R implementation

Sleiman Bassim, PhD

March 24, 2017

Loaded functions:

↰ Project started July 10 2015

```
#source("/media/Data/Dropbox/humanR/01funcs.R")
rm(list=ls())
```

Load packages.

```
pkgs <- c('gdata','caret','leaps','glmnet','lattice','latticeExtra',
          'ggplot2', 'dplyr', 'tidyr', 'RColorBrewer','igraph')
lapply(pkgs, require, character.only = TRUE)
```

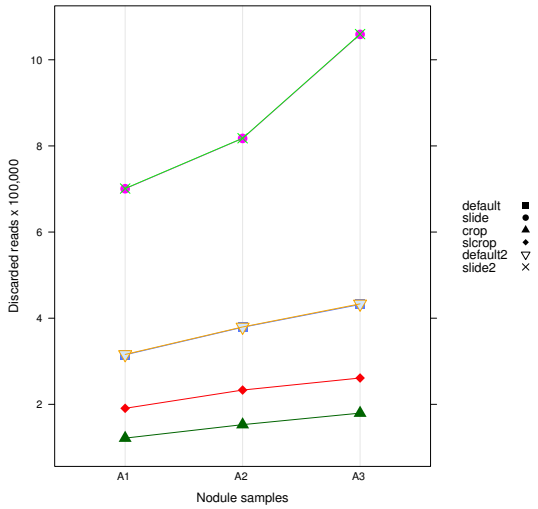
1 Quality controls and preprocessing

Many different options are available while trimming reads.

- Nature of PCR adapters (trueSeq2 or trueSeq3)
- Sliding window while reading contigs
- Crop less than a desired read length
- Minimum length for reads
- Trailing while removing the ends of reads with low quality

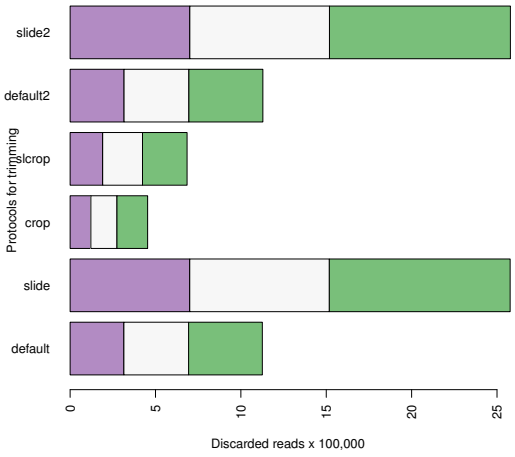
Different trimming options were tested. Iterations were run on combination of the trimming options. The plot shows the number of reads remaining after trimming using different adapters and a combination of trimming parameters (shapes). The *default* parameters includes clipping low quality segments and size of reads. The *slide* option include frameshiftting while filtering out low quality reads and the defaults. The *crop* option includes trimming the end of the reads and the defaults. The *slcrop* option includes sliding and cropping and the defaults. The *default2* parameters reliy on using different adapters for paired end sequencing. The *slide2* option include frameshiftting with the default parameters and different set of adapters.

```
trim <- read.xls("./data/Classeur1.xlsx", header = T, sheet = "Feuil1")
trim <- trim[1:3, ]
key.variety <- list(space = "right",
                    text = list(colnames(trim[, -c(1:2)])),
                    points = list(pch = c(15:18,25,4)))

dotplot(c(trim$Total-trim$default)/100000 +
           c(trim$Total-trim$slide)/100000 +
               c(trim$Total-trim$crop)/100000 +
                   c(trim$Total-trim$slcrop)/100000 +
                       c(trim$Total-trim$default2)/100000 +
                           c(trim$Total-trim$slide2)/100000
       ~ trim$Sample,
       data = trim,
       type = 'o',
        pch = c(15:18,25,4),
        key = key.variety,
        lty = 1, cex = 1.5,
       xlab = 'Nodule samples',
       ylab = 'Discarded reads x 100,000')
```



Another way to visualize the discarded reads. A1 = yellow, A2 = orange, A3 = red.

```
custom.colors <- brewer.pal(3, name = "PRGn")
barplot(as.matrix((trim$Total-trim[, -c(1:2)])/100000),
        horiz = TRUE,
        col = custom.colors,
        xlab = 'Discarded reads x 100,000',
        ylab = 'Protocols for trimming',
        las = 2)
```



2 Mapping reads to reference

Two sets of reads were mapped to 3 different assembled references. First batch from the trimmed reads with the default parameters (adapters clipping, trailing, and minimum length) and TrueSeq3 adapters. Second batch were also trimmed with default settings but using TrueSeq2 adapters. With the second batch of adapters, more reads were trimmed and discarded. This analysis will try to show the regression of mapped reads to the length of each reference.

↰ The references v017 and genome v21 are test references, while genome v015 is the one used in the published work.

```
ref.genome1 <- read.table("./data/refGenome/A1.htseq.counts.txt")
ref.genome2 <- read.table("./data/refGenome/A2.htseq.counts.txt")
ref.genome3 <- read.table("./data/refGenome/A3.htseq.counts.txt")
ref.genome4 <- read.table("./data/refGenome/A1-4.htseq.counts.txt")
ref.genome5 <- read.table("./data/refGenome/A2-4.htseq.counts.txt")
ref.genome6 <- read.table("./data/refGenome/A3-4.htseq.counts.txt")
```

Merge by reference position all the mapped reads from different trimming options.

```
ref.genome <- data.frame(
    A1 = ref.genome1[-c(556:560), 2],
    A2 = ref.genome2[-c(556:560), 2],
    A3 = ref.genome3[-c(556:560), 2],
    A1.4 = ref.genome4[-c(556:560), 2],
    A2.4 = ref.genome5[-c(556:560), 2],
    A3.4 = ref.genome6[-c(556:560), 2],
    contigs = ref.genome1[-c(556:560), 1])
dim(ref.genome)

[1] 555   7

genome <- read.table("./data/QPX_Genome_v017.gff3")
genome1 <- data.frame(contigs= genome[,1], length = genome$V5)
```

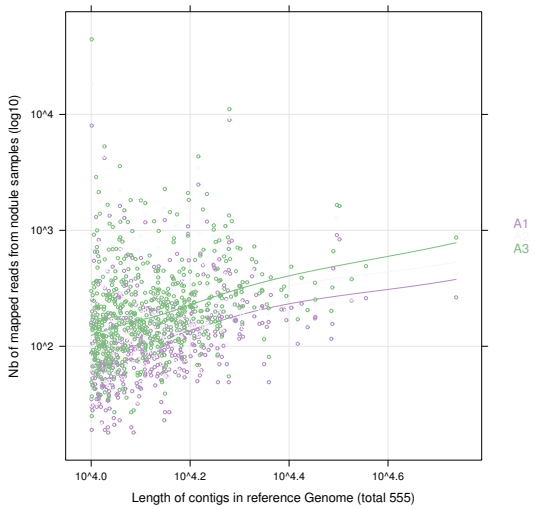
Merge by reference position the length and number of mapped reads.

↰ Data are from trimming with TrueSeq3 (selected) and TrueSeq2 (testing). We only plot TrueSeq3.

```
ref.genome.mix <- merge(genome1, ref.genome)
head(ref.genome.mix)

               contigs length  A1  A2  A3 A1.4 A2.4 A3.4
1 QPX_v017_contig_1007  15433 117 197 249  117  197  249
2 QPX_v017_contig_1020  12397 123 164 171  123  164  171
3 QPX_v017_contig_1021  18562 335 487 596  335  488  596
4 QPX_v017_contig_1023  19919 116 198 331  116  198  331
5  QPX_v017_contig_103  10989  71 111 107   71  111  105
6 QPX_v017_contig_1034  10178 196 289 655  196  289  655

dim(ref.genome.mix)

[1] 555   8
```

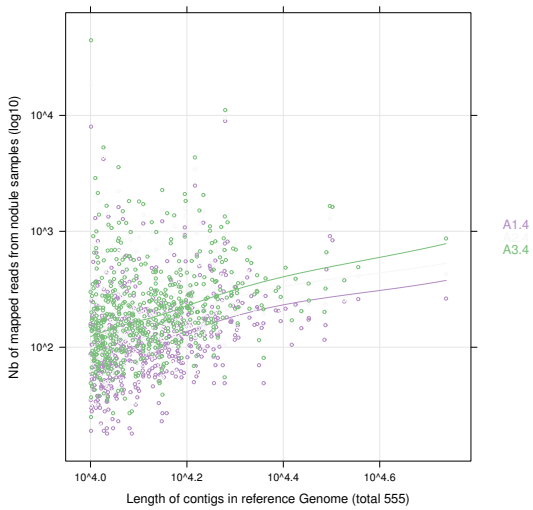
Correlation between read length and number of mapped reads on the (testing) genome of QPX with the remaining reads from the default trimming with TrueSeq3 adapters.

```
custom.colors <- brewer.pal(3, name = "PRGn")
key.variety <- list(space = "right",
                    text = list(colnames(ref.genome.mix[, 3:5])),
                    col = custom.colors)
xyplot(A1 + A2 + A3 ~ length,
       data = ref.genome.mix,
       xlab = 'Length of contigs in reference Genome (total 555)',
       ylab = 'Nb of mapped reads from nodule samples (log10)',
       col = custom.colors,
       cex = 0.5,
       type = c("g", "p", "smooth"),
       scales = list(log = 10),
       key = key.variety)
```



Regression between reads and length of contigs in reference genome with (testing) adapters TrueSeq2 under default trimming settings.

```
custom.colors <- brewer.pal(3, name = "PRGn")
key.variety <- list(space = "right",
                    text = list(colnames(ref.genome.mix[, 6:8])),
                    col = custom.colors)
xyplot(A1.4 + A2.4 + A3.4 ~ length,
#                    alpha = .5,
       data = ref.genome.mix,
       xlab = 'Length of contigs in reference Genome (total 555)',
       ylab = 'Nb of mapped reads from nodule samples (log10)',
       col = custom.colors,
       cex = 0.5,
       type = c("g", "p", "smooth"),
       scales = list(log = 10),
       key = key.variety)
```



Previously we regressed the number of mapped reads over the reference genome (v21) of QPX. Now its time to do the same thing over the reference transcriptome of QPX (v17). Both references belong the Steve Roberts.

```
ref.transcriptome1 <- read.table("./data/refTranscriptome/A1.htseq.counts.txt")
ref.transcriptome2 <- read.table("./data/refTranscriptome/A2.htseq.counts.txt")
ref.transcriptome3 <- read.table("./data/refTranscriptome/A3.htseq.counts.txt")
ref.transcriptome4 <- read.table("./data/refTranscriptome/A1-4.htseq.counts.txt")
ref.transcriptome5 <- read.table("./data/refTranscriptome/A2-4.htseq.counts.txt")
ref.transcriptome6 <- read.table("./data/refTranscriptome/A3-4.htseq.counts.txt")
dim(ref.transcriptome1)
```

```
[1] 11779     2
```

```
tail(ref.transcriptome1)
```

```
                                    V1       V2
11774 QPX_transcriptome_v2_Contig_9_3        6
11775                    __no_feature        0
11776                     __ambiguous      568
11777                  __too_low_aQual   147407
11778                    __not_aligned 29746511
11779          __alignment_not_unique        0
```

Merge by the position on the reference transcriptome all the mapped reads.

```
ref.transcriptome <- data.frame(
```

↰ Data is from trimming with TrueSeq3 (selected) and TrueSeq2 (testing). We only plot TruSeq3

```
    A1 = ref.transcriptome1[-c(11775:11779), 2],
    A2 = ref.transcriptome2[-c(11775:11779), 2],
    A3 = ref.transcriptome3[-c(11775:11779), 2],
    A1.4 = ref.transcriptome4[-c(11775:11779), 2],
    A2.4 = ref.transcriptome5[-c(11775:11779), 2],
    A3.4 = ref.transcriptome6[-c(11775:11779), 2],
    contigs = ref.transcriptome1[-c(11775:11779), 1])
dim(ref.transcriptome)
```

```
[1] 11774     7
```

```
head(ref.transcriptome)
```

```
  A1 A2 A3 A1.4 A2.4 A3.4                              contigs
1  0  2  0    0    2    0 QPX_transcriptome_v2_Contig_10002_1
2  0  0  0    0    0    0 QPX_transcriptome_v2_Contig_10002_2
3  2  2  7    2    2    7  QPX_transcriptome_v2_Contig_1000_1
4  1  0  0    1    0    0  QPX_transcriptome_v2_Contig_1000_2
5  0  0  0    0    0    0  QPX_transcriptome_v2_Contig_1000_3
6 58 72 99   59   72  100  QPX_transcriptome_v2_Contig_1000_4
```

```
transcriptome <- read.table("./data/QPX_transcriptome_v2orf.gff3")
transcriptome1 <- data.frame(contigs= transcriptome[,1], length = transcriptome$V5)
```

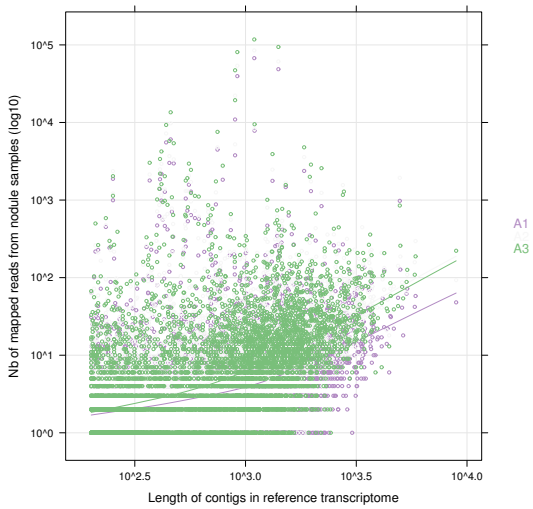
39 Merge by the position on the reference transcriptome the length and number of the mapped reads.

```
ref.transcriptome.mix <- merge(transcriptome1, ref.transcriptome)
head(ref.transcriptome.mix)
```

```
                            contigs length  A1  A2  A3 A1.4 A2.4
1   QPX_transcriptome_v2_Contig_10_1    345   0   0   0    0    0
2   QPX_transcriptome_v2_Contig_10_2   1416 222 253 469  222  253
3   QPX_transcriptome_v2_Contig_10_3    234   0   0   0    0    0
4   QPX_transcriptome_v2_Contig_10_4    201   0   0   0    0    0
5 QPX_transcriptome_v2_Contig_1000_1    201   2   2   7    2    2
6 QPX_transcriptome_v2_Contig_1000_2    258   1   0   0    1    0
  A3.4
1    0
2  469
3    0
4    0
5    7
6    0
```

40 Plot the correlation between read length and number of mapped reads on the transcriptome of QPX with
41 the remaining reads from the default trimming with TrueSeq3 adapters.

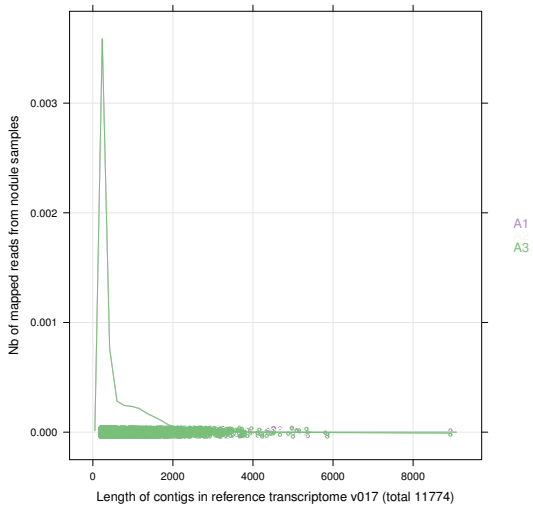
```
custom.colors <- brewer.pal(3, name = "PRGn")
key.variety <- list(space = "right",
                    text = list(colnames(ref.transcriptome.mix[, 3:5])),
                    col = custom.colors)
xyplot(A1 + A2 + A3 ~ length,
       data = ref.transcriptome.mix,
       xlab = 'Length of contigs in reference transcriptome',
       ylab = 'Nb of mapped reads from nodule samples (log10)',
       col = custom.colors,
       cex = 0.5,
       type = c("g", "p", "smooth"),
       scales = list(log = 10),
       key = key.variety)
```



2.1 Concentration of contigs in different libraries

Density plot between reference transcriptome and assembled contigs. The plot shows a high concentration of contigs under 2000 base pair. Trimming parameters are of default with TrueSeq3 adapters.

```
custom.colors <- brewer.pal(3, name = "PRGn")
key.variety <- list(space = "right",
                    text = list(colnames(ref.genome.mix[, 3:5])),
                    col = custom.colors)
densityplot(A1 + A2 + A3 ~ length,
       data = ref.transcriptome.mix,
#             alpha = .7,
            xlab = 'Length of contigs in reference transcriptome v017 (total 11774)',
       ylab = 'Nb of mapped reads from nodule samples',
       col = custom.colors,
       cex = 0.5,
       type = c("g", "p", "smooth"),
#       scales = list(log = 10),
       key = key.variety)
```

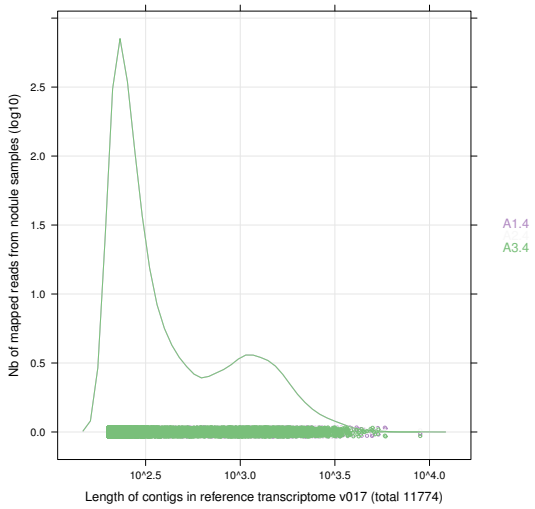


Plot correlation same as above but with TrueSeq2 adapters with default parameters.

```
custom.colors <- brewer.pal(3, name = "PRGn")
```

```
key.variety <- list(space = "right",
                    text = list(colnames(ref.genome.mix[, 6:8])),
                    col = custom.colors)
densityplot(A1.4 + A2.4 + A3.4 ~ length,
       data = ref.transcriptome.mix,
#            alpha = .9,
       xlab = 'Length of contigs in reference transcriptome v017 (total 11774)',
       ylab = 'Nb of mapped reads from nodule samples (log10)',
       col = custom.colors,
       cex = 0.5,
       type = c("g", "p", "smooth"),
       scales = list(log = 10),
       key = key.variety)
```

```
Warning in densityplot.formula(A1.4 + A2.4 + A3.4 ~ length, data = ref.transcriptome.mix,
:  Can't have log Y-scale
```



2.2 Testing contig length and mapping with different assembled references

Load the number of mapped reads to the MMETSP0098 transcriptome before discarding duplicates.

```
ref.dupA1R3 <- read.table("./data/refMME98/A1.htseq.counts.txt")
ref.dupA2R3 <- read.table("./data/refMME98/A2.htseq.counts.txt")
ref.dupA3R3 <- read.table("./data/refMME98/A3.htseq.counts.txt")
```

Merge all mapped reads to MMETSP0098 reference transcriptome before discarding duplicates (ie, raw counts).

```
ref.mme98 <- data.frame(A1 = ref.dupA1R3$V2,
                        A2 = ref.dupA2R3$V2,
                        A3 = ref.dupA3R3$V2,
                        contigs = ref.dupA1R3$V1)
```

Add the length values to each contig mapped to MMETsp0098. But first remove extra rows.

```
nr <- nrow(ref.mme98)
ref.mme98 <- ref.mme98[1:(nr-5), ]
tail(ref.mme98)
```

```
      A1 A2 A3                    contigs
15484  0  0  0 MMETSP0098-20131031|9992
15485 12  6 17 MMETSP0098-20131031|9993
15486  0  0  0 MMETSP0098-20131031|9995
15487  3  3  3 MMETSP0098-20131031|9996
15488  0  0  0 MMETSP0098-20131031|9998
15489  2  9  5 MMETSP0098-20131031|9999
```

Load the number of mapped reads to the MMETSP0099_2 transcriptome before discarding duplicates.

↰ MMETSP is a project for sequencing different strains of QPX. MMETSP0098 and MMETPS00992 were used here. Both strains come fron New York and Virginia respectively. Official project can be found here.

```
ref.dupA1R4 <- read.table("./data/refMME992/A1.htseq.counts.txt")
ref.dupA2R4 <- read.table("./data/refMME992/A2.htseq.counts.txt")
ref.dupA3R4 <- read.table("./data/refMME992/A3.htseq.counts.txt")
```

Merge all mapped reads to MMETSP0099_2 reference transcriptome before discarding duplicates (ie, raw counts).

```
ref.mme992 <- data.frame(A1 = ref.dupA1R4$V2,
                         A2 = ref.dupA2R4$V2,
                         A3 = ref.dupA3R4$V2,
                         contigs = ref.dupA1R4$V1)
```

Add the length values to each contig mapped to MMETsp0099_2. But first remove extra rows.

```
nr <- nrow(ref.mme992)
ref.mme992 <- ref.mme992[1:(nr-5), ]
tail(ref.mme992)
```

```
      A1 A2 A3                     contigs
11762  0  0  0 MMETSP0099_2-20121227|9994
11763  0  8  9 MMETSP0099_2-20121227|9995
11764  0  0  0 MMETSP0099_2-20121227|9996
11765  0  0  1 MMETSP0099_2-20121227|9997
11766  0  0  0 MMETSP0099_2-20121227|9998
11767  0  0  0 MMETSP0099_2-20121227|9999
```

Load the number of mapped reads to SR v015 genome before discarding duplicates.

```
ref.dupA1R5 <- read.table("./data/refGenomV015/A1.htseq.counts.txt")
ref.dupA2R5 <- read.table("./data/refGenomV015/A2.htseq.counts.txt")
ref.dupA3R5 <- read.table("./data/refGenomV015/A3.htseq.counts.txt")
```

Merge all mapped reads to SR v015 reference genome before discarding duplicates (ie, raw counts).

```
ref.genomv015 <- data.frame(A1 = ref.dupA1R5$V2,
                         A2 = ref.dupA2R5$V2,
                         A3 = ref.dupA3R5$V2,
                         contigs = ref.dupA1R5$V1)
```

Add the length values to each contig mapped to SR v015 reference genome. But first remove extra rows.

```
nr <- nrow(ref.genomv015)
ref.genomv015 <- ref.genomv015[1:(nr-5), ]
tail(ref.genomv015)
```

```
      A1 A2 A3              contigs
21275  0  1  2 QPX_v015_contig_9994
21276  0  0  0 QPX_v015_contig_9995
21277  1  5  3 QPX_v015_contig_9996
21278  1  2  2 QPX_v015_contig_9997
21279  3  1  2 QPX_v015_contig_9998
21280  9  5 21 QPX_v015_contig_9999
```

After aligning the reads to a reference duplicates must be removed. Testing was done with different reference transcriptomes and genomes to assess the strength of the parameters used for removing the duplicate reads and reducing bias for better coverage.
First load the sample reads mapped to reference genome (without duplication) of Steve Roberts.

```
nodupA1R1 <- read.table("./data/nodupR1/A1.htseq.nodup.counts.txt")
nodupA2R1 <- read.table("./data/nodupR1/A2.htseq.nodup.counts.txt")
nodupA3R1 <- read.table("./data/nodupR1/A3.htseq.nodup.counts.txt")
```

Second load the sample reads mapped to reference transcriptome (withtout duplication) of Steve Roberts.

```
nodupA1R2 <- read.table("./data/nodupR2/A1.htseq.nodup.counts.txt")
nodupA2R2 <- read.table("./data/nodupR2/A2.htseq.nodup.counts.txt")
nodupA3R2 <- read.table("./data/nodupR2/A3.htseq.nodup.counts.txt")
```

Third load the sample reads mapped to reference transcriptome (without duplication) of MMESTO0098.

```
nodupA1R3 <- read.table("./data/nodupR3/A1.htseq.counts.nodup.txt")
nodupA2R3 <- read.table("./data/nodupR3/A2.htseq.counts.nodup.txt")
nodupA3R3 <- read.table("./data/nodupR3/A3.htseq.counts.nodup.txt")
```

Forth load of sample reads mapped to reference transcriptome MMETSP0099_2.

```
nodupA1R4 <- read.table("./data/nodupR4/A1.htseq.counts.nodup.txt")
nodupA2R4 <- read.table("./data/nodupR4/A2.htseq.counts.nodup.txt")
nodupA3R4 <- read.table("./data/nodupR4/A3.htseq.counts.nodup.txt")
```

Forth load of sample reads mapped to reference SR genome v015 with approximately 21,000 contigs.

```
nodupA1R5 <- read.table("./data/nodupR5/A1.htseq.counts.nodup.txt")
nodupA2R5 <- read.table("./data/nodupR5/A2.htseq.counts.nodup.txt")
nodupA3R5 <- read.table("./data/nodupR5/A3.htseq.counts.nodup.txt")
```

Merge mapped reads relative to the followingf references.

- R1 = genome of QPX (steve roberts, 555 contigs)
- R2 = transcriptome of QPX (steve roberts)
- R3 = transcriptome of QPX MMETSP0098, New York strain
- R4 = transcriptome of QPX MMETSP0099_2, Virginia strain
- R5 = genome of QPX (steve roberts v015, approx. 21,000 contigs)

```
allR1 <- data.frame(A1n = nodupA1R1$V2,
                    A2n = nodupA2R1$V2,
                    A3n = nodupA3R1$V2,
                    reference = rep("genomSRv017", nrow(nodupA1R1)),
                      contigs = nodupA1R1$V1)
allR1 <- allR1[1:555, ]

allR2 <- data.frame(A1n = nodupA1R2$V2,
                    A2n = nodupA2R2$V2,
                    A3n = nodupA3R2$V2,
                    reference = rep("trxSRv022", nrow(nodupA1R2)),
                      contigs = nodupA1R2$V1)
allR2 <- allR2[1:11774, ]

allR3 <- data.frame(A1n = nodupA1R3$V2,
                    A2n = nodupA2R3$V2,
                    A3n = nodupA3R3$V2,
                    reference = rep("trxMME98", nrow(nodupA1R3)),
                      contigs = nodupA1R3$V1)
allR3 <- allR3[1:15489, ]

allR4 <- data.frame(A1n = nodupA1R4$V2,
                    A2n = nodupA2R4$V2,
                    A3n = nodupA3R4$V2,
                    reference = rep("trxMME992", nrow(nodupA1R4)),
                      contigs = nodupA1R4$V1)
allR4 <- allR4[1:c(nrow(nodupA1R4)-5), ]

allR5 <- data.frame(A1n = nodupA1R5$V2,
                    A2n = nodupA2R5$V2,
                    A3n = nodupA3R5$V2,
                    reference = rep("genomSRv015", nrow(nodupA1R5)),
                      contigs = nodupA1R5$V1)
allR5 <- allR5[1:c(nrow(nodupA1R5)-5), ]
```

Put before /after duplicates removal in one dataset for genome of Steve Roberts.

```
genomeSR <- merge(ref.genome.mix[, 1:5], allR1)
head(genomeSR)
```

```
               contigs length  A1  A2  A3 A1n A2n A3n   reference
1 QPX_v017_contig_1007  15433 117 197 249 109 191 240 genomSRv017
2 QPX_v017_contig_1020  12397 123 164 171 118 157 159 genomSRv017
3 QPX_v017_contig_1021  18562 335 487 596 319 460 568 genomSRv017
4 QPX_v017_contig_1023  19919 116 198 331 109 196 326 genomSRv017
5  QPX_v017_contig_103  10989  71 111 107  68 106 104 genomSRv017
6 QPX_v017_contig_1034  10178 196 289 655 185 279 592 genomSRv017
```

```
rownames(genomeSR) <- genomeSR$contigs
genomeSR <- t(genomeSR[, -c(1,9)])
genomeSR[, 1:3]
```

```
       QPX_v017_contig_1007 QPX_v017_contig_1020 QPX_v017_contig_1021
length                15433                12397                18562
A1                      117                  123                  335
A2                      197                  164                  487
A3                      249                  171                  596
A1n                     109                  118                  319
A2n                     191                  157                  460
A3n                     240                  159                  568
```

```
genomeSR <- data.frame(genomeSR,
                       y = c(2, rep(0,3), rep(1,3)))
```

Put the before /after duplicates removal in one datasetfor transcriptome of SR.

```
transcriptomeSR <- merge(ref.transcriptome.mix[, 1:5], allR2)
head(transcriptomeSR)
```

```
                            contigs length  A1  A2  A3 A1n A2n A3n
1   QPX_transcriptome_v2_Contig_10_1    345   0   0   0   0   0   0
2   QPX_transcriptome_v2_Contig_10_2   1416 222 253 469 202 238 409
3   QPX_transcriptome_v2_Contig_10_3    234   0   0   0   0   0   0
4   QPX_transcriptome_v2_Contig_10_4    201   0   0   0   0   0   0
5 QPX_transcriptome_v2_Contig_1000_1    201   2   2   7   2   2   7
6 QPX_transcriptome_v2_Contig_1000_2    258   1   0   0   1   0   0
  reference
1 trxSRv022
2 trxSRv022
3 trxSRv022
4 trxSRv022
5 trxSRv022
6 trxSRv022
```

Present difference for each sample mapped to the references. First merge all samples before /after duplicates were removed.

```
allRefs <- rbind(allR1, allR2, allR3, allR4, allR5)
```

```
dim(allRefs)
```

```
[1] 60865     5
```

```
summary(allRefs$reference)
```

```
genomSRv017   trxSRv022    trxMME98   trxMME992 genomSRv015
        555       11774       15489       11767       21280
```

```
allRefs.raw <- rbind(ref.genome.mix[, 3:5],
                     ref.transcriptome.mix[, 3:5],
                     ref.mme98[, 1:3],
                     ref.mme992[, 1:3],
                     ref.genomv015[, 1:3])
dim(allRefs.raw)
```

```
[1] 60865     3
```

```
allDF <- cbind(allRefs, allRefs.raw)
allDF[sample(1:20000, 5), ]
```

```
      A1n A2n A3n reference                              contigs A1
3237    6   9   5 trxSRv022  QPX_transcriptome_v2_Contig_2139_4  0
7391    1   1   3 trxSRv022 QPX_transcriptome_v2_Contig_4465_11  0
17093  13  17  27  trxMME98           MMETSP0098-20131031|15506 15
12330  69 100 112  trxMME98               MMETSP0098-20131031|1 75
12498  23  18  21  trxMME98           MMETSP0098-20131031|10176 24
       A2  A3
3237    0   1
7391    0   0
17093  18  27
12330 104 117
12498  18  22
```

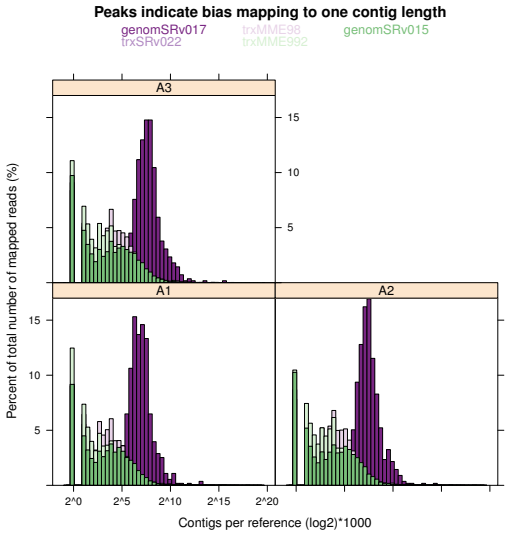
Plot the difference before and after duplicates were discarded. The number of mapped reads to the reference contigs is descriptive for any bias in contig assembly. For example in the case of SR genome v017, more than 20 % of A1, A2, A3 reads align to a small set of contigs. The best distribution is a constant one.

Even though we did not plot length of contigs, the analyzes above demonstrate that length is linearly correlated to the number of mapped reads. Therefore, peaks indicate a specific preference that reads have to map to an assembled reference.

↰ A high resolution version of the plot below can be found in the Supplemental Information

```
custom.colors <- brewer.pal(6, name = "PRGn")
histogram( ~ A1 + A2 + A3 ,
     data = allDF,
     nint = 50,
     scales = list(log = 2),
     type = "p",
     ylim = c(0,17),
     groups = allDF$reference,
     panel = function(...) panel.superpose(...,
         panel.groups = panel.histogram,
         col = custom.colors,
         alpha = 1),
     auto.key=list(columns=3,
         rectangles = FALSE,
         col = custom.colors),
          main = 'Peaks indicate bias mapping to one contig length',
          ylab = 'Percent of total number of mapped reads (%)',
          xlab = 'Contigs per reference (log2)*1000'
     )
```

```
Warning in histogram.formula(~A1 + A2 + A3, data = allDF, nint = 50, scales =
list(log = 2), :  Can't have log Y-scale
```



3 Extracting QPX reads

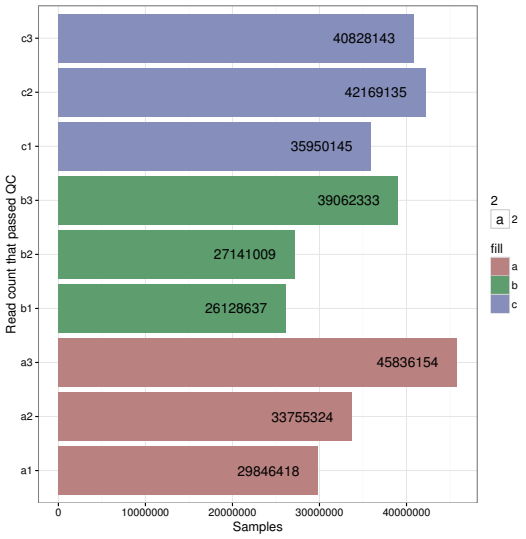
Load file with read counts per sample.

- A samples as nodule tissue
- B samples as non nodule diseased tissue
- C samples as non nodule non diseased tissue

Raw reads have been trimmed, mapped to reference genome (Steve Roberts v15 with 21280 contigs), sorted by position on the genome, and cleaned from duplicated reads.

↰ Hypothetically these raw reads includes specific QPX reads

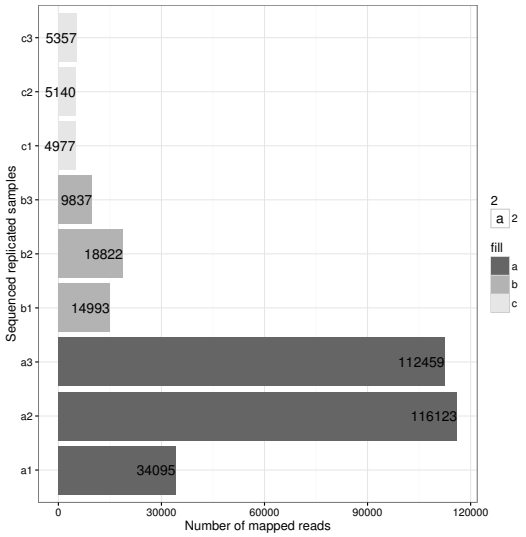
```
reads.counts <- read.xls("./data/mappedNodules.xlsx", sheet = 1)
reads.counts$fill <- gl(3, 3, 9, labels = c("a", "b", "c"))
ggplot(reads.counts,
       aes(x = sample,
           y = raw.reads,
           fill = fill)) +
    coord_flip() +
        theme_bw() +
            geom_bar(stat = "identity") +
                geom_text(aes(x = sample,
                              y = raw.reads,
                              ymax = raw.reads,
                              label = raw.reads,
                              size = 2,
                              hjust = 1.3)) +
                    scale_fill_hue(c = 40, l = 60) +
                labs(x = "Read count that passed QC",
                     y = "Samples")
```



Number of reads that mapped to the reference genome of QPX.

↰ These reads are probably those of QPXs'

```
ggplot(reads.counts,
       aes(x = sample,
           y = mapped.reads,
           fill = fill)) +
    geom_bar(stat = "identity") +
        geom_text(aes(x = sample,
                      y = mapped.reads,
                      ymax = mapped.reads,
                      label = mapped.reads,
                      hjust = 1,
                      size = 2)) +
             coord_flip() +
                 theme_bw() +
                     scale_fill_grey(start = .4, end = .9) +
                 labs(y = "Number of mapped reads",
                      x = "Sequenced replicated samples")
```



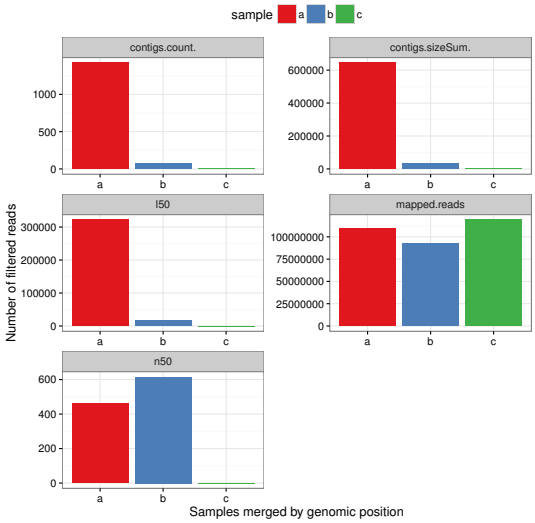
3.1 QPX transcripts assembled without the hosts transcripts

Mapped reads to the QPX reference genome are then assembled into contigs (ie, the reads showing in the chart above).

↰ These contigs must be specific transcripts to QPX

```
read.xls("./data/mappedNodules.xlsx", sheet = 2) %>%
```

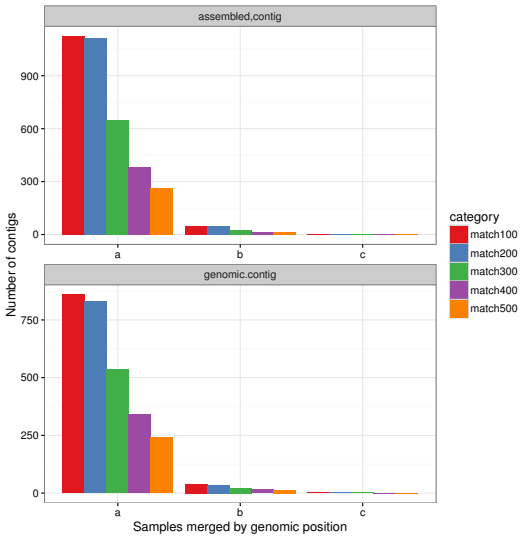
```
    gather("category", "counts", 3:7) %>%
      ggplot(aes(x = sample,
                 y = counts,
                 fill = sample)) +
    geom_bar(stat = "identity") +
        theme_bw() +
            facet_wrap( ~ category,
                        ncol = 2,
                        scales = "free") +
                theme(legend.position = "top") +
            labs(x = "Samples merged by genomic position",
                 y = "Number of filtered reads") +
    scale_fill_brewer(type = "qual", palette = 6)
```



All contigs were then mapped to the reference genome of QPX. The chart shows the number of contigs that maps with an increasing length going from 100 to 500.

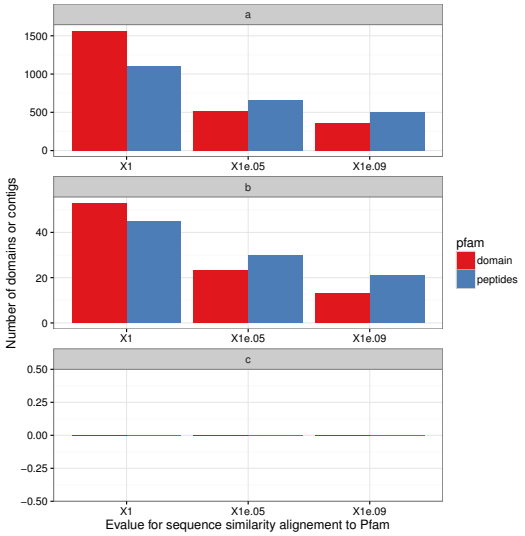
↰ Helps discard misassembled contigs or non QPX ones

```
read.xls("./data/mappedNodules.xlsx", sheet = 4) %>%
    gather("category", "count", 3:7) %>%
      ggplot(aes(x = sample,
            y = count,
            fill = category)) +
    theme_bw() +
        geom_bar(stat = "identity",
                 position = "dodge") +
            facet_wrap( ~ blat, ncol = 1, scale = "free") +
#               scale_fill_manual(values = brewer.pal(5, "Greens")) +
            labs(x = "Samples merged by genomic position",
                 y = "Number of contigs") +
    scale_fill_brewer(type = "qual", palette = 6)
```



All contigs were than translated to peptides in 3 frames. Each peptide was than aligned to the whole PFAM library (v28, date: Jul 14 2015).

```
read.xls("./data/mappedNodules.xlsx", sheet = 3) %>%
    gather("category", "count", 3:5) %>%
      ggplot(aes(x = category,
            y = count,
            fill = pfam)) +
    geom_bar(stat = "identity",
             position = "dodge") +
        theme_bw() +
            facet_wrap( ~ sample, ncol = 1,
                        scales = "free") +
            labs(x = "Evalue for sequence similarity alignement to Pfam",
                 y = "Number of domains or contigs") +
    scale_fill_brewer(type = "qual", palette = 6)
```

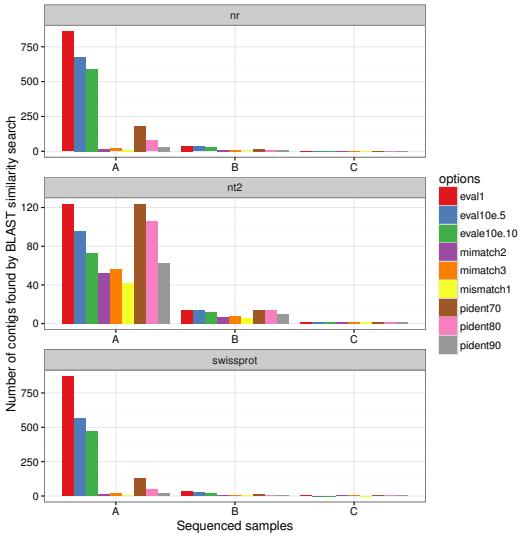


3.2 Evaluating identified QPX contigs with sequence similarities

BLAST is used at this step to align the contigs found to non redundant (NR), nucleotide (NT), and swissprot databases. Contigs were translated into peptide sequencing with EMBOSS *transeq* in all 3 possible frames. All contigs were mapped to genome v15 of S. Roberts. Top hit sequences were filtered either with an evalue score, the percentage of identity between query and target sequences, and the number of mismatches found in the aligned region.

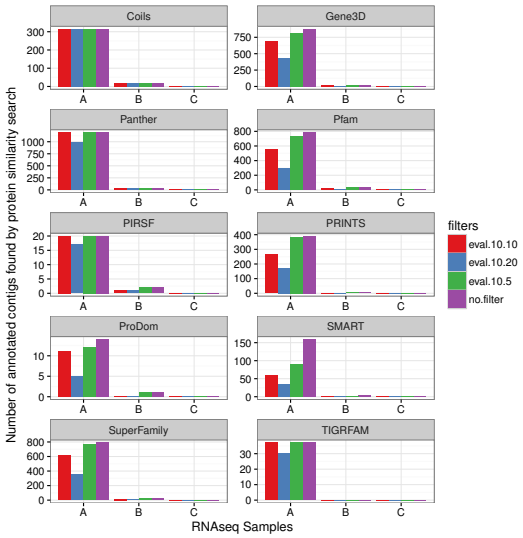
```
read.xls("./data/blast.xlsx", sheet = 1) %>%
```

```
    gather("options", "counts", 4:12) %>%
      ggplot(aes(x = sample,
            y = counts,
            fill = options)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    theme_bw() +
#     geom_text(aes(x = sample,
#                   y = counts,
#                   ymax = counts,
#                   label = counts,
#                   size = 1,
#                   hjust = .5),
#               position = position_dodge(width = 1)) +
    facet_wrap( ~ ncbi, ncol = 1,
                scales = 'free') +
    labs(x = "Sequenced samples",
         y = "Number of contigs found by BLAST similarity search") +
    scale_fill_brewer(type = "qual",
                      palette = 6)
```



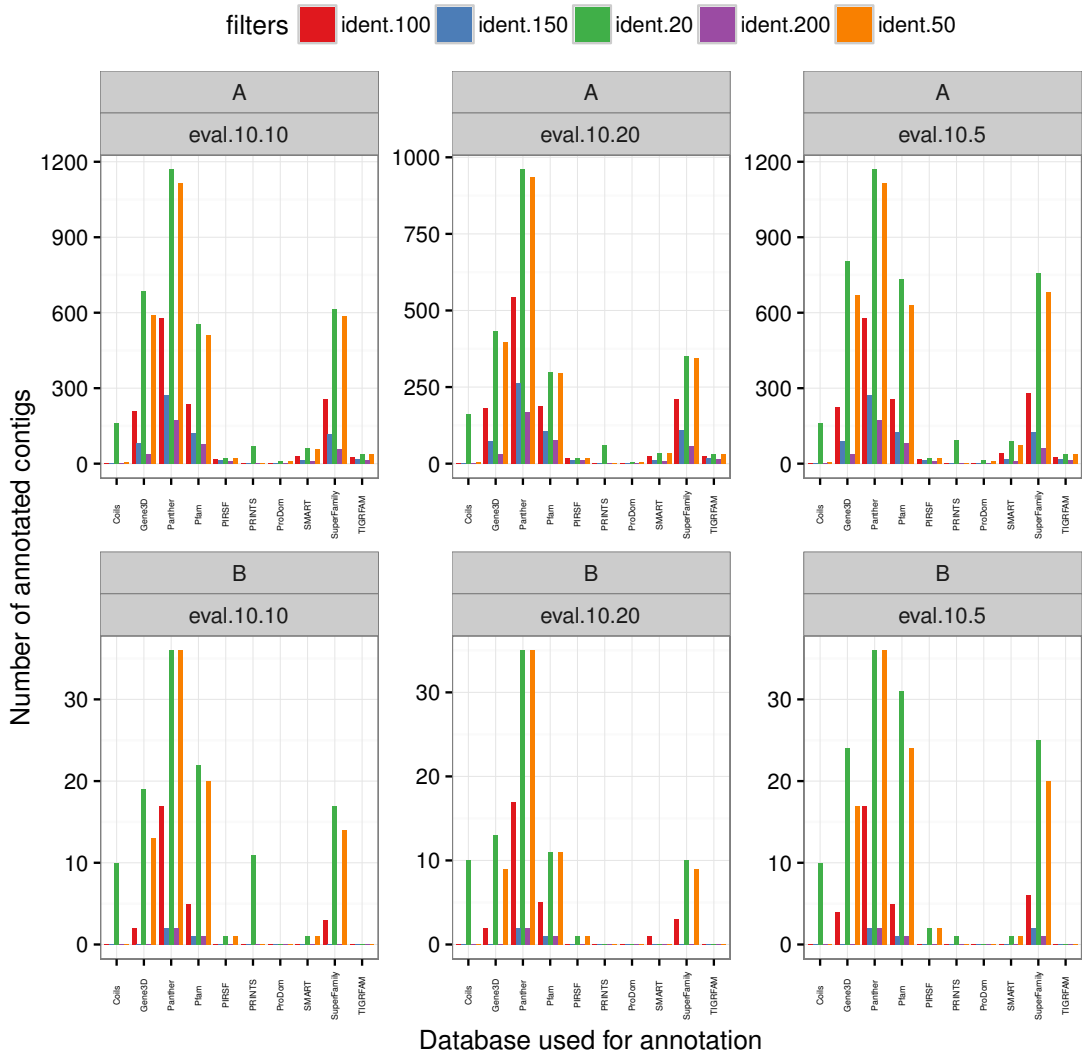
Sequences from A, B, C samples are also aligned to 10 other databases. We used interpro accession numbers to get GO-terms too. We show below the number of contigs if we filter them only by e-value. The no-filter label represent the total number of sequences per database.

```
read.xls("./data/interpro.xlsx", sheet = 1) %>%
    gather("filters", "value", 2:5) %>%
      ggplot(aes(x = sample,
            y = value,
            fill = filters)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    theme_bw() +
        facet_wrap( ~ database,
                ncol = 2,
                scales = "free") +
    labs(x = "RNAseq Samples",
         y = "Number of annotated contigs found by protein similarity search") +
    scale_fill_brewer(type = "qual",
                      palette = 6)
```



We show here the number of contigs annotated by database and filtered by both evalue and the length of the similarity between query and target sequences.

```
read.xls("./data/interpro.xlsx", sheet = 2) %>%
    gather("filters", "value", 3:7) %>%
    ggplot(aes(x = database,
               y = value,
               fill = filters)) +
                   theme_bw() +
                       geom_bar(stat = "identity",
                                position = "dodge") +
                       facet_wrap( ~ sample + option,
                                   ncol = 3,
                                   scales = "free") +
                       labs(x = "Database used for annotation",
                            y = "Number of annotated contigs") +
                       scale_fill_brewer(type = "qual",
                                         palette = 6) +
        theme(legend.position = "top",
          axis.text.x = element_text(angle = 90,
                                     vjust = .5,
                                     size = 4))
```



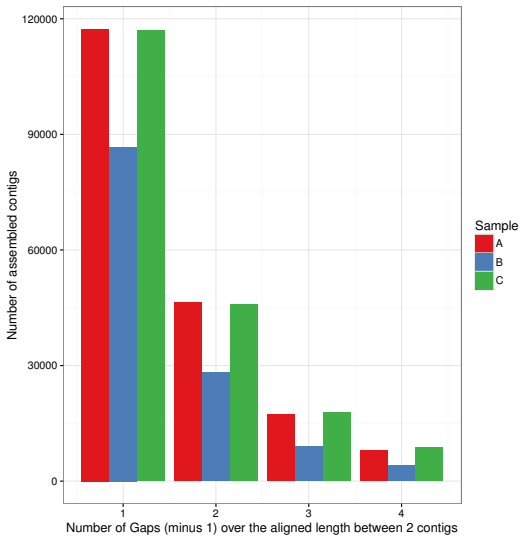
4 Removing clam genes from QPX inferred genes

For each sequenced tissue we assembled a list of transcripts. The A-list theoretically contains QPX genes, B- and C- contain mostly clam genes. Confidence in the A-list is only valid if clam genes were discarded, and QPX only genes remained.

The same strategy of contig assembly is carried out for clam. A, B, C raw reads are aligned to the clam reference transcriptome. Only the aligned reads are used to assemble 3 lists of hypothetically clam transcripts. There is many transcripts so HMMER is used to select the ones containing significant protein domains. The selected transcripts are aligned to the clam reference and only those that align without any gap (between reference and query) are kept.

```
gapsA <- read.table("./data/A.e10.blat.gaps.clam.txt")
```

```
gapsB <- read.table("./data/B.e10.blat.gaps.clam.txt")
gapsC <- read.table("./data/C.e10.blat.gaps.clam.txt")
df <- full_join(gapsA, gapsB, by="V2") %>%
    full_join(gapsC, by="V2")
colnames(df) <- c("A", "Gaps", "B", "C")
df[, c(2,1,3,4)] %>%
    gather("Sample", "Count", 2:4) %>%
    filter(Gaps<5) %>%
    ggplot(aes(x = Gaps,
               y = Count,
               fill = Sample)) +
    theme_bw() +
    geom_bar(stat = "identity",
             position = "dodge") +
    labs(x = "Number of Gaps (minus 1) over the aligned length between 2 contigs",
         y = "Number of assembled contigs") +
        scale_fill_brewer(type = "qual", palette = 6)
```



Get the number of clam contigs from A, B, and C that align to the clam reference.

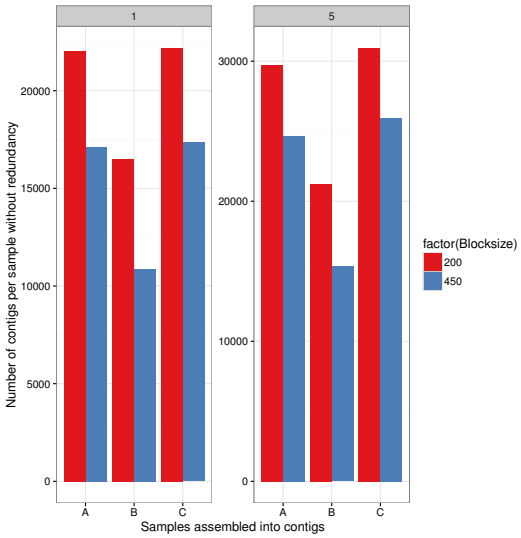
↰ One sample contig can align to several reference contigs. The latter are mainly length-isoforms of 1 gene.

- A contig can align entirely to a reference.
- A contig can align to 2 regions of a reference with 1 gap in between
- A contig can align to *n* regions of a reference with *n-1* gaps in between

The blocksize in the chart below is the length of the aligned region between contig and reference in case there is **no gap** (ie., 1) or there is **4 gaps** (ie., 5).

↰ We chose gaps=5 and blocksize=450

```
read.table("./data/gaps.blocksize.clam.txt", header = T) %>%
    ggplot(aes(x = Sample,
               y = Count,
               fill = factor(Blocksize))) +
    theme_bw() +
    geom_bar(stat = "identity",
             position = "dodge") +
    facet_wrap( ~ Gaps,
               ncol = 2,
               scales = "free") +
    labs(x = "Samples assembled into contigs",
         y = "Number of contigs per sample without redundancy") +
        scale_fill_brewer(type = "qual", palette = 6)
```



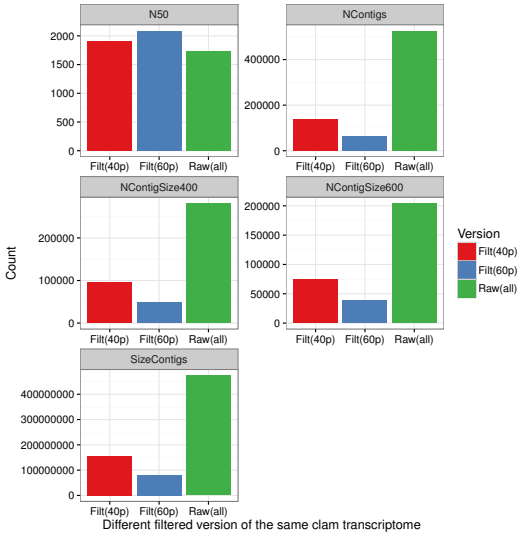
4.1 Preprocessing the clam transcriptome

A, B, and C reads were merged and used to assemble one transcriptome (previous study, not here). The raw output transcriptome done in Trinity was then reduced by 40% and 60% using different strategy of quality controls (not here). Transcriptome size, N50, and contig length are shown below for each transcriptome.

↰ We used the transcriptome reduced by 40 % only

↰ N50 is the mean length of contigs assembled from 50% of sequenced nucleotides

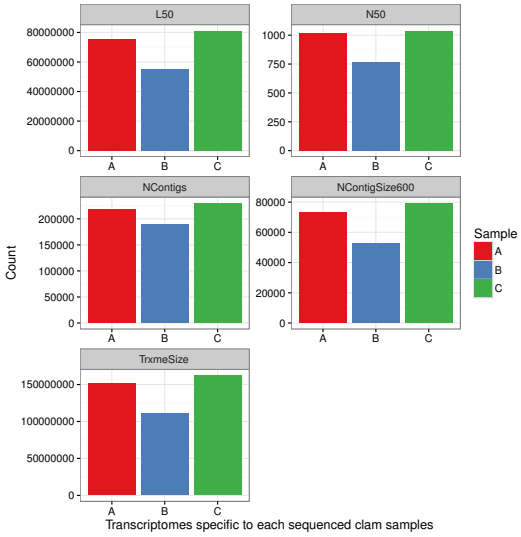
```
read.table("./data/clam.filtered.trxome.txt", header = T) %>%
    gather("Category", "Count", 2:6) %>%
    ggplot(aes(x = Version,
               y = Count,
               fill = Version)) +
        theme_bw() +
        geom_bar(stat = "identity",
                 position = "dodge") +
        facet_wrap( ~ Category,
                    ncol = 2,
                    scales = "free") +
        labs(x = "Different filtered version of the same clam transcriptome",
             y = "Count") +
    scale_fill_brewer(type = "qual", palette = 6)
```



From the selected transcriptome (40%) a guided Trinity assembly is done for each of sample which generated 3 separate clam transcriptomes. These 3 raw transcriptomes have lots of contigs, many of which can be discarded.

```
read.table("./data/clam.trxome.txt", header = T) %>%
```

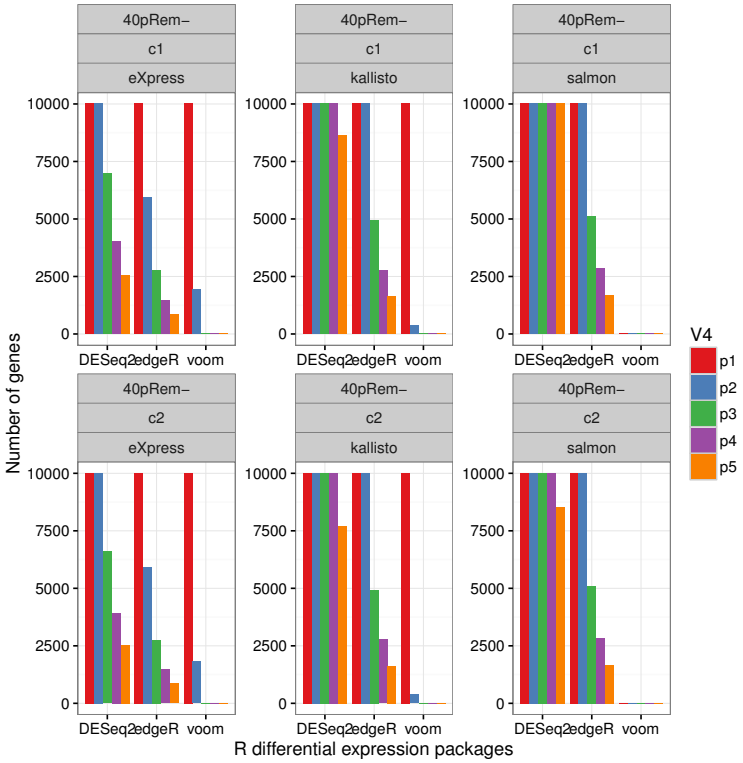
```
    gather("Category", "Count", 2:6) %>%
    ggplot(aes(x = Sample,
               y = Count,
               fill = Sample)) +
        theme_bw() +
        geom_bar(stat = "identity",
                 position = "dodge") +
        facet_wrap( ~ Category,
                    ncol = 2,
                    scales = "free") +
        labs(x = "Transcriptomes specific to each sequenced clam samples",
             y = "Count") +
    scale_fill_brewer(type = "qual", palette = 6)
```



4.2 Differential gene expression of the clam

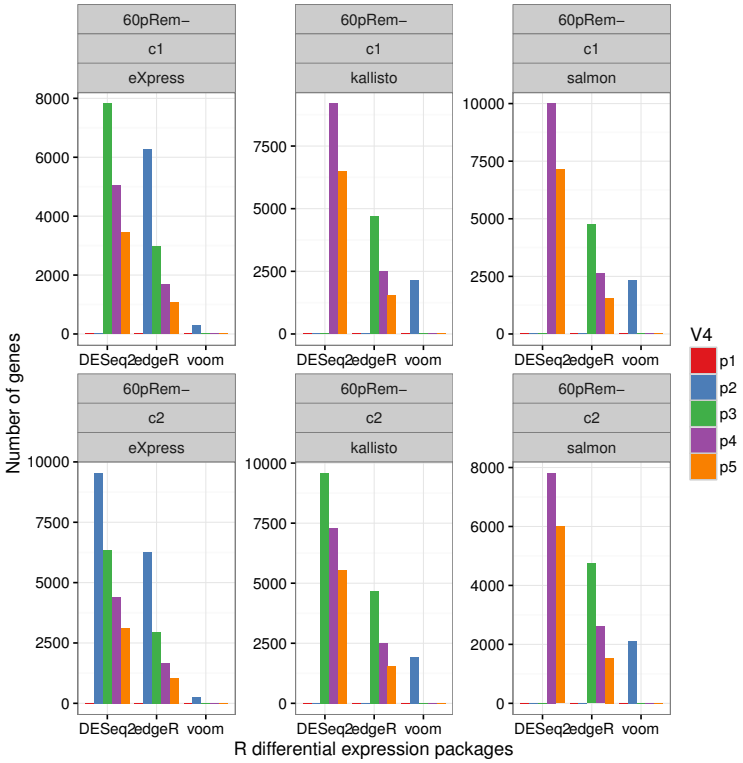
Three conditions were evaluated, 2 samples originated from infected animals but from different location, first a nodule and second a not inflamed tissue. 1 sample originated from a healthy animal. Also the transcriptome of the clam has been reduced in size by removing low aligned contigs and redundant isoforms. Accordingly 40% and 60% of the contigs have been removed.

```
read.table("./data/summary.40p.txt", header = F) %>%
    cbind(size = c("40pRem-")) %>%
    ggplot(aes(
    x = V1,
    y = V7,
    fill = V4)) +
    theme_bw() +
    geom_bar(stat = "identity",
             position = "dodge") +
    facet_wrap( ~ size + V5 + V2,
                ncol = 3,
                 scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    labs(x = "R differential expression packages",
         y = "Number of genes")
```



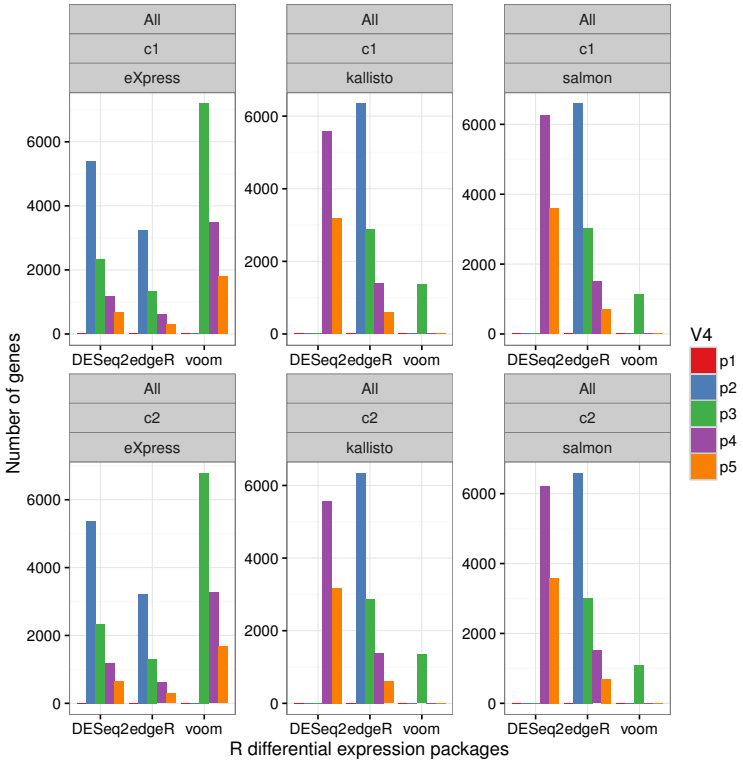
By removing 60% of the contigs.

```
read.table("./data/summary.60p.txt", header = F) %>%
    cbind(size = c("60pRem-")) %>%
    ggplot(aes(
    x = V1,
    y = V8,
    fill = V4)) +
    theme_bw() +
    geom_bar(stat = "identity",
             position = "dodge") +
    facet_wrap( ~ size + V5 + V2,
              ncol = 3,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    labs(x = "R differential expression packages",
         y = "Number of genes")
```



Without reducing clam transcriptome by removing unnecessary isoforms and low abundant reads.

```
read.table("./data/summary.100p.txt", header = F) %>%
    cbind(size = c("All")) %>%
    ggplot(aes(
    x = V1,
    y = V8,
    fill = V4)) +
    theme_bw() +
    geom_bar(stat = "identity",
             position = "dodge") +
    facet_wrap( ~ size + V5 + V2,
              ncol = 3,
               scales = "free") +
    scale_fill_brewer(type = "qual", palette = 6) +
    labs(x = "R differential expression packages",
         y = "Number of genes")
```



Distribution of differential expression between R packages.

```
require("yarrr")
df100 <- read.table("./data/summary.100p.txt", header = F) %>% cbind(size = c("All"))
df60 <- read.table("./data/summary.60p.txt", header = F) %>% cbind(size = c("60p"))
df40 <- read.table("./data/summary.40p.txt", header = F) %>% cbind(size = c("40p"))
pdf("./graphics/pirateplot.norm.pdf")
rbind(df100, df60, df40) %>%
#    pirateplot(formula = V7 ~ V4 + V2 + V1,
    pirateplot(formula = V8 ~ V2 + V1,
           theme = 2,
            pal = "basel",
            avg.line.o = .5,
            point.o = .35, point.pch = 21, point.bg = "white",
            point.col = "black", point.cex = 1.5,
            hdi.o = .7,
            inf.disp = "bean",
            bean.b.o = .3,
            jitter.val = .1)
dev.off()

X11cairo
       2
```

Distribution of differential expression between transcriptome sizes.

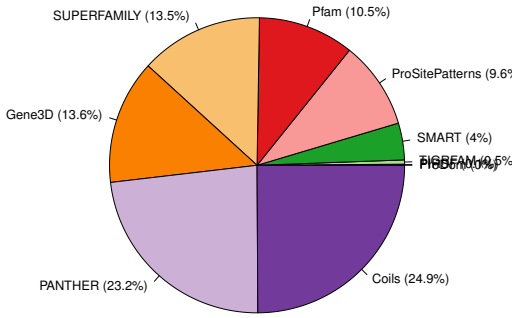
```
df100 <- read.table("./data/summary.100p.txt", header = F) %>% cbind(size = c("All"))
```

```
df60 <- read.table("./data/summary.60p.txt", header = F) %>% cbind(size = c("60p"))
df40 <- read.table("./data/summary.40p.txt", header = F) %>% cbind(size = c("40p"))
pdf("./graphics/pirateplot.size.pdf")
rbind(df100, df60, df40) %>%
    pirateplot(formula = V8 ~ V1 + size,
           theme = 2,
           pal = "basel",
           point.o = .35, point.pch = 21, point.bg = "white",
           point.col = "black", point.cex = 1.5,
           avg.line.o = .5,
           hdi.o = .7,
           bean.b.o = .4,
           inf.disp = "bean",
           jitter.val = .1)
dev.off()

X11cairo
       2
```

4.3 Annotation of selected differentially expressed genes

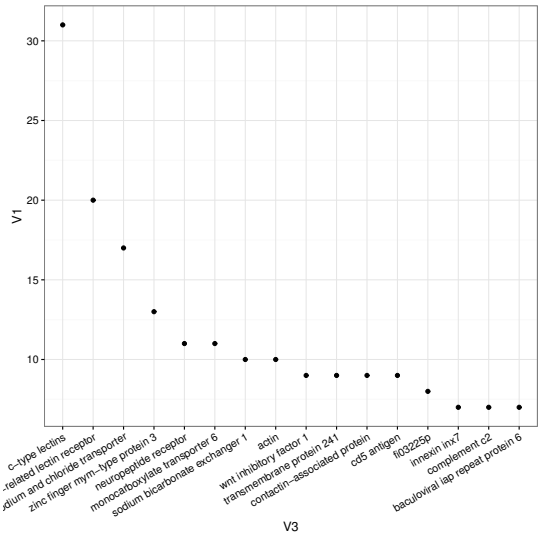
The differentially expressed genes will be selected from edgeR at an adjusted p-value of 10^{-3} with a fold change of 4 using the transcriptome that is reduced by only 40% (137,022 contigs). The annotated contigs are at an e-value of 10^{-10} from **protein domain alignment**.

```
palette <- brewer.pal(12, name = "Paired")
df <- read.table("./data/ips.clam.txt", header = F)
df$Vp <- apply(df[, 1, drop = F], 2, function(x){round((x/sum(df$V1))*100, digits = 1)})
colnames(df) <- c("counts", "db", "perc")
df$laby <- paste(df$db," (",df$perc,"%)",sep = "")
pie(x = df$counts, labels = df$laby, col = palette)
```



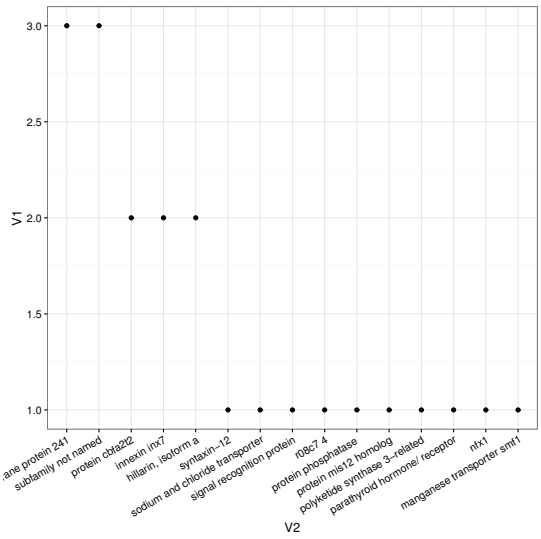
Most abundant species with domain annotations when comparing all samples simultaneously. This show clam genes.

```
df <- read.table("./data/clam_40p.functions.panther.len20.eval10.abc.txt", header = F, sep = "\t")
df$V3 <- factor(df$V3, levels = df[, 3])
ggplot(df, aes(x = V3,
               y = V1)) +
    theme_bw() +
    geom_point() +
    theme(axis.text.x=element_text(angle = 30, vjust = 1, hjust = 1))
```



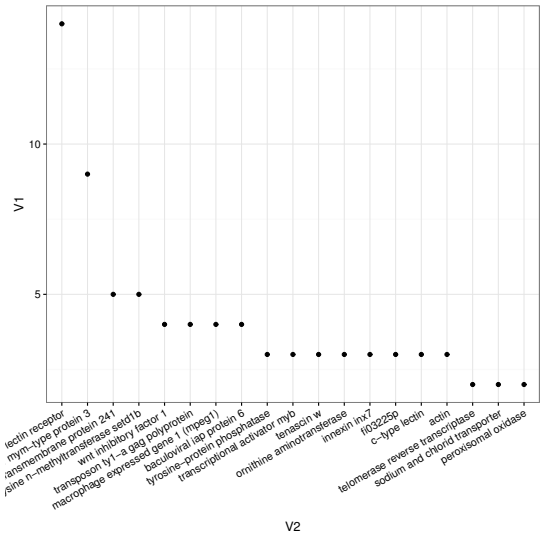
Most abundant species with domain annotations when comparing samples of infected animals only (A vs B). This show clam genes.

```
df <- read.table("./data/clam_40p.functions.panther.len20.eval10.ab.txt", header = F, sep = "\t")
df$V2 <- factor(df$V2, levels = df[, 2])
ggplot(df, aes(x = V2,
               y = V1)) +
    theme_bw() +
    geom_point() +
    theme(axis.text.x=element_text(angle = 30, vjust = 1, hjust = 1))
```



Most abundant species with domain annotations when comparing infected samples A versus healthy animals C. This show clam genes.

```
df <- read.table("./data/clam_40p.functions.panther.len20.eval10.ac.txt", header = F, sep = "\t")
df$V2 <- factor(df$V2, levels = df[, 2])
ggplot(df, aes(x = V2,
               y = V1)) +
    theme_bw() +
    geom_point() +
    theme(axis.text.x=element_text(angle = 30, vjust = 1, hjust = 1))
```



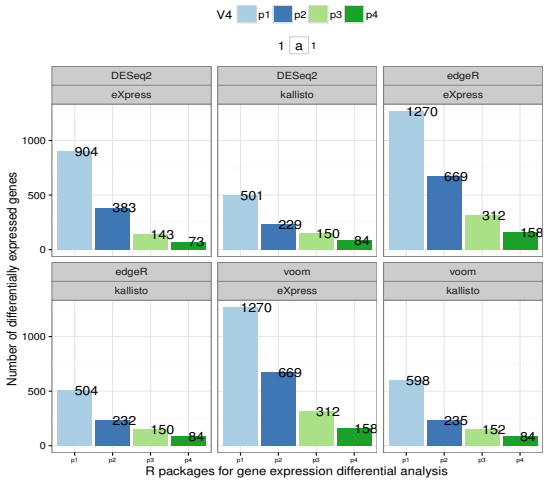
5 QPX genes remaining after cleanup

Raw reads of A, B, C samples are mapped to QPX reference genome. The true mapped reads are then assembled into contigs with Trinity genome guided approach. Number of genes resulted for each sample are shown in Section 3.1. These genes are then processed to select **932, 44, 0** genes/isoforms for QPX in sample A, B, and C respectively.

1. Remove redundant clam genes common with A, B, C clam assemblies
2. Remove PFAM domains with an e-value higher than 10^{-5}
3. Remove non-mapped contigs to QPX reference genome that have more than 4 gaps, less than 100 in contig size, and less than 250 in blocksize (if gaps exist)

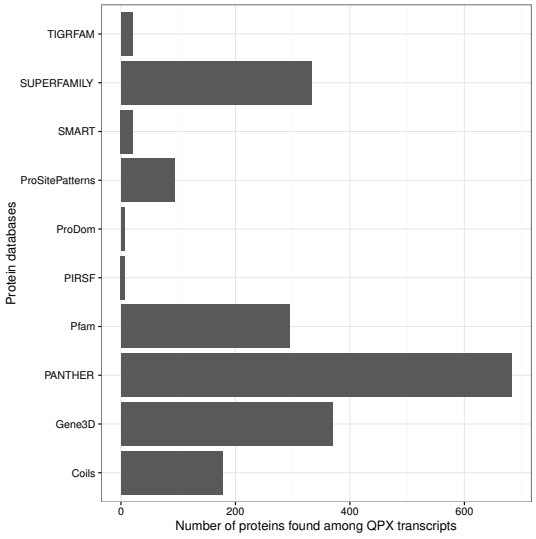
Differential gene expression between the three tissues at a minimum of 2 fold change.

```
read.table("./data/summary.eXpress.729521.txt") %>%
    ggplot(aes(
    x = V4,
    y = V8,
    fill = V4)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    geom_text(aes(x = V4,
                  y = V8,
                  ymax = V8,
                  label = V8,
                  size = 1,
                  hjust = 0),
              position = position_dodge(width = .5)) +
    facet_wrap(~ V1+V2,
               ncol = 3)+
    scale_fill_brewer(type = "qual", palette = "Paired") +
    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")
```



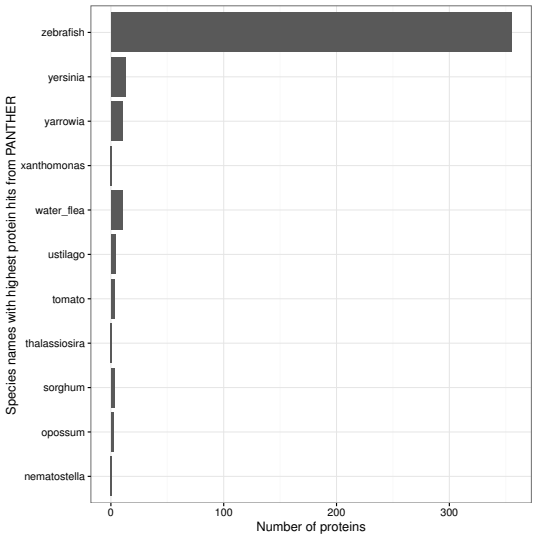
Annotations at an evalue of 10^{-10} of QPX transcripts assembled only from sample As. There is over 600 proteins found in PANTHER, but only 403 that have an alignment length of 20 aa between query and target.

```
read.table("./data/ips.txt", row.names = 1) %>%
    ggplot(aes(
    x = V3,
    y = V2)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    coord_flip() +
    theme_bw() +
    labs(x = "Protein databases",
         y = "Number of proteins found among QPX transcripts")
```



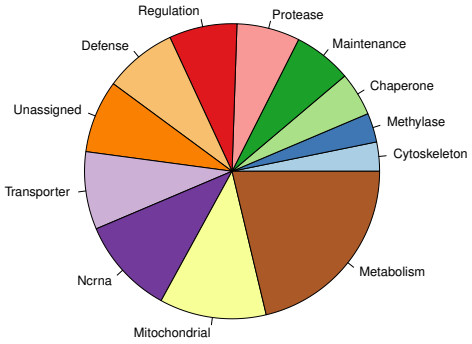
Distribution of species from PANTHER annotation at evalue of 10^{-10} and alignment 20 aa minimum.

```
read.table("./data/A.peptides.fa.tsv.PANspecies.LEN20.EVAL10.tab", row.names = 1) %>%
    ggplot(aes(
    x = V3,
    y = V2)) +
    geom_bar(stat = "identity",
             position = "dodge") +
    coord_flip() +
    theme_bw() +
    labs(x = "Species names with highest protein hits from PANTHER",
         y = "Number of proteins")
```



What are the enriched functional categories of the 312 differntially expressed genes between the QPX infected clam tissues and the healthy controls.

```
palette <- brewer.pal(12, name = "Paired")
df <- read.table("./data/degs.pie.txt", header = T, sep = "")
    pie(x = df$count, labels = df$fun, col = palette)
```



6 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 3.3.1 (2016-06-21)
Platform: x86_64-pc-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=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
 [1] WGCNA_1.51            fastcluster_1.1.20    dynamicTreeCut_1.63-1
 [4] gplots_3.0.1          reshape2_1.4.1        limma_3.30.12
 [7] yarrr_0.1.5           BayesFactor_0.9.12-2  coda_0.18-1
[10] jpeg_0.1-8            igraph_1.0.1          tidyr_0.5.1
[13] dplyr_0.5.0           latticeExtra_0.6-28   RColorBrewer_1.1-2
[16] glmnet_2.0-5          foreach_1.4.3         Matrix_1.2-6
[19] leaps_2.9             caret_6.0-70          ggplot2_2.1.0
[22] lattice_0.20-33       gdata_2.17.0          knitr_1.13

loaded via a namespace (and not attached):
 [1] Biobase_2.34.0        splines_3.3.1         gtools_3.5.0
 [4] Formula_1.2-1         assertthat_0.1        highr_0.6
 [7] stats4_3.3.1          impute_1.48.0         RSQLite_1.0.0
[10] quantreg_5.26         chron_2.3-47          digest_0.6.9
[13] minqa_1.2.4           colorspace_1.2-6      preprocessCore_1.36.0
[16] plyr_1.8.4            SparseM_1.7           GO.db_3.4.0
[19] mvtnorm_1.0-5         scales_0.4.0          lme4_1.1-12
[22] MatrixModels_0.4-1    tibble_1.0            mgcv_1.8-12
[25] IRanges_2.8.1         car_2.1-2             pbapply_1.2-1
[28] nnet_7.3-12           BiocGenerics_0.20.0   lazyeval_0.2.0
[31] pbkrtest_0.4-6        survival_2.39-5       magrittr_1.5
[34] evaluate_0.9          doParallel_1.0.10     nlme_3.1-128
[37] MASS_7.3-45           foreign_0.8-66        data.table_1.9.6
[40] tools_3.3.1           formatR_1.4           matrixStats_0.50.2
[43] stringr_1.0.0         S4Vectors_0.12.0      munsell_0.4.3
[46] cluster_2.0.4         AnnotationDbi_1.36.0  compiler_3.3.1
[49] caTools_1.17.1        grid_3.3.1            nloptr_1.0.4
[52] iterators_1.0.8       bitops_1.0-6          labeling_0.3
[55] gtable_0.2.0          codetools_0.2-14      DBI_0.4-1
[58] R6_2.1.2              gridExtra_2.2.1       Hmisc_3.17-4
[61] KernSmooth_2.23-15    stringi_1.1.1         parallel_3.3.1
[64] Rcpp_0.12.5           RevoUtils_10.0.1      rpart_4.1-10
[67] acepack_1.3-3.3
```