Sb to SB differences heatamp

Carlos Martinez Ruiz

18 January 2019

```
##                _
## platform       x86_64-pc-linux-gnu
## arch           x86_64
## os             linux-gnu
## system         x86_64, linux-gnu
## status
## major          3
## minor          4.4
## year           2018
## month          03
## day            15
## svn rev        74408
## language       R
## version.string R version 3.4.4 (2018-03-15)
## nickname       Someone to Lean On
```

```
## [1] "Package plyr version 1.8.4"
## [1] "Package ggplot2 version 3.0.0"
## [1] "Package GenomicFeatures version 1.26.4"
## [1] "Package AnnotationDbi version 1.36.2"
## [1] "Package DESeq2 version 1.14.1"
## [1] "Package SummarizedExperiment version 1.4.0"
## [1] "Package Biobase version 2.34.0"
## [1] "Package GenomicRanges version 1.26.4"
## [1] "Package GenomeInfoDb version 1.10.3"
## [1] "Package IRanges version 2.8.2"
## [1] "Package S4Vectors version 0.12.2"
## [1] "Package BiocGenerics version 0.20.0"
## [1] "Package parallel version 3.4.4"
## [1] "Package stats4 version 3.4.4"
## [1] "Package readr version 1.1.1"
## [1] "Package tximport version 1.2.0"
## [1] "Package stats version 3.4.4"
## [1] "Package graphics version 3.4.4"
## [1] "Package grDevices version 3.4.4"
## [1] "Package utils version 3.4.4"
## [1] "Package datasets version 3.4.4"
## [1] "Package methods version 3.4.4"
## [1] "Package base version 3.4.4"
```

Load the data from the Sb vs SB allele specific comparison for both North American and South American populations, extract read counts.

```
load("input/dds_Sb_vs_SB_north_america.RData")
load("input/dds_Sb_vs_SB_south_america.RData")

#Get normalised counts
dds_north_america <- estimateSizeFactors(dds_Bb_DE)
dds_south_america <- estimateSizeFactors(dds_deg_ar)
```

```
counts_north_america <- counts(dds_north_america, normalized = TRUE)
counts_south_america <- counts(dds_south_america, normalized = TRUE)
```

How many genes do both datasets have in common?

```
genes_south_america <- row.names(counts_south_america)
genes_north_america <- row.names(counts_north_america)

genes_both <- genes_north_america[genes_north_america %in% genes_south_america]

#Number of genes present in both datsets:
length(genes_both)
```

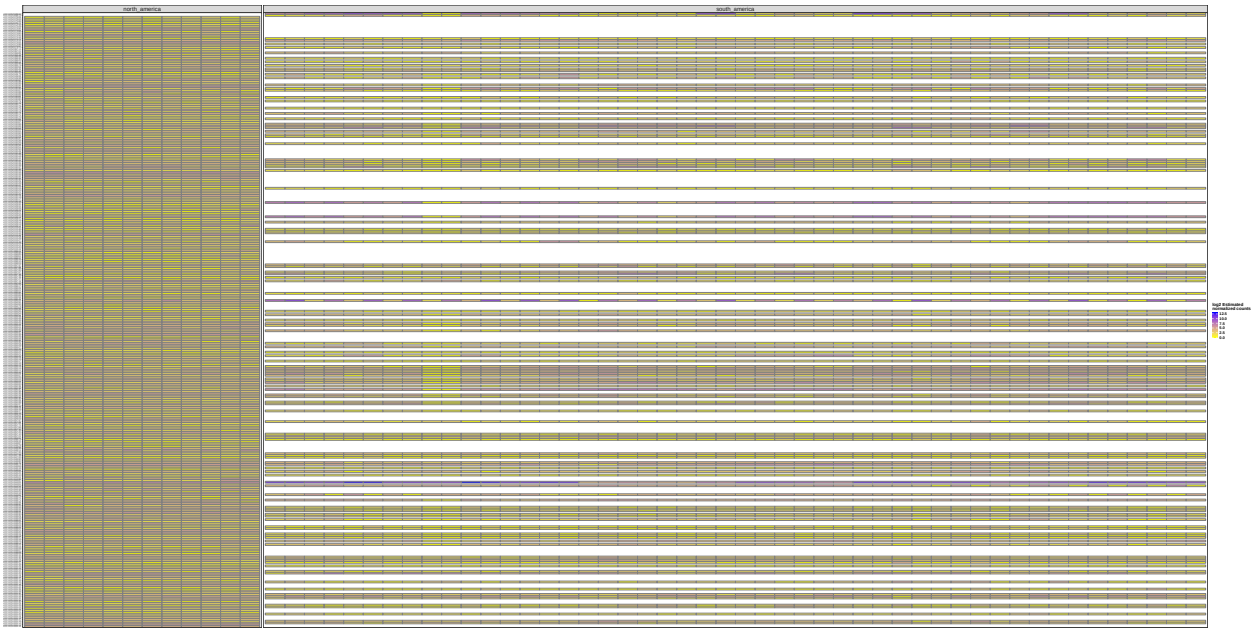
```
## [1] 123
```

123 out of 125 genes found with fixed differences in South America are also present in North America

Plot heatmap

Heatmap with raw read counts

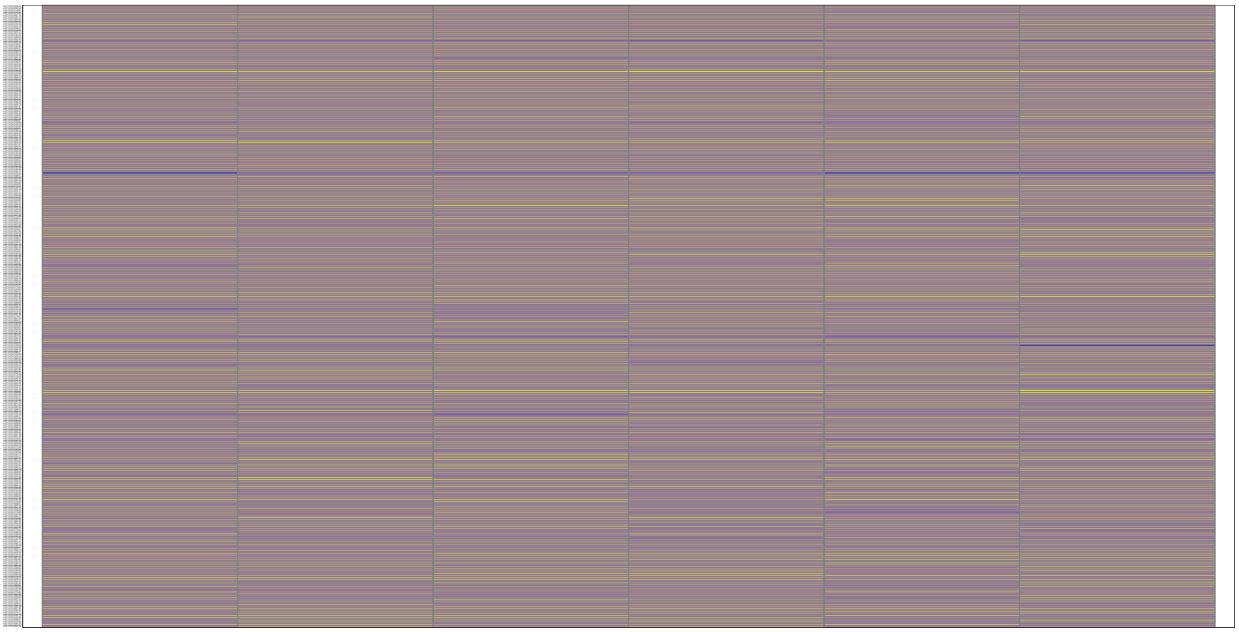
```
#Parse the dataset to make it ggplot friendly
samples_north_america <- colnames(counts_north_america)
parsed_counts_north_america <- data.frame(c(counts_north_america),
                                          rep(samples_north_america, each = nrow(counts_north_america)),
                                          rep(genes_north_america, ncol(counts_north_america)),
                                          rep("north_america", (nrow(counts_north_america) * ncol(counts
                                          )
colnames(parsed_counts_north_america) <- c("counts", "sample", "gene", "population")

samples_south_america <- colnames(counts_south_america)
parsed_counts_south_america <- data.frame(c(counts_south_america),
                                          rep(samples_south_america, each = nrow(counts_south_america)),
                                          rep(genes_south_america, ncol(counts_south_america)),
                                          rep("south_america", (nrow(counts_south_america) * ncol(counts
                                          )
colnames(parsed_counts_south_america) <- c("counts", "sample", "gene", "population")


#Merge both datasets
parsed_counts_all <- rbind(parsed_counts_north_america, parsed_counts_south_america)
parsed_counts_all$allele <- gsub(x = parsed_counts_all$sample, pattern = ".+_", replacement = "")
parsed_counts_all$allele <- ifelse(parsed_counts_all$allele == "B" , "bigB", "littleb")
parsed_counts_all <- parsed_counts_all[order(parsed_counts_all$allele, parsed_counts_all$sample), ]

#Plot heatmap
ggplot(parsed_counts_all, aes(x = sample, y = gene)) + geom_tile(aes(fill = log2(counts + 1)), colour =
  facet_grid(. ~ population, scales = "free", space = "free") +
  scale_fill_gradient(low = "yellow" , high = "blue", name="log2 Estimated \nnormalized counts") +
  theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x=element_blank(), axis.ticks
  theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
  theme(strip.text.x = element_text(size=20)) + theme(legend.title = element_text(size=15, face="bold")
                                                      legend.text = element_text(size=13, face="bold"))
```



Heatmap with logarithm of the read count ratios between variants for North American populations

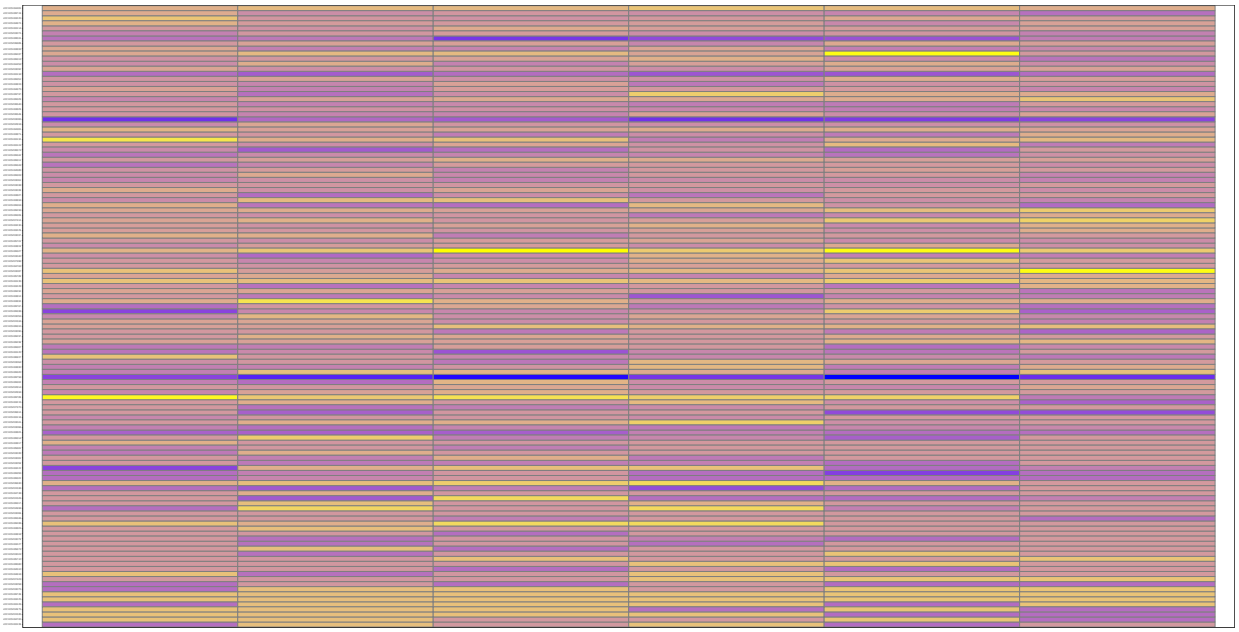
```
#Parse the dataset to make it ggplot friendly
#Separate big B and little b samples
samples_north_america_littleb <- grep(x = samples_north_america, pattern = "_b", value = TRUE)
samples_north_america_bigB <- grep(x = samples_north_america, pattern = "_B", value = TRUE)
samples_north_america_no_allele <- unique(gsub(x = samples_north_america, pattern = "_[Bb]", replacement

counts_north_america_littleb <- counts_north_america[, samples_north_america_littleb]
counts_north_america_bigB <- counts_north_america[, samples_north_america_bigB]


parsed_counts_north_america_ratio <- data.frame(c(counts_north_america_littleb),
                                                c(counts_north_america_bigB),
                                                rep(samples_north_america_no_allele, each = nrow(counts_
                                                rep(genes_north_america, length(samples_north_america_no
                                                rep("north_america", (nrow(counts_north_america) * length
                                                )
colnames(parsed_counts_north_america_ratio) <- c("littleb", "bigB", "sample", "gene", "population")

parsed_counts_north_america_ratio$total_counts <- parsed_counts_north_america_ratio$littleb + parsed_cou
parsed_counts_north_america_ratio$lfc <- log2((parsed_counts_north_america_ratio$bigB +1) / (parsed_coun

#Order the genes by mean read count
mean_reads_north_america <- ddply(parsed_counts_north_america_ratio, .(gene), summarize, mean =
                                    mean(total_counts))
mean_reads_north_america <- mean_reads_north_america[order(mean_reads_north_america$mean), ]
gene_order_north_america <- rep(mean_reads_north_america$gene,
                                length(unique(samples_north_america_no_allele)))

#Use the sorted genes to sort the parsed counts
parsed_counts_north_america_ratio$gene <- factor(parsed_counts_north_america_ratio$gene, levels = mean_r

#Plot heatmap for North America
```

```
ggplot(parsed_counts_north_america_ratio, aes(x = sample, y = gene)) + geom_tile(aes(fill = lfc), colou
       scale_fill_gradient(low = "yellow" , high = "blue", name="lfc") +
       theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x=element_blank(), axis.
       theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
       theme(strip.text.y = element_text(size=.5)) + theme(legend.title = element_text(size=15, face="b
                                                           legend.text = element_text(size=13, face="bo
```



Same plot for genes present only in both populations

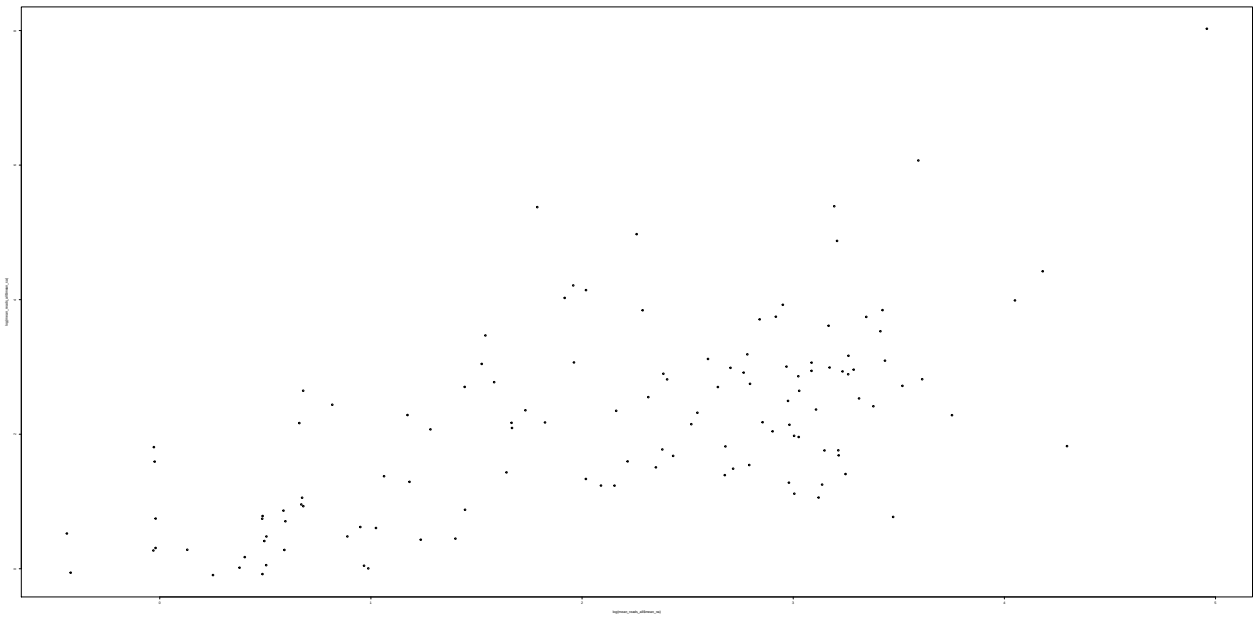
```
parsed_counts_north_america_ratio_subset <- parsed_counts_north_america_ratio[parsed_counts_north_ameri

ggplot(parsed_counts_north_america_ratio_subset, aes(x = sample, y = gene)) + geom_tile(aes(fill = lfc)
       scale_fill_gradient(low = "yellow" , high = "blue", name="lfc") +
       theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x=element_blank(), axis.
       theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
       theme(strip.text.y = element_text(size=.5)) + theme(legend.title = element_text(size=15, face="b
                                                           legend.text = element_text(size=13, face="bo
```



Comparison of mean read counts between South American and North American populations

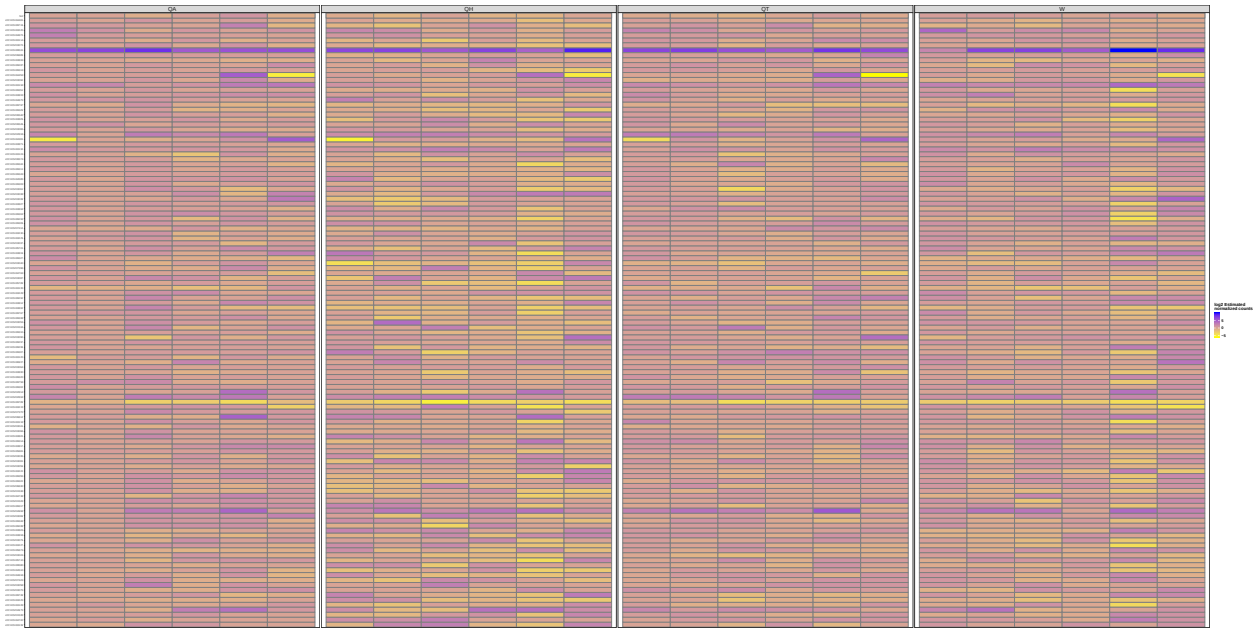
```
#Parse the dataset to make it ggplot friendly
#Separate big B and little b samples
samples_south_america_littleb <- grep(x = samples_south_america, pattern = "_b", value = TRUE)
samples_south_america_bigB <- grep(x = samples_south_america, pattern = "_B", value = TRUE)
samples_south_america_no_allele <- unique(gsub(x = samples_south_america, pattern = "_[Bb]", replacement

counts_south_america_littleb <- counts_south_america[, samples_south_america_littleb]
counts_south_america_bigB <- counts_south_america[, samples_south_america_bigB]

parsed_counts_south_america_ratio <- data.frame(c(counts_south_america_littleb),
                                                c(counts_south_america_bigB),
                                                rep(samples_south_america_no_allele,
                                                    each =nrow(counts_south_america)),
                                                rep(genes_south_america,
                                                    length(samples_south_america_no_allele)),
                                                rep("south_america",
                                                    (nrow(counts_south_america) * length(samples_south_a
                                                )
colnames(parsed_counts_south_america_ratio) <- c("littleb", "bigB", "sample", "gene", "population")

parsed_counts_south_america_ratio$total_counts <- parsed_counts_south_america_ratio$littleb + parsed_cou
parsed_counts_south_america_ratio$lfc <- log2((parsed_counts_south_america_ratio$bigB +1) / (parsed_coun

#Get mean read count for South American samples
mean_reads_south_america <- ddply(parsed_counts_south_america_ratio, .(gene), summarize, mean =
                                  mean(total_counts))
mean_reads_south_america <- mean_reads_south_america[order(mean_reads_south_america$mean), ]

#Get the list of genes in North America that overlap with South America
mean_reads_all <- merge(mean_reads_north_america, mean_reads_south_america, by = "gene")
colnames(mean_reads_all) <- c("gene", "mean_na", "mean_sa")
```

```r
#Plot means per population
plot(log(mean_reads_all$mean_na),
     log(mean_reads_all$mean_sa))
```



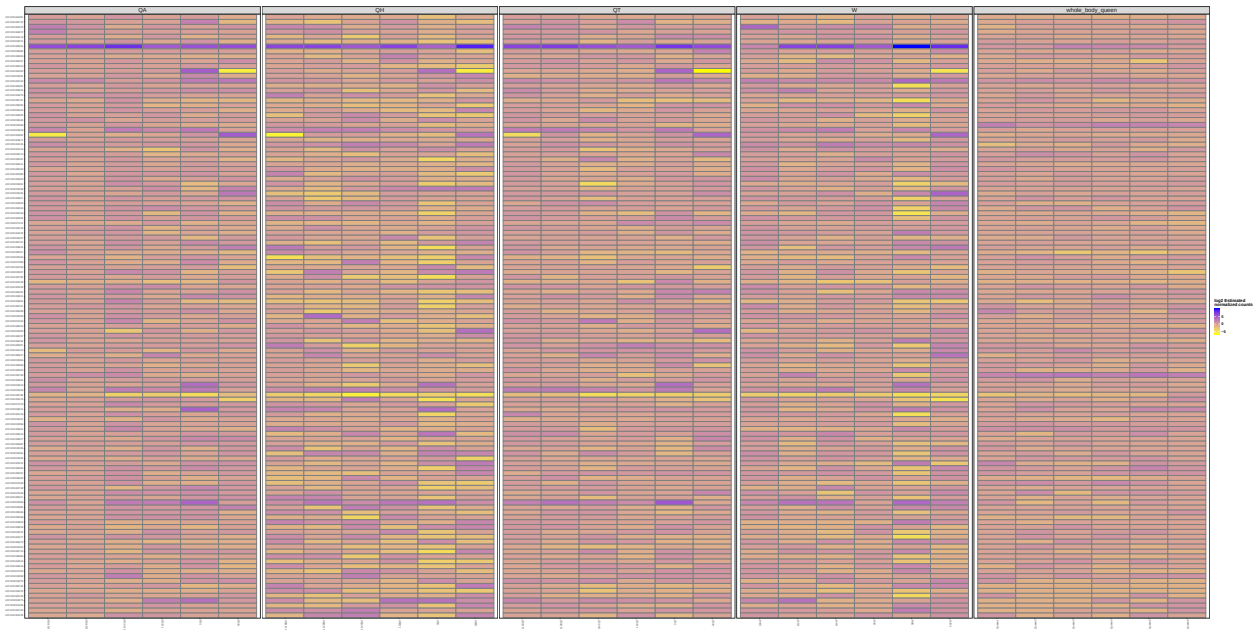
Mean read counts across populations is roughly similar. For next plots, the order for the North American dataset will be kept. Heatmap with logarithm of the read count ratios between variants for South American populations

```r
#Use the sorted genes from the North America dataset to sort the parsed counts
mean_reads_all_sorted <- mean_reads_all[order(mean_reads_all$mean_na), ]
parsed_counts_south_america_ratio$gene <- factor(parsed_counts_south_america_ratio$gene, levels = mean_

#Add body part as a factor
parsed_counts_south_america_ratio$body_part <- gsub(x = parsed_counts_south_america_ratio$sample,
                                                    pattern = "([0-9]+[BC]?)([A-Z]+)", replacement = "\

#Plot heatmap for South America
ggplot(parsed_counts_south_america_ratio, aes(x = sample, y = gene)) + geom_tile(aes(fill = lfc), colou
       facet_grid(. ~ body_part, scales = "free") +
       scale_fill_gradient(low = "yellow" , high = "blue", name="log2 Estimated \nnormalized counts") +
       theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x=element_blank(), axis.
       theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
       theme(strip.text.x = element_text(size=20)) + theme(legend.title = element_text(size=15, face="b
                                                           legend.text = element_text(size=13, face="bo
```



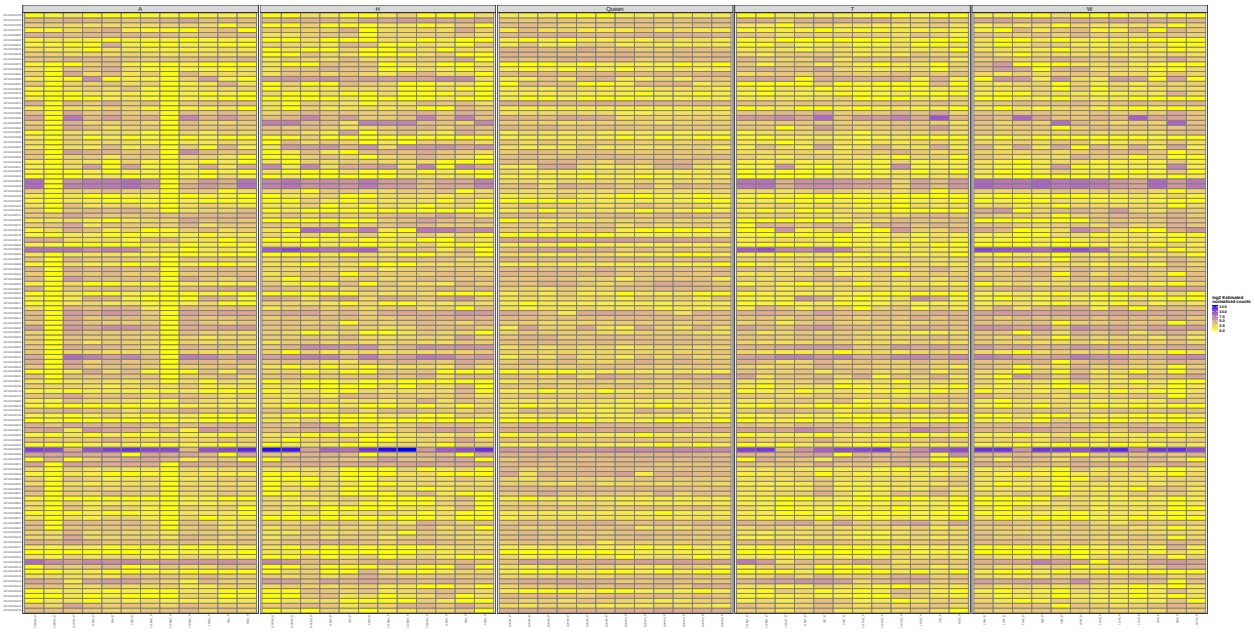
Plot heatmap for genes present only in both datasets

```
#Ensure the datasets for both populations have the same genes.
parsed_counts_north_america_ratio$body_part <- rep("whole_body_queen", nrow(parsed_counts_north_america_
parsed_counts_all_ratios_common_genes <- rbind(parsed_counts_north_america_ratio, parsed_counts_south_am
parsed_counts_all_ratios_common_genes <- parsed_counts_all_ratios_common_genes[parsed_counts_all_ratios_

ggplot(parsed_counts_all_ratios_common_genes, aes(x = sample, y = gene)) + geom_tile(aes(fill = lfc), co
       facet_grid(. ~ body_part, scales = "free", space = "free") +
       scale_fill_gradient(low = "yellow" , high = "blue", name="log2 Estimated \nnormalized counts") +
       theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x = element_text(angle =
       theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
       theme(strip.text.x = element_text(size=20)) + theme(legend.title = element_text(size=15, face="bo
                                                           legend.text = element_text(size=13, face="bol
```



Plot heatmap by read counts for genes present in both populations

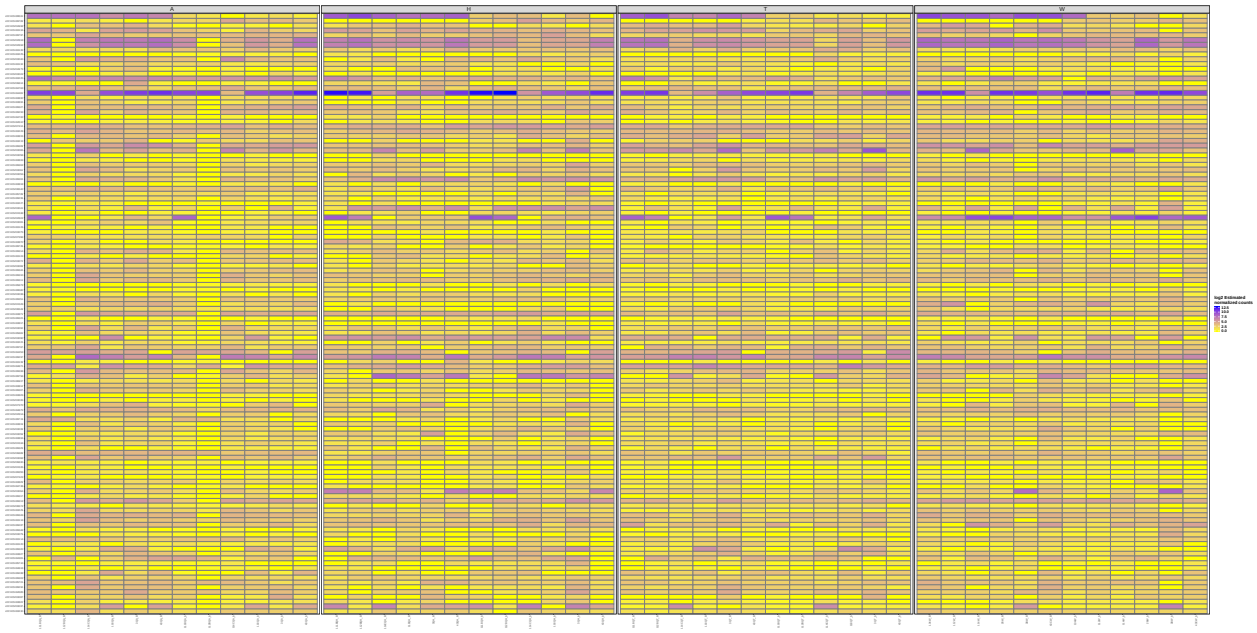
```
#Ensure the datasets for both populations have the same genes.
parsed_counts_all_common_genes <- parsed_counts_all[parsed_counts_all$gene %in% genes_both, ]

#Add a body_part factor
parsed_counts_all_common_genes$body_part <- gsub(x = parsed_counts_all_common_genes$sample,
                                                 pattern = "([0-9]+[BC]?Q?)([A-Z])", replacement = "\\2
parsed_counts_all_common_genes$body_part <- gsub(x = parsed_counts_all_common_genes$body_part,
                                                 pattern = "[0-9]?_[Bb]", replacement = "")

#Make sure the heatmap is plotted ordered by allele
levels_bigB <- grep("_B", levels(parsed_counts_all_common_genes$sample), value = TRUE)
levels_littleb <- grep("_b", levels(parsed_counts_all_common_genes$sample), value = TRUE)
all_levels <- c(levels_bigB, levels_littleb)
parsed_counts_all_common_genes$sample <- factor(parsed_counts_all_common_genes$sample,
                                                levels = all_levels)

#Get the names for the x lables in the heatmap (SB vs Sb)
allele_names_x_axis <- gsub(x = all_levels, pattern = ".+_", replacement = "")

ggplot(parsed_counts_all_common_genes, aes(x = sample, y = gene)) + geom_tile(aes(fill = log2(counts +
  facet_grid(. ~ body_part, scales = "free", space = "free") +
  scale_fill_gradient(low = "yellow" , high = "blue", name="log2 Estimated \nnormalized counts") +
  theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x = element_text(angle = 90,
  theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
  theme(strip.text.x = element_text(size=20)) + theme(legend.title = element_text(size=15, face="bold")
                                                      legend.text = element_text(size=13, face="bold"))
```



Plot each population individually, ordering genes by p value For South America

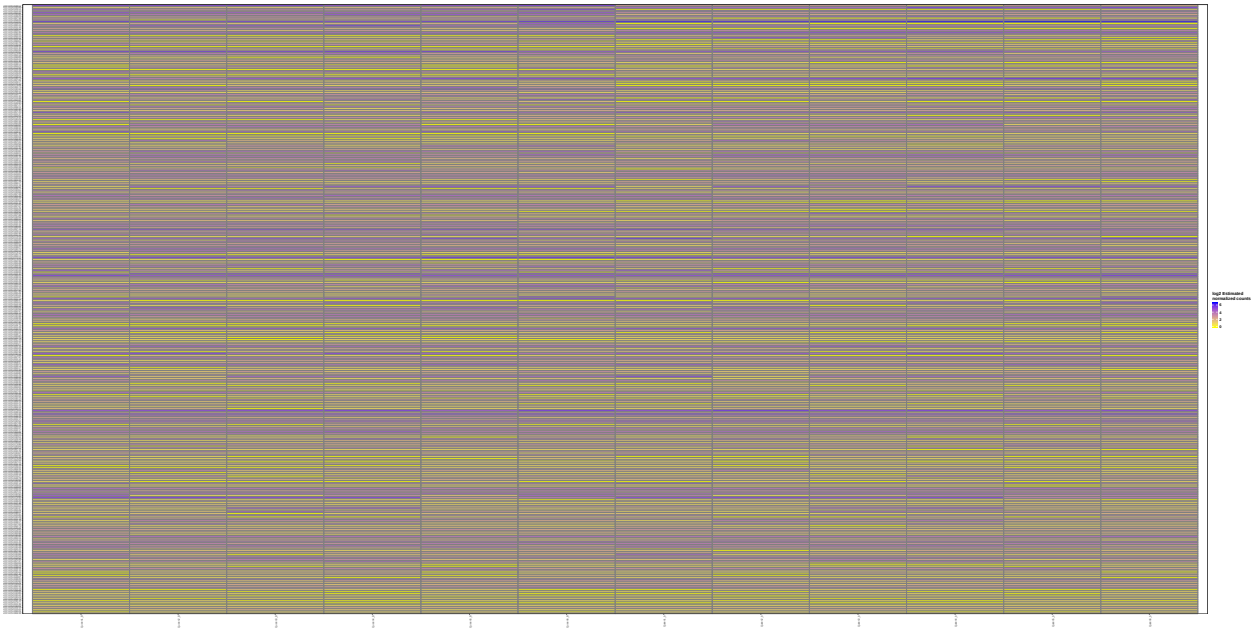
```
#Get p values for all genes
results_north_america <- results(dds_Bb_DE)
results_south_america <- results(dds_deg_ar)
```

```r
#Order genes by pvalues
results_north_america <- results_north_america[order(results_north_america$pvalue, decreasing = TRUE), ]
results_south_america <- results_south_america[order(results_south_america$pvalue, decreasing = TRUE), ]

#Get the names for the x lables in the heatmap (SB vs Sb)
gene_order_pval_north_america <- rownames(results_north_america)
gene_order_pval_south_america <- rownames(results_south_america)

#Add body_part as a factor
parsed_counts_south_america$body_part <- gsub(x = parsed_counts_south_america$sample,
                                              pattern = "([0-9]+[BC]?Q?)([A-Z])", replacement = "\\2")

parsed_counts_south_america$body_part <- gsub(x = parsed_counts_south_america$body_part,
                                              pattern = "_.+", replacement = "")

#Get the right order for genes and samples
parsed_counts_south_america$sample <- factor(parsed_counts_south_america$sample,
                                             levels = all_levels)

parsed_counts_south_america$gene <- factor(parsed_counts_south_america$gene,
                                           levels = gene_order_pval_south_america)

#Plot the heatmap
ggplot(parsed_counts_south_america, aes(x = sample, y = gene)) + geom_tile(aes(fill = log2(counts + 1))
       facet_grid(. ~ body_part, scales = "free", space = "free") +
       scale_fill_gradient(low = "yellow" , high = "blue", name="log2 Estimated \nnormalized counts") +
       theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x = element_text(angle =
       theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
       theme(strip.text.x = element_text(size=20)) + theme(legend.title = element_text(size=15, face="b
                                                           legend.text = element_text(size=13, face="bold"))
```



For North America

```
#Get the right order for genes and samples
parsed_counts_north_america$sample <- factor(parsed_counts_north_america$sample,
                                             levels = all_levels)

parsed_counts_north_america$gene <- factor(parsed_counts_north_america$gene,
                                           levels = gene_order_pval_north_america)

#Plot the heatmap
ggplot(parsed_counts_north_america, aes(x = sample, y = gene)) + geom_tile(aes(fill = log2(counts + 1))
       scale_fill_gradient(low = "yellow" , high = "blue", name="log2 Estimated \nnormalized counts") +
       theme_bw() + theme( panel.grid.major=element_blank()) + theme(axis.text.x = element_text(angle =
       theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
       theme(strip.text.x = element_text(size=20)) + theme(legend.title = element_text(size=15, face="b
                                                           legend.text = element_text(size=13, face="bold"))
```



Fisher's combined probability test

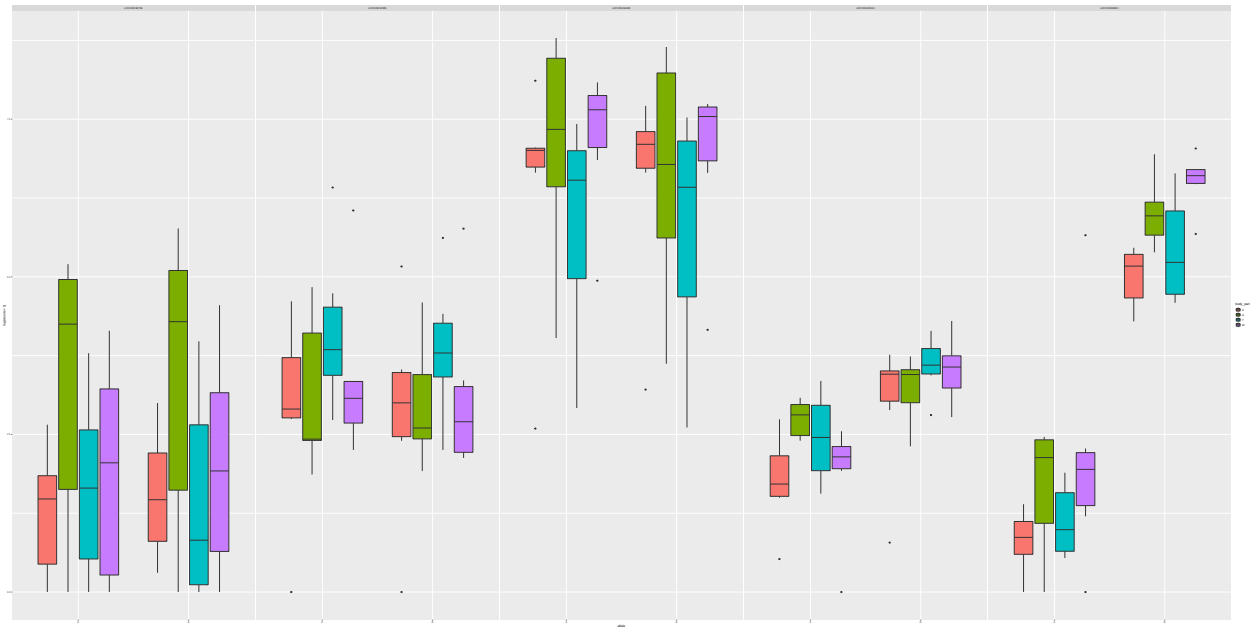
Because it is difficult to compare North and South American p values directly, the results of both tests will be fused together using (Fisher's method)https://en.wikipedia.org/wiki/Fisher%27s_method. This method will test whether the combined p value of both analyses is significant.

```
#Obtain p values for genes present only in both datasets
results_north_america_subset <- results_north_america[genes_both, ]
results_south_america_subset <- results_south_america[genes_both, ]

#Get a vector of X2 values based on the combination of p values of both datasets
combined_ps <- -2 * log((results_north_america_subset$pvalue + results_north_america_subset$pvalue))

#Obtain new p values based on X2 distribution
new_ps <- pchisq(combined_ps, df = 4, lower.tail = FALSE)

#Adjust p values using the Benjamini and Hochberg method
new_ps_adjust <- p.adjust(new_ps, method = "BH")
```

```r
#Because the genes where in the same order, the new p values can be named directly:
names(new_ps_adjust) <- genes_both

#Retrieve significant genes from the combined p values
sig_genes_both <- names(new_ps_adjust[new_ps_adjust < 0.05])
```

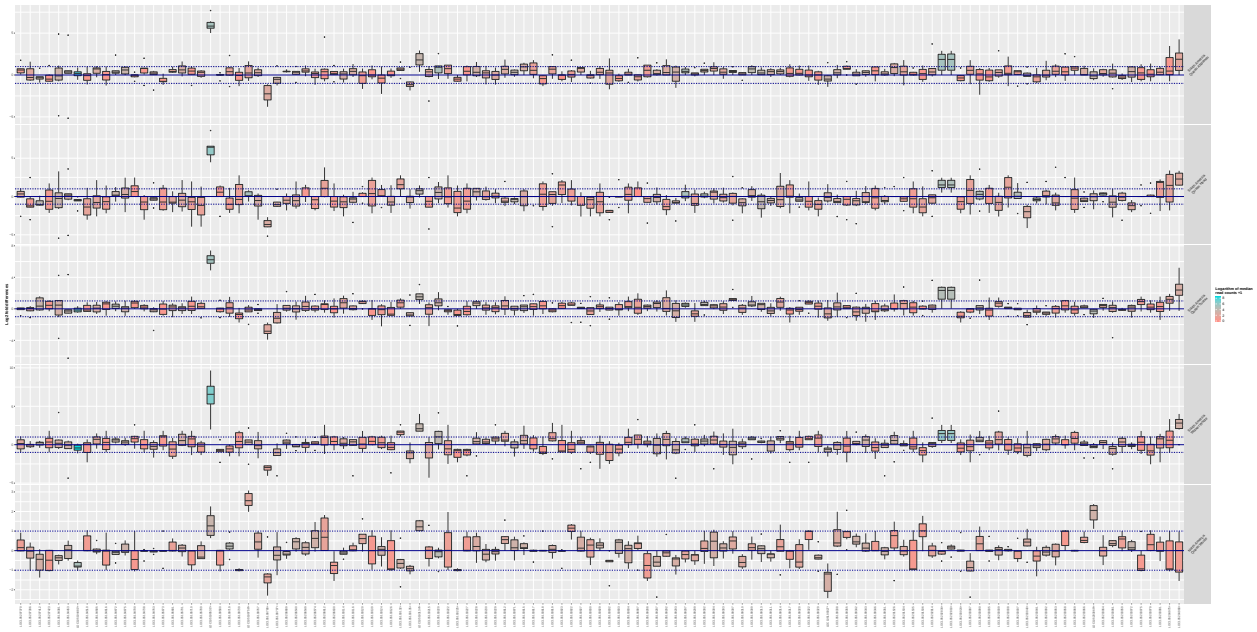
Re-run the DESeq2 analysis for the allele:body_part interaction, but enriched with genes differentially expressed in both populations.

```r
#Retrieve colData from the DESeq dataset and subset for genes of interest
colData_subset <- colData(dds_deg_ar)

#Subset dataset to include only genes of interest
raw_counts_south_america_subset <- counts(dds_deg_ar, normalized = FALSE)[sig_genes_both, ]

#Re-run DESeq2
dds_ar_interaction_subset <- DESeqDataSetFromMatrix(countData = raw_counts_south_america_subset, colDat
#Perform the analysis replacing the sizeFactors:
sizeFactors(dds_ar_interaction_subset) <- rep(1, ncol(raw_counts_south_america_subset))
#If using test = LRT, deseq2 performs a test for detecting DE loci, using first a likelihood ratio test
#reduced model, ~body_part. The p values indicate which genes are significantly DE accross ALL levels o
#has been done only by boy_part, as ASE data comes always from the same sample. If test is not specifie
dds_deg_ar_interaction_subset <- DESeq(dds_ar_interaction_subset, test = "LRT", reduced = ~ colony + bo
res_interaction_subset <- results(dds_deg_ar_interaction_subset)

#Plot the genes of interest for each allele
subset_parsed_counts_south_america <- parsed_counts_south_america[parsed_counts_south_america$gene %in%

subset_parsed_counts_south_america$allele <- gsub(x = subset_parsed_counts_south_america$sample,
                                                  pattern = ".+_",
                                                  replacement = "")

ggplot(subset_parsed_counts_south_america, aes(x = allele, y = log(counts +1), fill = body_part)) + geo
  facet_grid(. ~ gene)
```



```
#Calculate median read counts per group (gene and body part)
agg_medians <- aggregate(total_counts ~ gene + body_part, data = parsed_counts_all_ratios_common_genes,
agg_medians <- agg_medians[order(agg_medians$gene, agg_medians$body_part), ]
colnames(agg_medians) <- c("gene", "body_part", "medians")

#Add medians to the main dataset
parsed_counts_all_ratios_common_genes <- merge(agg_medians, parsed_counts_all_ratios_common_genes)

#Get the order of all genes by position in the supergene
#Load the annotation for the gnG assembly of the Solenopsis invicta reference genome.
si_ann <- makeTxDbFromGFF(file = "input/GCF_000188075.1_Si_gnG_genomic.gff",
                          format="gff3")

#Generate a table with the gene names and its position in the reference
gene_ids <- keys(si_ann, "GENEID")
gene_positions <- AnnotationDbi::select(x = si_ann,
                                        keys = gene_ids,
                                        columns = c("GENEID","TXCHROM", "TXSTART", "TXEND"),
                                        keytype = "GENEID")

colnames(gene_positions) <- c("gene","contig", "start", "end")

#Get only the minimum start position in the scaffolds for all the transcripts per gene
gene_positions_min <- aggregate(x = gene_positions$start, by = list(gene_positions$contig, gene_positio
names(gene_positions_min) <- c("contig", "gene", "start_position")

#Select genes only in the analysis
gene_positions_min <- gene_positions_min[gene_positions_min$gene %in% unique(parsed_counts_all_ratios_c

#Sort by start position per contig
gene_positions_min <- gene_positions_min[order(gene_positions_min$contig, gene_positions_min$start_posi

#Mark the genes with significant DE in the different populations and in both
sig_genes_na <- rownames(results_north_america_subset[which(results_north_america_subset$padj < 0.05),
```

```r
sig_genes_sa <- rownames(results_south_america_subset[which(results_south_america_subset$padj < 0.05), ]

parsed_counts_all_ratios_common_genes$gene <- as.character(parsed_counts_all_ratios_common_genes$gene)

parsed_counts_all_ratios_common_genes$gene[parsed_counts_all_ratios_common_genes$gene %in% sig_genes_bo

parsed_counts_all_ratios_common_genes$gene[parsed_counts_all_ratios_common_genes$gene %in% sig_genes_na

parsed_counts_all_ratios_common_genes$gene[parsed_counts_all_ratios_common_genes$gene %in% sig_genes_sa

#Mark in both datasets
gene_positions_min$gene[gene_positions_min$gene %in% sig_genes_both] <- paste0(gene_positions_min$gene[

gene_positions_min$gene[gene_positions_min$gene %in% sig_genes_na] <- paste0(gene_positions_min$gene[ge

gene_positions_min$gene[gene_positions_min$gene %in% sig_genes_sa] <- paste0(gene_positions_min$gene[ge

#Sort the levels in the main dataframe
parsed_counts_all_ratios_common_genes$gene <- factor(parsed_counts_all_ratios_common_genes$gene,
                                                     levels = gene_positions_min$gene)

facet_labels <- c(QA = "South America\nQueen Abdomen", QH = "South America\nQueen Head",
                  QT = "South America\nQueen Thorax", W = "South America\nWorker Whole",
                  whole_body_queen = "North America\nQueen Whole")
ggplot(data = parsed_counts_all_ratios_common_genes, aes(x = gene, y = lfc)) + geom_boxplot(aes(fill = 
                                                                                              outlier.size
  scale_fill_gradient(name = "Logarithm of median\nread counts +1", low = hcl(15,100,75), high = hcl(195
  facet_grid(rows = vars(body_part), scales = "free", labeller = labeller(body_part = facet_labels)) +
  geom_hline(yintercept = 0, color = "darkblue") + geom_hline(yintercept = -1, color = "darkblue", linet
  geom_hline(yintercept = 1, color = "darkblue", linetype = "dashed") + labs(y = "Log2 fold differences
  #geom_rect(aes(xmin = sig_genes_both[1], xmax = sig_genes_both[1], ymin = -8, ymax = 8),
  #          fill = "transparent", color = "red", size = 1.5) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 10), axis.title.x = element_blank(),
        axis.text.y = element_text(vjust = 0.5, size = 12), axis.title.y = element_text(size = 14, face
        strip.text.y= element_text(size = 12, angle = 45), legend.title = element_text(size=13, face="b
        legend.text = element_text(size=12))
```



Interesting patterns to note emerging from this graph:

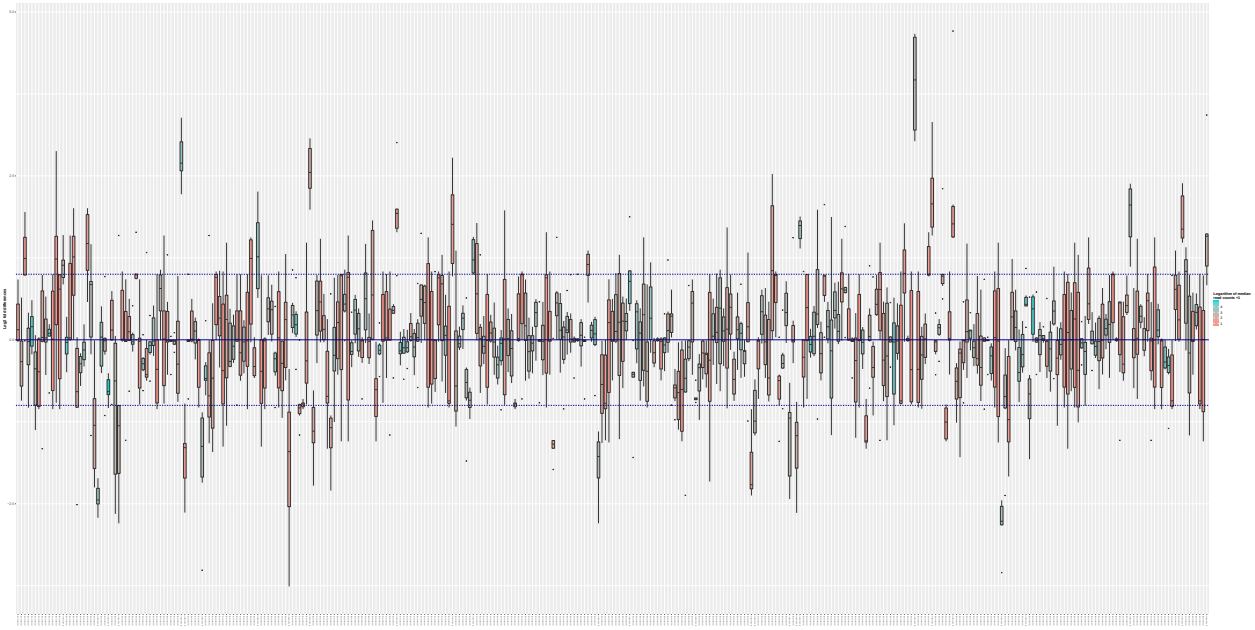
- (LOC105202834)https://www.ncbi.nlm.nih.gov/gene/?term=LOC105202834 and (LOC105202818)https://www.ncbi.nlm.nih.gov/gene/?term=LOC105202818 differences between SB and Sb are exactly the same in all SA samples. Both genes are annotated as "cytochrome P450 4C1" Presumably the reads here mapped to the same place. Is it duplicated in NA only (if not an artifact)?
- (LOC105193134)https://www.ncbi.nlm.nih.gov/gene/?term=LOC105193134 was picked up as significant with the Fisher combination method for p values, but the LFCs estimates looked all over the place. In this graph they are clearly above 0 in every comparison and almost always above 1 LFC (2 fold difference between SB and Sb).

```
p1 <- ggplot(data = parsed_counts_all_ratios_common_genes, aes(x = gene, y = lfc)) + geom_boxplot(aes(fi
                                                                                                    ou
              scale_fill_gradient(name = "Logarithm of median\nread counts +1", low = hcl(15,100,75), hi
              facet_grid(rows = vars(body_part), scales = "free", labeller = labeller(body_part = facet_
              geom_hline(yintercept = 0, color = "darkblue") + geom_hline(yintercept = -1, color = "dark
              geom_hline(yintercept = 1, color = "darkblue", linetype = "dashed") + labs(y = "Log2 fold
              #geom_rect(aes(xmin = sig_genes_both[1], xmax = sig_genes_both[1], ymin = -8, ymax = 8),
              #          fill = "transparent", color = "red", size = 1.5) +
              theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 10), axis.title.x = eleme
                    axis.text.y = element_text(vjust = 0.5, size = 12), axis.title.y = element_text(size
                    strip.text.y= element_text(size = 12, angle = 45), legend.title = element_text(size=
                    legend.text = element_text(size=12))

ggsave(filename = "results/boxplot_lfcs_common.pdf", plot = p1, device = "pdf", height = 30, width = 60
```

Same plot with North American populations only

```
#Calculate median read counts per group (gene and body part)
agg_medians_north_america <- aggregate(total_counts ~ gene + body_part, data = parsed_counts_north_ameri
agg_medians_north_america <- agg_medians_north_america[order(agg_medians_north_america$gene, agg_medians
colnames(agg_medians_north_america) <- c("gene", "body_part", "medians")

#Add medians to the main dataset
parsed_counts_north_america_ratio <- merge(agg_medians_north_america, parsed_counts_north_america_ratio)
```

```
#Get only the minimum start position in the scaffolds for all the transcripts per gene
gene_positions_min_north_america <- aggregate(x = gene_positions$start, by = list(gene_positions$contig
names(gene_positions_min_north_america) <- c("contig", "gene", "start_position")

#Select genes only in the analysis
gene_positions_min_north_america <- gene_positions_min_north_america[gene_positions_min_north_america$g

#Sort by start position per contig
gene_positions_min_north_america <- gene_positions_min_north_america[order(gene_positions_min_north_ame

#Mark the genes with significant DE in the different populations and in both
sig_genes_na_all <- rownames(results_north_america[which(results_north_america$padj < 0.05), ])
gene_positions_min_north_america$gene[gene_positions_min_north_america$gene %in% sig_genes_na_all] <- pa
parsed_counts_north_america_ratio$gene <- as.character(parsed_counts_north_america_ratio$gene)
parsed_counts_north_america_ratio$gene[parsed_counts_north_america_ratio$gene %in% sig_genes_na_all] <-

#Sort the levels in the main dataframe
parsed_counts_north_america_ratio$gene <- factor(parsed_counts_north_america_ratio$gene,
                                                 levels = gene_positions_min_north_america$gene)

p2 <- ggplot(data = parsed_counts_north_america_ratio, aes(x = gene, y = lfc)) + geom_boxplot(aes(fill =
                                                                                                outlier.si
              scale_fill_gradient(name = "Logarithm of median\nread counts +1", low = hcl(15,100,75), hi
              geom_hline(yintercept = 0, color = "darkblue") + geom_hline(yintercept = -1, color = "dark
              geom_hline(yintercept = 1, color = "darkblue", linetype = "dashed") + labs(y = "Log2 fold
              #geom_rect(aes(xmin = sig_genes_both[1], xmax = sig_genes_both[1], ymin = -8, ymax = 8),
              #          fill = "transparent", color = "red", size = 1.5) +
              theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 5), axis.title.x = elemen
                    axis.text.y = element_text(vjust = 0.5, size = 12), axis.title.y = element_text(size
                    strip.text.y= element_text(size = 12, angle = 45), legend.title = element_text(size=
                    legend.text = element_text(size=12))

p2
```



```r
ggsave(filename = "results/boxplot_lfcs_north_america.pdf", plot = p2, device = "pdf", height = 30, wid
```