R-squared:  0.2626
F-statistic: 116.4 on 1 and 323 DF,  p-value: < 2.2e-16
> oldp = par(no.readonly = TRUE)
> par(mfrow = c(2, 2))
> plot(tlm)
> par(oldp)
```

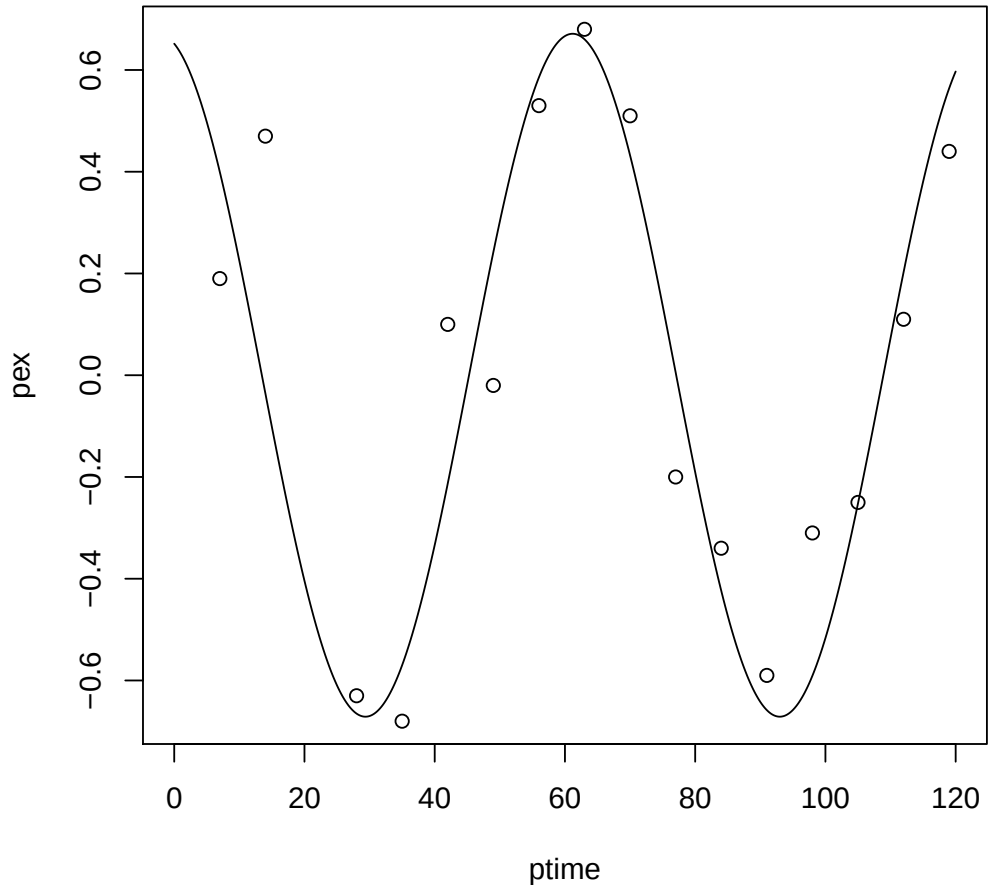


1 Modeling transcription regulation in the yeast cell cycle

Exercises 3

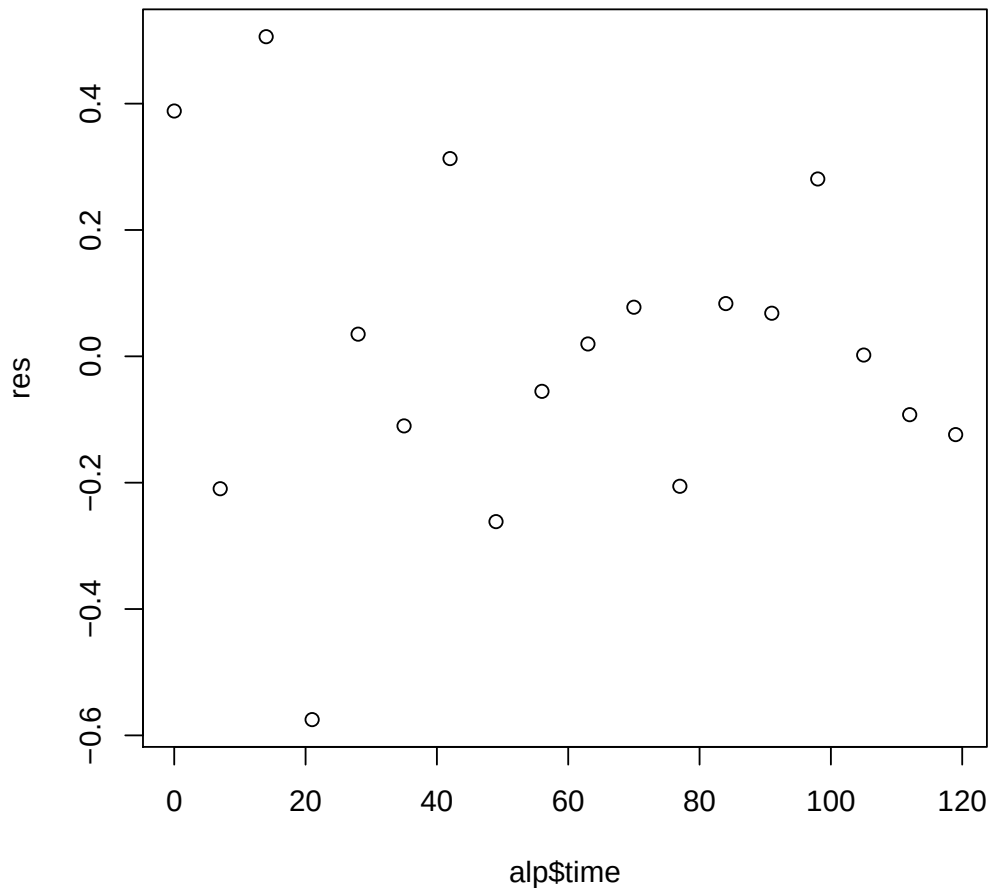
1. Plot the predictions from the nonlinear regression on a fine grid of points from 0 to 120 minutes. Use type = 'l'.
2. Superimpose the data on these predictions.

```
> ptime = seq(0, 120, 0.1)
> pex = predict(m1, newdata = list(time = ptime))
> plot(pex ~ ptime, type = "l")
> points(alp$time, yal040c)
```



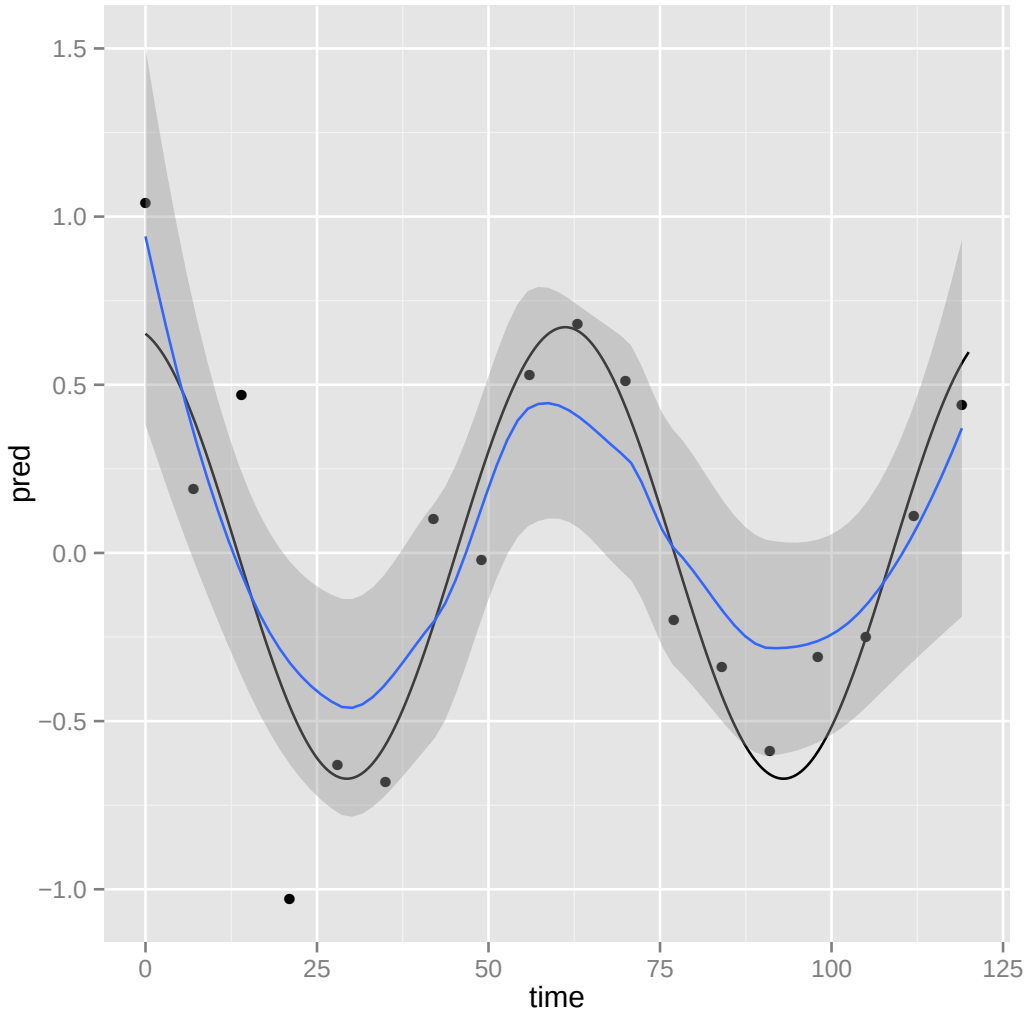
3. Plot the residuals from the model over time.

```
> res = resid(m1)
> plot(res ~ alp$time)
```



4. If you did not use *ggplot2* for these visualizations, please do so. If you did use *ggplot2*, use the standard graphics.
5. Enhance the *ggplot2*-based version with a nonparametric model including pointwise standard errors. Interpret.

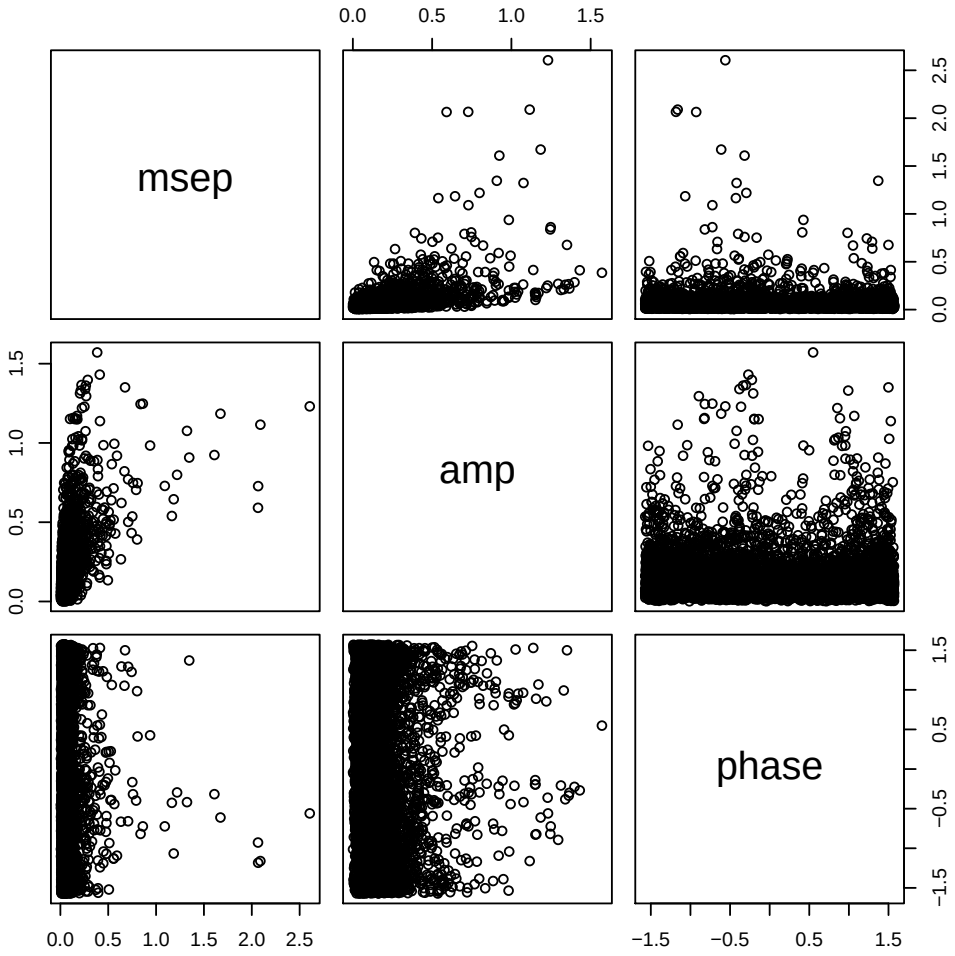
```
> library(ggplot2)
> prdf = data.frame(pred = pex, time = ptime)
> g1 = ggplot(prdf, aes(x = time, y = pred))
> print(g1 + geom_line() + geom_point(data = df, aes(y = yal040c, x = time)) +
+     stat_smooth(data = df, aes(y = yal040c, x = time)))
```



Exercises 4

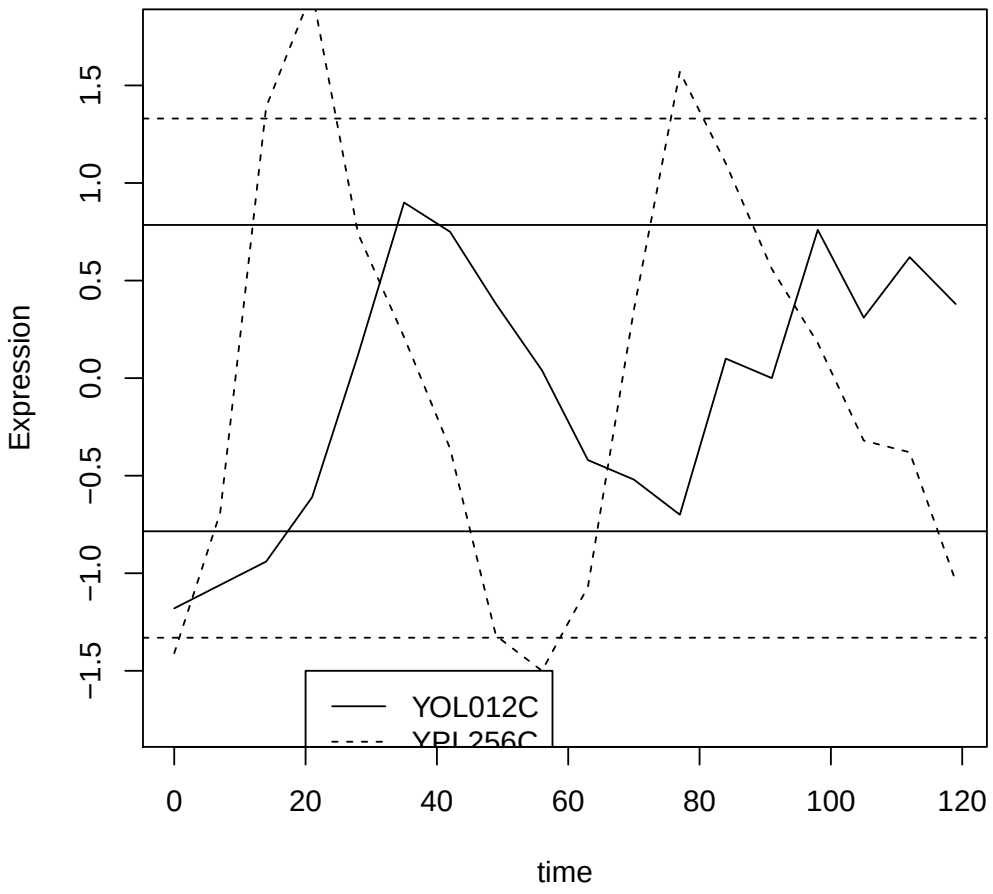
1. Interpret `pairs(outs)`. Choose boundaries on msep and amp that identify genes with robust cyclic transcription pattern, and subset `outs` to this set of genes. Is the overlap with Spellman's orf800 as you would expect?

```
> pairs(outs)
> dou = data.frame(outs)
> limdou <- dou[which(dou$msep < 0.25 & dou$amp > 0.75), ]
> mean(rownames(limdou) %in% orf800)
[1] 1
```



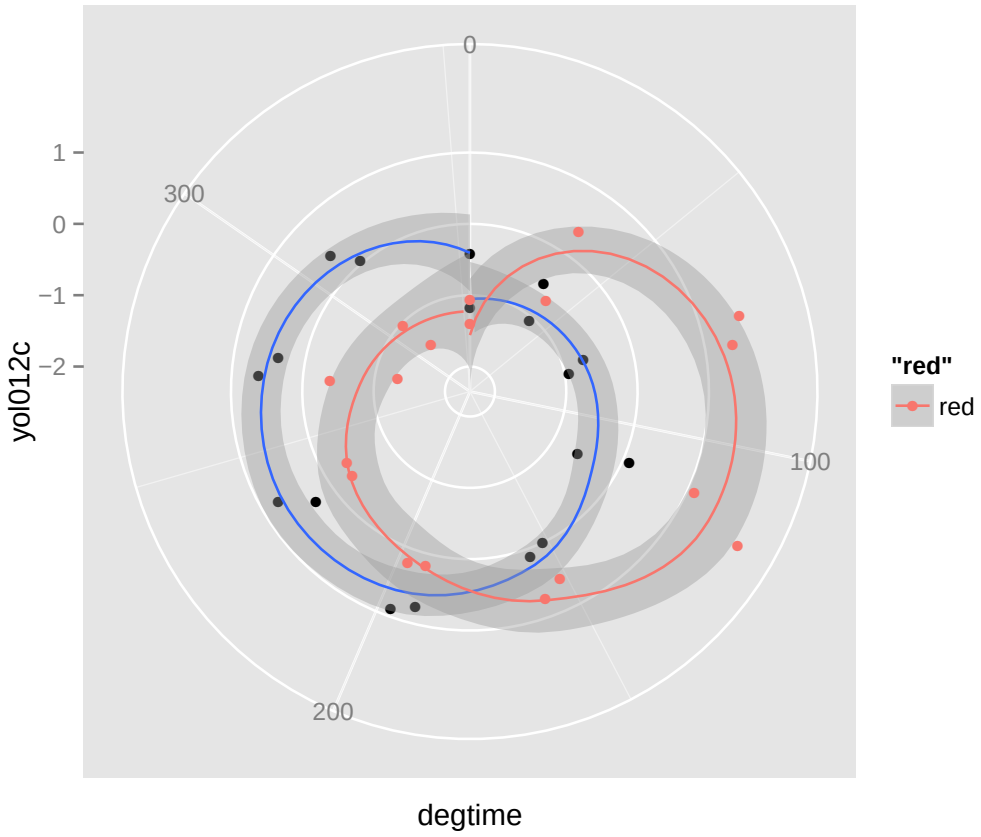
2. Find a pair of genes with estimated phase values near -1.0 and 1.0 respectively. Plot the expression trajectories, superimposed, in the (expression,time) plane. Justify and suitably display the estimated values of amp for these genes.

```
> plot(exprs(alp)["YOL012C", ] ~ alp$time, type = "l", ylim = c(-1.75,
+     1.75), ylab = "Expression", xlab = "time")
> lines(exprs(alp)["YPL256C", ] ~ alp$time, lty = 2)
> ampsP = limdou["YPL256C", "amp"]
> abline(h = ampsP, lty = 2)
> abline(h = -ampsP, lty = 2)
> ampsO = limdou["YOL012C", "amp"]
> abline(h = ampsO, lty = 1)
> abline(h = -ampsO, lty = 1)
> legend(20, -1.5, lty = c(1, 2), legend = c("YOL012C", "YPL256C"))
```



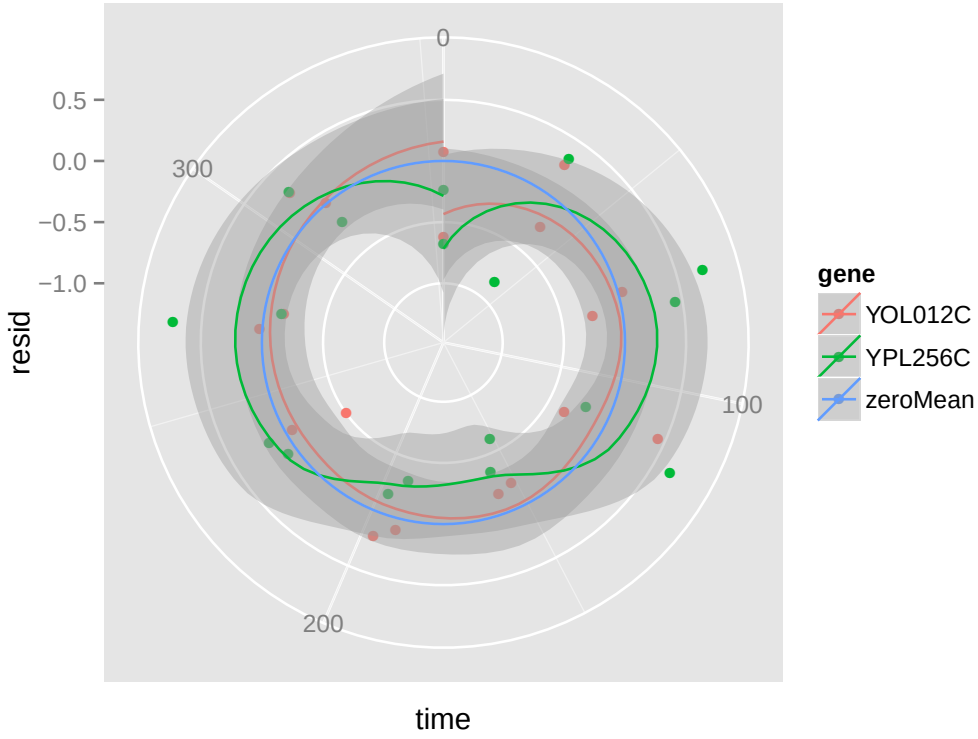
3. Following the *ggplot2* code patterns of the lecture, plot these trajectories in polar coordinates.

```
> degdf = function(genename, es, period = 64) {
+     stopifnot("time" %in% names(pData(es)))
+     ex = exprs(es)[genename, ]
+     et = es$time
+     degtime = 360 * (df$time%%period)/period
+     ndf = data.frame(time = et, degtime = degtime)
+     ndf[[tolower(substitute(genename))]] = exprs(es)[genename, ]
+     ndf
+ }
> d1 = degdf("YOL012C", alp)
> library(ggplot2)
> m012c = ggplot(d1, aes(y = yol012c, x = degtime)) + geom_point()
> print(m012c + coord_polar() + stat_smooth() + stat_smooth(data = degdf("YPL256C",
+     alp), aes(y = ypl256c, x = degtime, colour = "red")) + geom_point(data = degdf("YPL256C",
+     alp), aes(y = ypl256c, x = degtime, colour = "red")))
```



4. Residual analysis: Obtain the residuals for trigonmetric fits to YOL012C and YPL256C and display in polar coordinates. Interpret.

```
> exprs(alp)["YOL012C", 2] = -1.06
> r1 = resid(gettrm("YOL012C", alp))
> r2 = resid(gettrm("YPL256C", alp))
> dtime = 360 * (alp$time%%64)/64
> d1 = data.frame(resid = r1, gene = "YOL012C", time = dtime)
> d2 = data.frame(resid = r2, gene = "YPL256C", time = dtime)
> dres = rbind(d1, d2)
> print(ggplot(dres, aes(x = time, y = resid, colour = gene)) + geom_point() +
+     stat_smooth() + coord_polar() + stat_abline(slope = 0, intercept = 0,
+     aes(colour = "zeroMean")))
```



2 RNA-seq application: hnRNP C and Alu exon suppression

2.1 Exercises 5

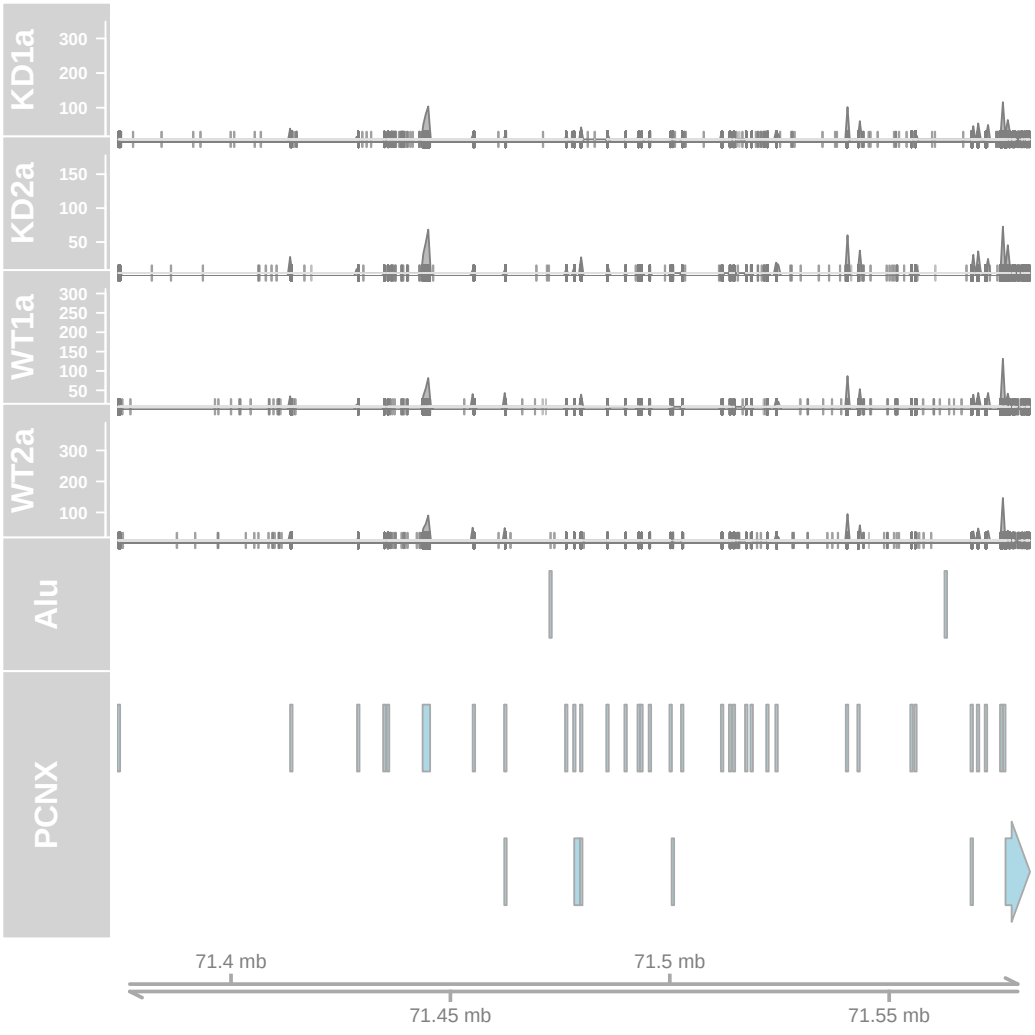
1. Show that, on chr14, genes harboring Alu elements are at least 5 times more likely to be identified as trait-associated in GWAS as those that do not harbor Alu elements.

```
> mean(symovalu %in% mapped14)/mean(symnoalu %in% mapped14)
[1] 5.689307
```

2. Transform the code for ADSSL1 into a function that will accept any gene symbol, and use it to investigate CGRRF1, PCNX (both mentioned in supplement to Zarnack paper) and other genes, for example, those harboring Alu and mapped to trait variation in GWAS.

```
> chkHN = function(GENENAME) {
+     library(GenomicFiles)
+     library(Gviz)
+     fn = dir(system.file("extdata", package = "RNAseqData.HNRNPC.bam.chr14"),
+         full = TRUE, patt = "bam$")
+     bfv = BamFileViews(fn)
+     STACKTYPE = "hide"
```

```
+     kd1a = AlignmentsTrack(path(fileList(bfv)[[1]]), isPaired = TRUE,
+         name = "KD1a", chromosome = "chr14", stacking = STACKTYPE)
+     wt1a = AlignmentsTrack(path(fileList(bfv)[[5]]), isPaired = TRUE,
+         name = "WT1a", chromosome = "chr14", stacking = STACKTYPE)
+     kd2a = AlignmentsTrack(path(fileList(bfv)[[3]]), isPaired = TRUE,
+         name = "KD2a", chromosome = "chr14", stacking = STACKTYPE)
+     wt2a = AlignmentsTrack(path(fileList(bfv)[[7]]), isPaired = TRUE,
+         name = "WT2a", chromosome = "chr14", stacking = STACKTYPE)
+     gm = genemodel(GENENAME)
+     rgm = range(gm)
+     at = AnnotationTrack(gm, genome = "hg19", name = GENENAME)
+     gt = GenomeAxisTrack(genome = "hg19")
+     data(nestrep)
+     rep14 = nestrep[which(seqnames(nestrep) == "chr14")]
+     alu14 = rep14[grep("^Alu", rep14$name)]
+     alut = AnnotationTrack(alu14, genome = "hg19", name = "Alu")
+     ps.options(font = "sans")
+     plotTracks(list(kd1a, kd2a, wt1a, wt2a, alut, at, gt), from = start(rgm),
+         to = end(rgm))
+ }
> chkHN("PCNX")
```



```
> sessionInfo()
R version 3.1.0 (2014-04-10)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] C

attached base packages:
 [1] grid      parallel  stats     graphics  grDevices datasets  utils     tools
 [9] methods   base

other attached packages:
 [1] Gviz_1.9.8                              GenomicFiles_1.1.11
 [3] BiocParallel_0.7.2                      rtracklayer_1.25.11
 [5] Rsamtools_1.17.27                       gwascat_1.9.3
 [7] org.Hs.eg.db_2.14.0                     RSQLite_0.11.4
 [9] DBI_0.2-7                               TxDb.Hsapiens.UCSC.hg19.knownGene_2.14.0
[11] GenomicFeatures_1.17.11                 AnnotationDbi_1.27.8
[13] ggbio_1.13.7                            ggplot2_1.0.0
[15] GenomicRanges_1.17.18                   GenomeInfoDb_1.1.8
[17] yeastCC_1.5.0                           VennDiagram_1.6.5
[19] harbChIP_1.3.0                          Biostrings_2.33.10
[21] XVector_0.5.6                           Biobase_2.25.0
[23] IRanges_1.99.15                         S4Vectors_0.0.8
[25] BiocGenerics_0.11.2                     viz14_0.0.11
[27] weaver_1.31.0                           codetools_0.2-8
[29] digest_0.6.4                            BiocInstaller_1.15.5

loaded via a namespace (and not attached):
 [1] BBmisc_1.6                BSgenome_1.33.8           BatchJobs_1.2
 [4] BiocStyle_1.3.2           Formula_1.1-1             GGally_0.4.6
 [7] GenomicAlignments_1.1.14  Hmisc_3.14-4              MASS_7.3-33
[10] Matrix_1.1-4              OrganismDbi_1.7.2         R.methodsS3_1.6.1
[13] RBGL_1.41.0               RColorBrewer_1.0-5        RCurl_1.95-4.1
[16] RJSONIO_1.2-0.2           Rcpp_0.11.2               VariantAnnotation_1.11.12
[19] XML_3.98-1.1              assertthat_0.1            biomaRt_2.21.0
[22] biovizBase_1.13.7         bitops_1.0-6              brew_1.0-6
[25] caTools_1.17              cluster_1.15.2            colorspace_1.2-4
[28] dichromat_2.0-0           dplyr_0.2                 fail_1.2
[31] foreach_1.4.2             ggvis_0.2.0.99            graph_1.43.0
[34] gridExtra_0.9.1           gtable_0.1.2              htmltools_0.2.4
[37] httpuv_1.3.0              iterators_1.0.7           labeling_0.2
[40] lattice_0.20-29           latticeExtra_0.6-26       matrixStats_0.10.0
[43] munsell_0.4.2             parody_1.23.0             plyr_1.8.1
[46] proto_0.3-10              reshape_0.8.5             reshape2_1.4
[49] scales_0.2.4              sendmailR_1.1-2           shiny_0.10.0
[52] snpStats_1.15.0           splines_3.1.0             stats4_3.1.0
[55] stringr_0.6.2             survival_2.37-7           xtable_1.7-3
[58] zlibbioc_1.11.1
```