R implementation

Sleiman Bassim, PhD

April 8, 2016

Loaded functions:

```r
#source("/media/Data/Dropbox/humanR/01funcs.R")
rm(list=ls())
#setwd("/media/Data/Dropbox/humanR/PD/")
#setwd("~/Dropbox/humanR/PD/")
###load("PD.Rdata", .GlobalEnv)
#lsos(pat="")
```

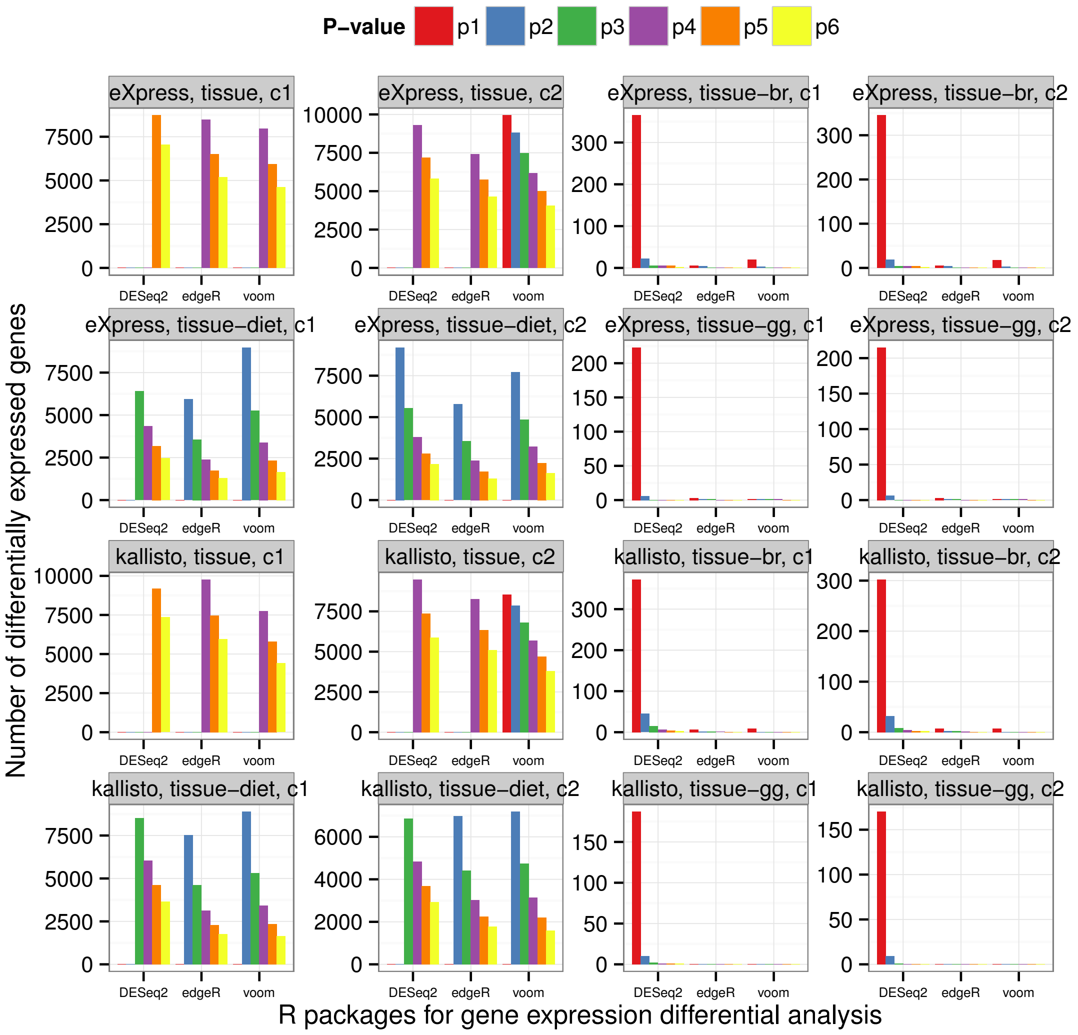
Load packages.

```r
pkgs <- c('xlsx','caret','leaps','glmnet','lattice',
          'latticeExtra', 'dplyr', 'tidyr')
lapply(pkgs, require, character.only = TRUE)
```

1 Differentially expressed genes

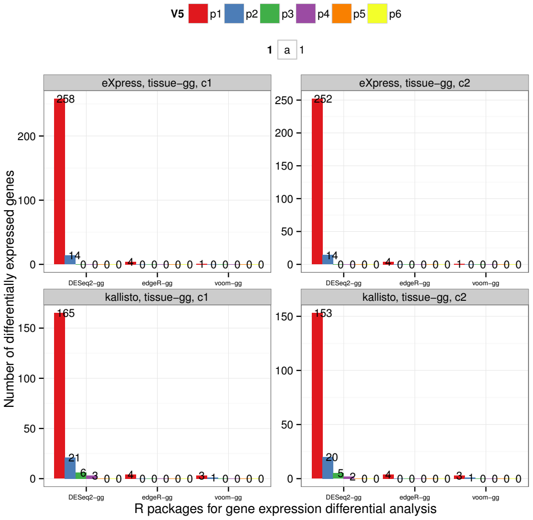
Differentially expressed genes are counted from mapping **both gills and ganglia** sequenced samples to reference transcriptome built from all samples.

```r
read.table("./data/summary.raw.all.txt") %>%
    ggplot(aes(
        x = V1,
        y = V8,
        fill = V6)) +
    theme_bw() +
    geom_bar(stat = "identity",
             position = "dodge") +
    facet_wrap(~ V4 + V5 + V7,
               ncol = 4,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6,
                      name = "P-value") +
    labs(x = "R packages for gene expression differential analysis",
         y = "Number of differentially expressed genes") +
    theme(legend.position = "top",
          axis.text.x = element_text(vjust = .5,
                                     size = 6))
```



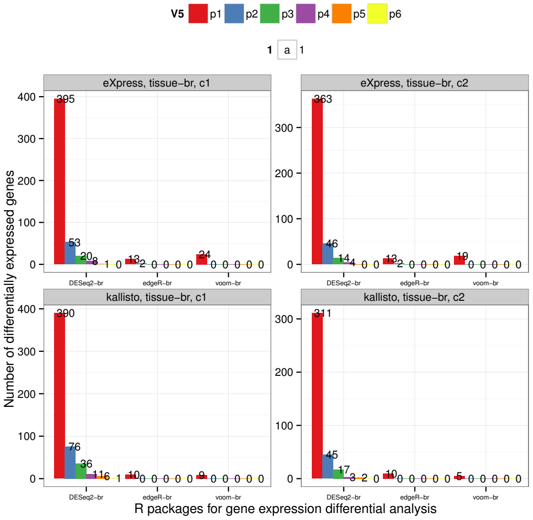
Differentially expressed genes are counted from mapping **gills** sequenced samples to reference transcriptome built from all samples.

```
read.table("./data/summary.gg.txt") %>%
ggplot(aes(
x = V2,
y = V8,
fill = V5)) +
geom_bar(stat = "identity",
         position = "dodge") +
geom_text(aes(x = V2,
              y = V8,
              ymax = V8,
              label = V8,
              size = 1,
              hjust = 0),
          position = position_dodge(width = 1)) +
facet_wrap(~ V3 + V4 + V6,
           ncol = 2,
           scales = "free") +
scale_fill_brewer(type = "qual", palette = 6) +
theme_bw() +
theme(legend.position = "top",
      axis.text.x = element_text(vjust = .5,
                                 size = 6)) +
labs(x = "R packages for gene expression differential analysis",
     y = "Number of differentially expressed genes")
```



Differentially expressed genes are counted from mapping **ganglia** sequenced samples to reference transcriptome built from all samples.

```
read.table("./data/summary.br.txt") %>%
    ggplot(aes(
    x = V2,
    y = V8,
    fill = V5)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    geom_text(aes(x = V2,
                  y = V8,
                  ymax = V8,
                  label = V8,
                  size = 1,
                  hjust = 0),
              position = position_dodge(width = 1)) +
    facet_wrap(~ V3 + V4 + V6,
               ncol = 2,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    theme_bw() +
    theme(legend.position = "top",
          axis.text.x = element_text(vjust = .5,
                                     size = 6)) +
    labs(x = "R packages for gene expression differential analysis",
         y = "Number of differentially expressed genes")
```

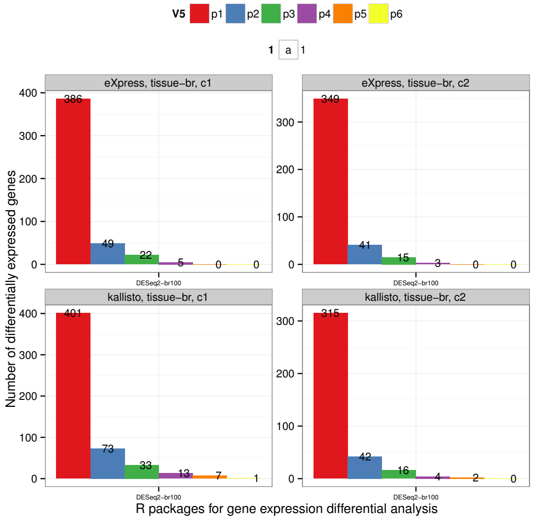


1.1 Increasing DEG by changing the trimming rates of raw reads

Getting gene expression by mapping the original raw reads **without trimming** to the **gills** de novo transcriptome.

↰ De novo assembly was carried out with trimmed reads though

```
read.table("./data/summary.br_35078.txt") %>%
    ggplot(aes(
    x = V2,
    y = V8,
    fill = V5)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    geom_text(aes(x = V2,
                  y = V8,
                  ymax = V8,
                  label = V8,
                  size = 1,
                  hjust = 0),
              position = position_dodge(width = 1)) +
    facet_wrap(~ V3 + V4 + V6,
               ncol = 2,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    theme_bw() +
    theme(legend.position = "top",
          axis.text.x = element_text(vjust = .5,
                                     size = 6)) +
    labs(x = "R packages for gene expression differential analysis",
         y = "Number of differentially expressed genes")
```

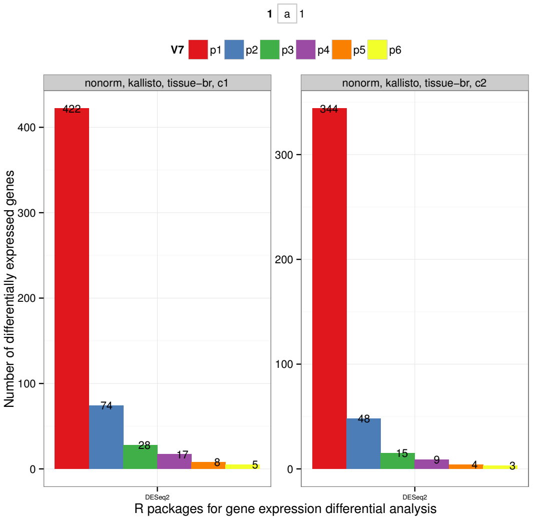


1.2 Increasing DEGs by changing the normalization strategy: Fast abundance quantification *kallisto*

The below graph shows the number of differentially expressed genes when raw reads were normalized **separately** for each biological sample.

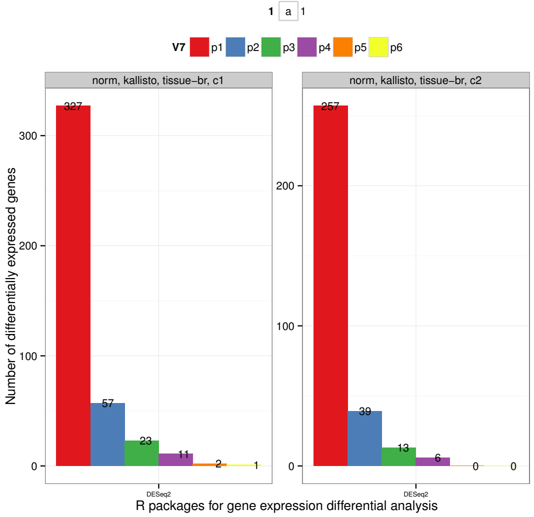
↰ All the analyses before were done on normalized reads by grouping all biological samples together

```
read.table("./data/summary.br.nonorm.txt") %>%
    ggplot(aes(
    x = V1,
    y = V10,
    fill = V7)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    geom_text(aes(x = V1,
                  y = V10,
                  ymax = V10,
                  label = V10,
                  size = 1,
                  hjust = 0),
              position = position_dodge(width = 1)) +
    facet_wrap(~ V4 + V5 + V6 + V8,
               ncol = 2,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    theme_bw() +
    theme(legend.position = "top",
          axis.text.x = element_text(vjust = .5,
                                     size = 6)) +
    labs(x = "R packages for gene expression differential analysis",
         y = "Number of differentially expressed genes")
```



The below graph shows the number of differentially expressed genes when raw reads were **NOT** normalized.

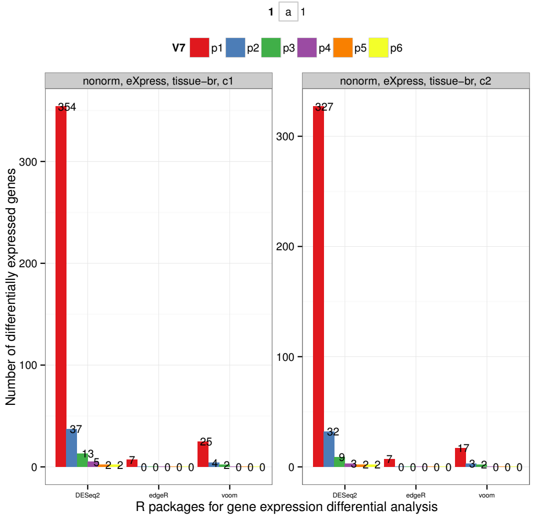
```
read.table("./data/summary.br.norm.txt") %>%
    ggplot(aes(
    x = V1,
    y = V10,
    fill = V7)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    geom_text(aes(x = V1,
                  y = V10,
                  ymax = V10,
                  label = V10,
                  size = 1,
                  hjust = 0),
              position = position_dodge(width = 1)) +
    facet_wrap(~ V4 + V5 + V6 + V8,
               ncol = 2,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    theme_bw() +
    theme(legend.position = "top",
          axis.text.x = element_text(vjust = .5,
                                     size = 6)) +
    labs(x = "R packages for gene expression differential analysis",
         y = "Number of differentially expressed genes")
```



1.3 Increasing DEGs by changing the normalization strategy: abundance quantification with alignment *express*

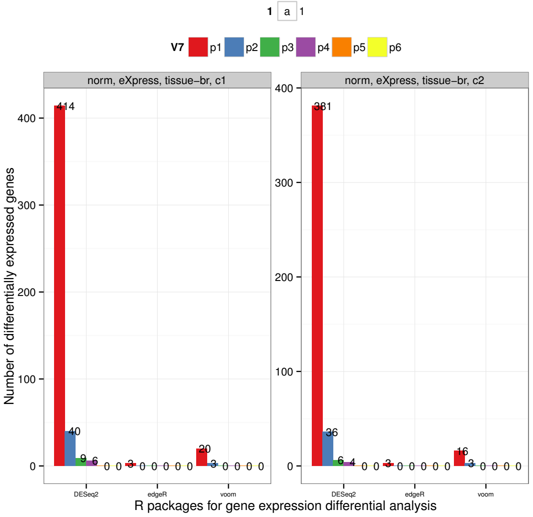
The R package *eXpress* is solely used to quantify the abundance of differentially expressed genes from **NOT normalized** reads and aligned with **Bowtie 1**.

```
read.table("./data/summary.br.nonorm.44062.txt") %>%
    ggplot(aes(
    x = V1,
    y = V10,
    fill = V7)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    geom_text(aes(x = V1,
                  y = V10,
                  ymax = V10,
                  label = V10,
                  size = 1,
                  hjust = 0),
              position = position_dodge(width = 1)) +
    facet_wrap(~ V4 + V5 + V6 + V8,
               ncol = 2,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    theme_bw() +
    theme(legend.position = "top",
          axis.text.x = element_text(vjust = .5,
                                     size = 6)) +
    labs(x = "R packages for gene expression differential analysis",
         y = "Number of differentially expressed genes")
```



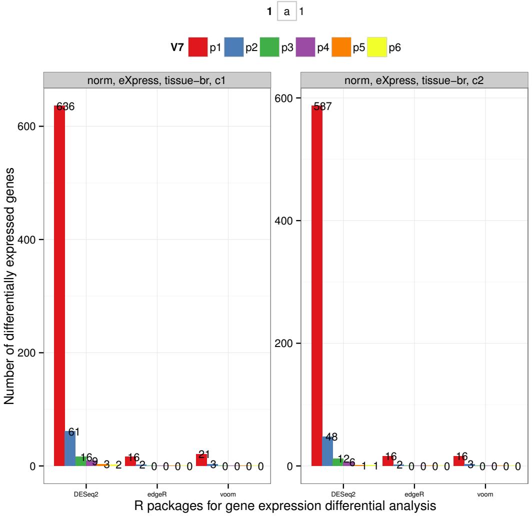
The R package *eXpress* is solely used to quantify the abundance of differentially expressed genes from **normalized** reads and aligned with **Bowtie 1**.

```
read.table("./data/summary.br.norm.44060.txt") %>%
ggplot(aes(
x = V1,
y = V10,
fill = V7)) +
geom_bar(stat = "identity",
         position = "dodge") +
geom_text(aes(x = V1,
              y = V10,
              ymax = V10,
              label = V10,
              size = 1,
              hjust = 0),
          position = position_dodge(width = 1)) +
facet_wrap(~ V4 + V5 + V6 + V8,
           ncol = 2,
           scales = "free") +
scale_fill_brewer(type = "qual", palette = 6) +
theme_bw() +
theme(legend.position = "top",
      axis.text.x = element_text(vjust = .5,
                                 size = 6)) +
labs(x = "R packages for gene expression differential analysis",
     y = "Number of differentially expressed genes")
```



The R package *eXpress* is solely used to quantify the abundance of differentially expressed genes from **normalized** reads and aligned with **Bowtie 2**.

```
read.table("./data/summary.br.norm.44061.txt") %>%
    ggplot(aes(
    x = V1,
    y = V10,
    fill = V7)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    geom_text(aes(x = V1,
                  y = V10,
                  ymax = V10,
                  label = V10,
                  size = 1,
                  hjust = 0),
              position = position_dodge(width = 1)) +
    facet_wrap(~ V4 + V5 + V6 + V8,
               ncol = 2,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    theme_bw() +
    theme(legend.position = "top",
          axis.text.x = element_text(vjust = .5,
                                     size = 6)) +
    labs(x = "R packages for gene expression differential analysis",
         y = "Number of differentially expressed genes")
```

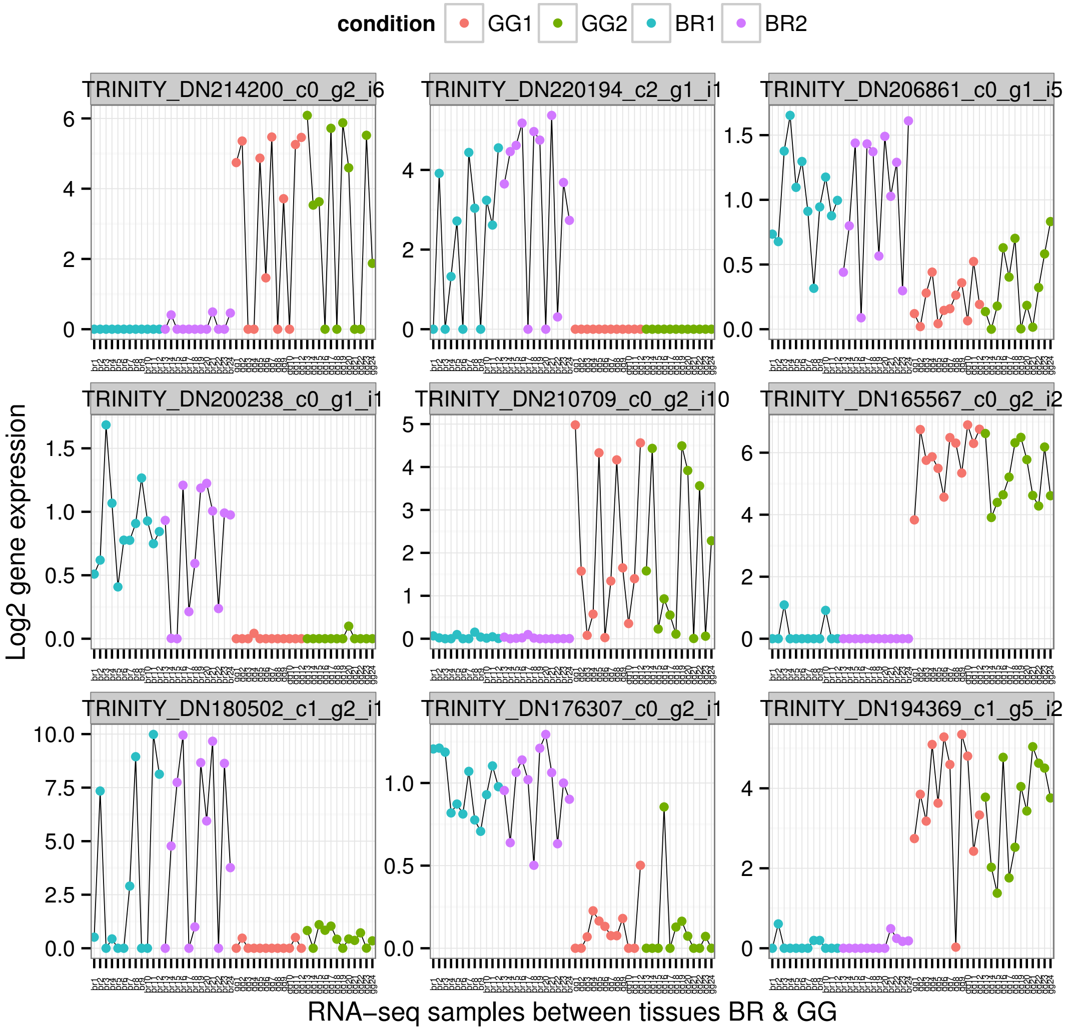


2 A linear representation of gene expression

Only selected genes can be represented as follow.

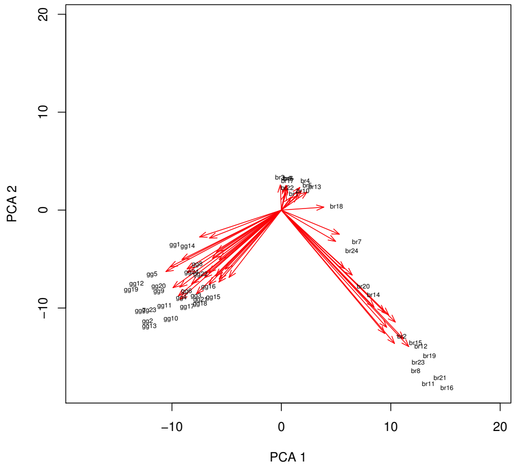
↰ Not more than 10 genes

```
dat <- t(read.table("./data/test.txt"))
dat <- data.frame(dat,
                  sample = rownames(dat),
                  condition = gl(4,12,48,
                    labels=c("GG1","GG2","BR1","BR2")))
dat %>%
    gather("genes","expression",1:(dim(dat)[2]-2)) %>%
    ggplot(aes(x = factor(sample,
                          c(paste("br",seq(1,24),sep=""),
                            paste("gg",seq(1,24),sep=""))),
               y = expression,
               group = condition)) +
    theme_bw() +
    geom_line(size = .2) +
    geom_point(aes(x = factor(sample),
                   y = expression,
                   colour = condition)) +
    facet_wrap(~ genes,
               ncol = 3,
               scales = "free") +
    labs(x = "RNA-seq samples between tissues BR & GG",
         y = "Log2 gene expression") +
    theme(legend.position = "top",
          axis.text.x = element_text(angle = 90,
                                     vjust = .5,
                                     size = 4)) +
    scale_fill_brewer(type = "qual", palette = 6,
                      name = "Oyster tissues and Diet conditions")
```



Principal component analysis on testing data.

```
dat <- read.table("./data/test.txt")
p = prcomp(dat, retx=T)
scores = p$x
loadings <- p$rotation
sd <- p$sdev
plot(scores[,1], scores[,2],
     xlab="PCA 1", ylab="PCA 2",
     type="n", xlim=c(min(scores[,1:2]),
                      max(scores[,1:2])),
     ylim=c(min(scores[,1:2]),
            max(scores[,1:2])))
arrows(0,0,loadings[,1]*50,loadings[,2]*50,
       length=0.1,angle=20, col="red")
text(loadings[,1]*50*1.3,loadings[,2]*50*1.3,
     rownames(loadings), col="black", cex=0.5)
```



3 System Information

The version number of R and packages loaded for generating the vignette were:

```
###save(list=ls(pattern=".*|.*"),file="PD.Rdata")
sessionInfo()

R version 3.2.1 (2015-06-18)
Platform: x86_64-unknown-linux-gnu (64-bit)
Running under: elementary OS Luna

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] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
 [1] dplyr_0.4.2         latticeExtra_0.6-26 RColorBrewer_1.1-2
 [4] glmnet_2.0-2        foreach_1.4.2       Matrix_1.2-1
 [7] leaps_2.9           caret_6.0-47        ggplot2_1.0.1
[10] lattice_0.20-31     xlsx_0.5.7          xlsxjars_0.6.1
[13] rJava_0.9-6         knitr_1.10.5        FactoMineR_1.30
[16] tidyr_0.2.0         RevoUtilsMath_3.2.1

loaded via a namespace (and not attached):
 [1] gtools_3.5.0         reshape2_1.4.1       splines_3.2.1
 [4] colorspace_1.2-6     mgcv_1.8-6           nloptr_1.0.4
 [7] DBI_0.3.1            plyr_1.8.3           stringr_1.0.0
[10] munsell_0.4.2        gtable_0.1.2         codetools_0.2-11
[13] evaluate_0.7         labeling_0.3         SparseM_1.6
[16] quantreg_5.11        pbkrtest_0.4-2       parallel_3.2.1
[19] highr_0.5            proto_0.3-10         Rcpp_0.11.6
[22] scales_0.2.5         flashClust_1.01-2    formatR_1.2
[25] BradleyTerry2_1.0-6  scatterplot3d_0.3-35 lme4_1.1-8
[28] digest_0.6.8         stringi_0.5-5        brglm_0.5-9
[31] grid_3.2.1           tools_3.2.1          magrittr_1.5
[34] lazyeval_0.1.10      cluster_2.0.2        car_2.0-25
[37] MASS_7.3-41          assertthat_0.1       minqa_1.2.4
[40] iterators_1.0.7      R6_2.0.1             nnet_7.3-10
[43] nlme_3.1-121         compiler_3.2.1
```