**Analysis of Bacteriome-Specific Differential Expression by Sex in *Glossina morsitans***

Copeland HD, Lynch BG, Pollio AR, Spencer NJ

***1. Response to reviewers***

The reviewers expressed a desire for the original proposal to emphasize the specific significance of sex-based differences in host interactions with the symbiont *Wigglesworthia glossinida* to the issue of tsetse flies as vectors of sleeping sickness. While the *Wigglesworthia* functions that can affect Trypanosome transmission tend to be more related to their nutrient-provisioning capabilities (*1*), the transmission of *Wigglesworthia* (normally intracellular bacteria) to the developing larva (which matures *in utero* in tsetse (*2*)) is still poorly understood, although there is evidence that *Wigglesworthia* may retain and use functional flagella for motility (*3*).

Because the migration of *Wigglesworthia* to the larva is a process specific to females, we sought to identify whether host facilitation of this movement could be identified at the teneral (newly emerged adult) stage, providing the groundwork for future work in reproductive flies. Understanding this process helps clarify how *Wigglesworthia* interacts with its host and some of the reasons besides its nutritional roles that it is so critical to tsetse reproduction (*4*). Characterizing this relationship will be important for potential new vector control strategies that manipulate the resident microbes of the vector to reduce pathogen transmission, so it will be important to understand how the flies themselves respond to changes in their microbiota.

The reviewers also asked for clarification on the prevalence and impact of sleeping sickness. Sleeping sickness or human African trypanosomiasis (HAT) is said to have a fatality rate close to 100%, and, while only 6,228 gambiense HAT cases and 86 rhosidense HAT cases were reported in 2013, the total number of cases is estimated to be in the tens of thousands (*5*).

The reviewers also pointed out a lack of clarity as to the specific sample sizes of the male and female flies used in this study. Ultimately, RNA-seq data from four female and two male bacteriomes were used with two technical replicates for each sample (indicated as sequencing lanes 7 and 8). These small sample sizes, especially for the male flies, represent a limitation of the study, as we discuss later. They also asked for clarification on concerns about the other organisms present in the sample. It is true that the genome of *Wigglesworthia* is roughly 0.2% the size of the tsetse fly genome and thus would not have a large impact on the total number of unique transcripts. Additionally, the highly divergent nature of these two organisms relative to, say, two different bacteria makes this an area of relatively minor concern.

To briefly clarify the conditions for running Ballgown, Ballgown was run in RStudio using R version 3.6 and the R packages Vctrs, Ballgown, RSkittleBrewer, Genefilter, Dplyr, and Devtools.

Finally, the distribution of tasks chosen, which raised concerns about skill development with each tool, was chosen to maintain consistency across samples and ensure accountability. In practice, frequent cross-talk and troubleshooting within the group allowed team members to interact with the data at each step.

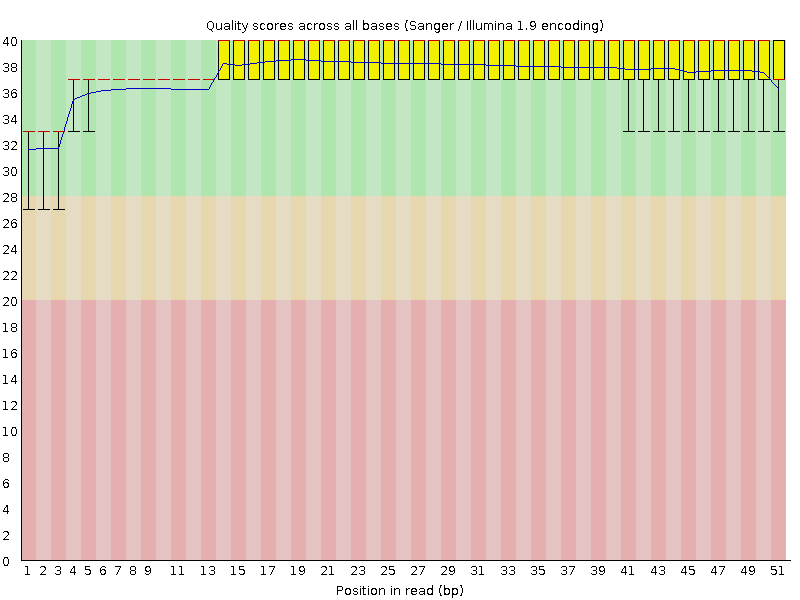
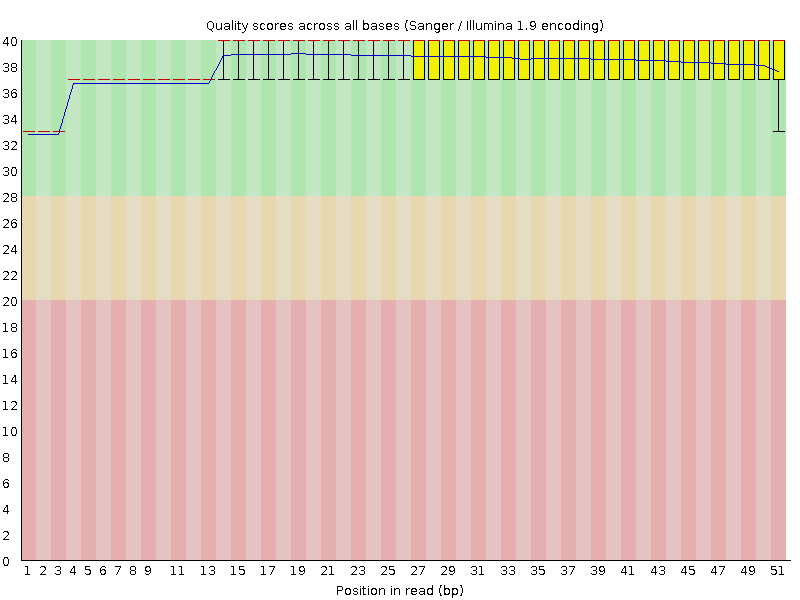
***2. Results***

*2.1 Data Retrieval and Quality Control*

In our original proposal, we described how BaseMount would be used to access the files stored on Illumina BaseSpace from the command line. However, because the FastQ files were accessible locally on one of the computers used, it proved easier to copy the archived files directly to the Spruce Knob cluster where they could be easily moved between working directories and accessed without additional software.

The genome sequence and genome features for *Glossina morsitans* were acquired from VectorBase (Genome assembly GmorY1) as FASTA and GTF files, respectively.

Quality control analysis with FastQC revealed that per-base sequence quality was consistently high as expected with no drop in quality at read ends. While the lower bounds of the base quality distribution at the beginning of the reverse reads reached below 30, this was considered not to be grounds for trimming, especially considering the short (50bp) length of these reads. Figure 1 shows representative examples of the per-base quality score plots for forward and reverse reads. Read files contained between eight and ten million reads. No adapter content was indicated for any of the samples, and no hits were obtained for any of the overrepresented sequences, indicating that they simply represented reads from highly-transcribed sequences.

**Figure 1. FastQC Per-Base Sequence Quality Plots.** Representative plots of per-base sequence quality (left=forward read, right=reverse read) output by FastQC.

*2.2 Mapping with HISAT2*

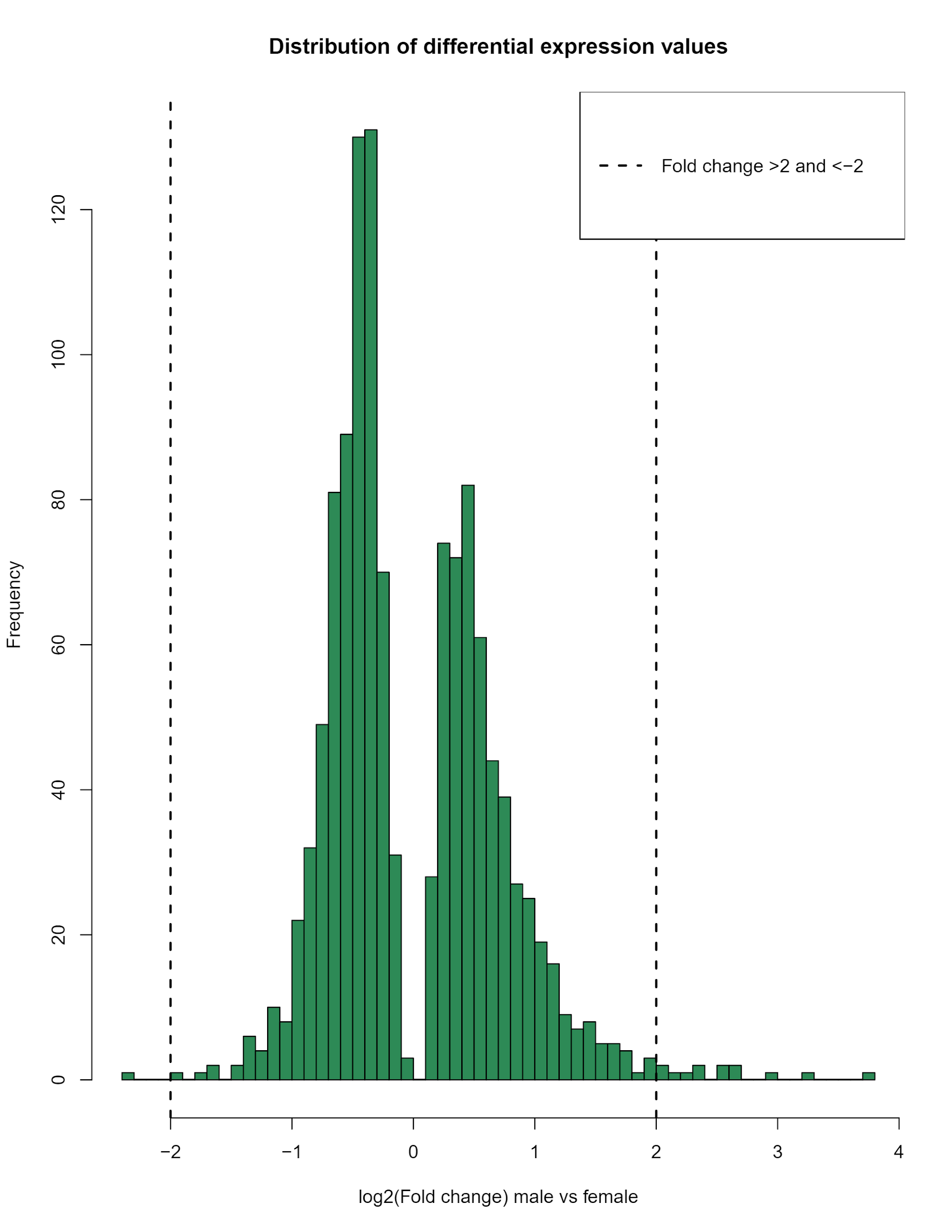
Initial runs of HISAT2 caused some difficulties, with the first mappings running indefinitely regardless of memory allocation. It was later determined that the index for this run was built using archived data, which was remedied by repeating the indexing step and mapping steps, taking care to use the unarchived reference genome, GTF, and data files. HiSat2-index and HiSat2 (mapping step) were run with default parameters. The overall alignment of each resulted in about 42.88% for the lane 7 replicates and 42.84% for the lane 8 replicates with no individual samples showing mapping rates below 40%. This is not altogether unexpected, given the high concentration of *Wigglesworthia* in these tissues. For each sample, between 1 and 3% of reads mapped more than once. While not necessarily ideal, this was not considered to be a consequence of easily remedied quality issues, and the SAM alignments were used for transcript assembly and quantification with StringTie.

*2.3 Transcript Assembly and Quantification with StringTie*

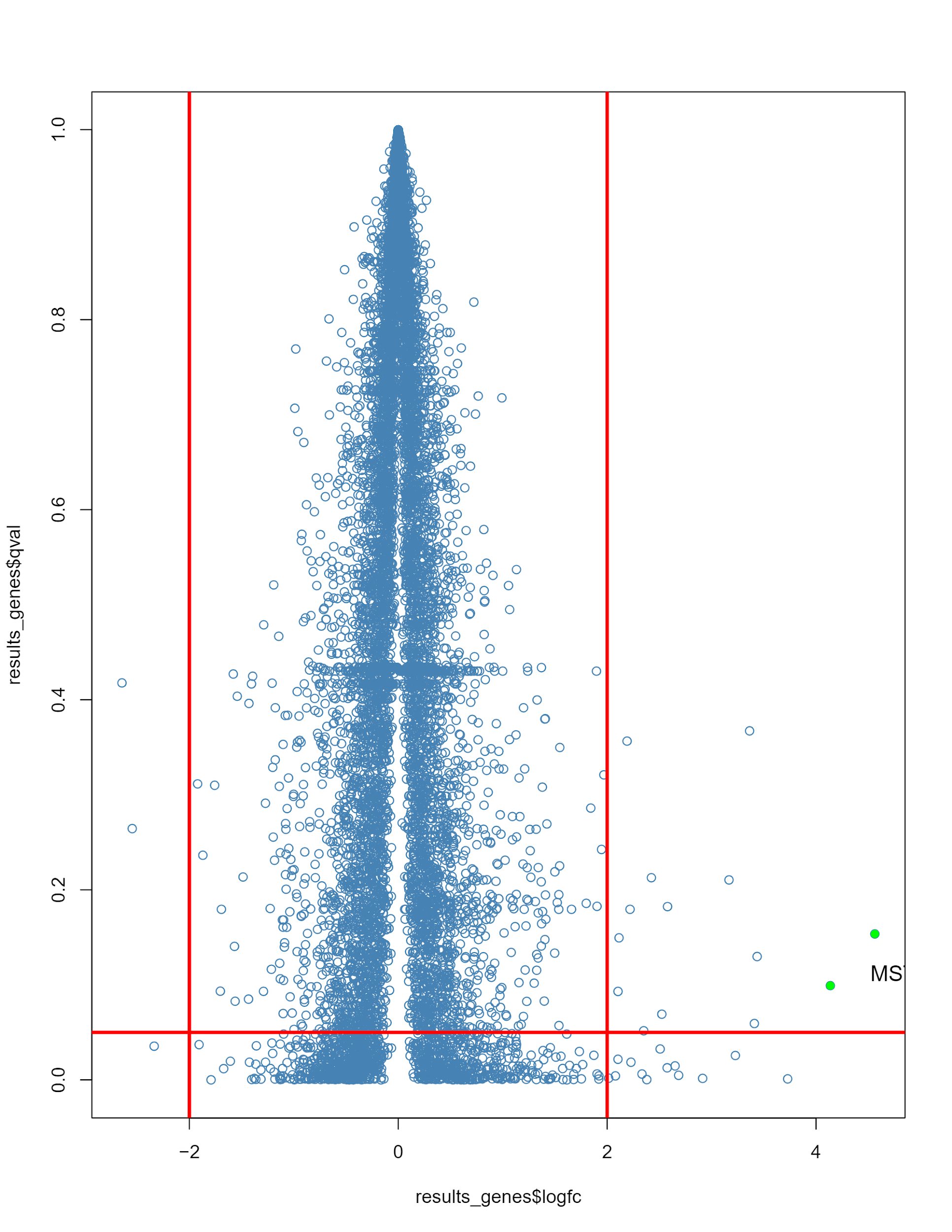
For *StringTie* there were several different files that were created by *HISTAT2* the script was written for each individual female/lane and male/lane to be run.The alignments were converted from SAM files to sorted BAM files using SamTools.The transcripts were assembled for individual samples and then merged with StringTie using default parameters, which took approximately 3 hours using 8 threads. Next, input files for Ballgown were created by running StringTie again using the -B option to output .ctab files as well as the -e option to skip reads with no reference transcripts, increasing speed. On this step, the merged transcripts previously output were used as the reference GTF file instead of the genome features.

*2.4 Expression Analysis with Ballgown*

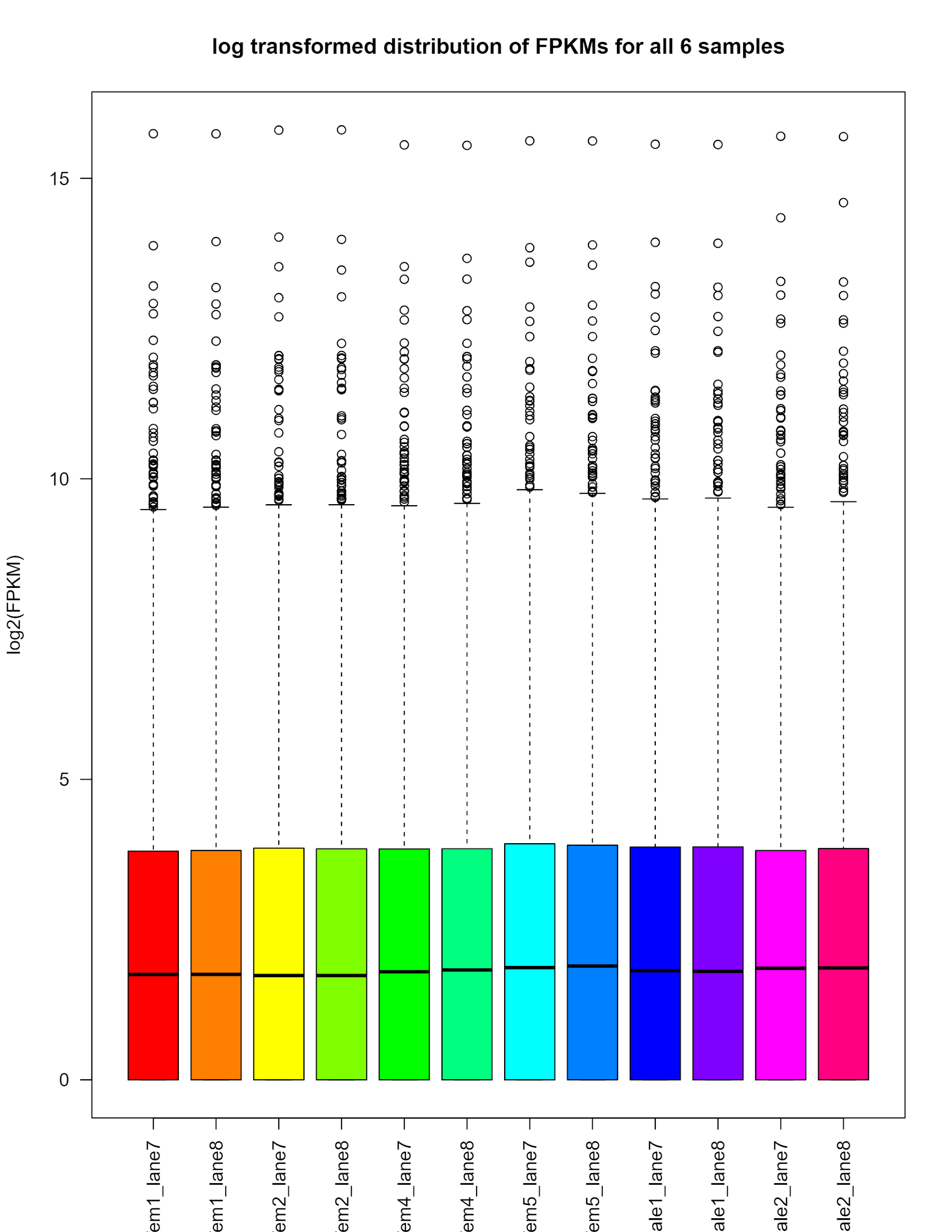
StringTie builds several files that can be read, and thus compiled to build a uniform database that can be used to extract trends. Here we used the Ballgown suite to specifically extract gene names and transcript levels to understand differential expression between our two groups (male and female). We then were able to take that data and summarize it overall a few ways. First we evaluated overall distribution of genes associated with the sex’s of interest (figure 2). We then created a volcano plot to re-emphasize the distribution of our reads (figure 3) Finally, to verify these summaries we compared overall distribution of samples using log converted data on an individual sample level (figure 4). This analysis shows that overall our samples had very few reads that were significantly different. All analyses that follow were done on only the genes that showed a significant difference.



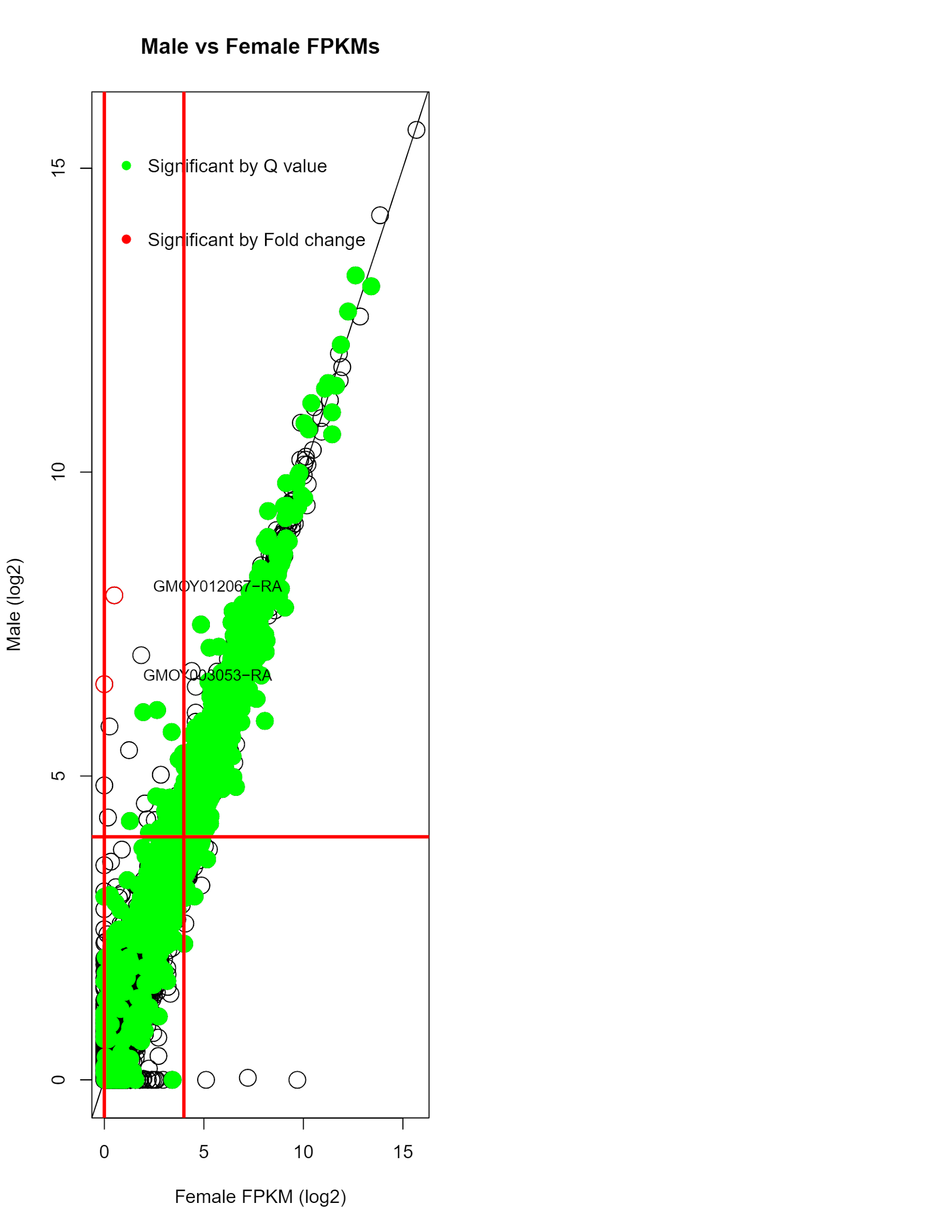
**Figure 2. Distribution Plot.** Shows the number of genes that fall within a specific fold change difference. All data has been transformed using a log2 to better visualize the overall trends. Dotted lines are meant to highlight transcripts that are expressed above or below a fold change of 2 or -2.



**Figure 3. Volcano plot of transcript data.** All mapped transcripts graphed by qvalue compared to the log transformed fold change values. All reads with a fold change higher than 4 and a qvalue higher than 0.05 are shown. Overall this shows that the majority of our reads are not significantly different. It also further emphasizes the relative similarity between all of our samples.

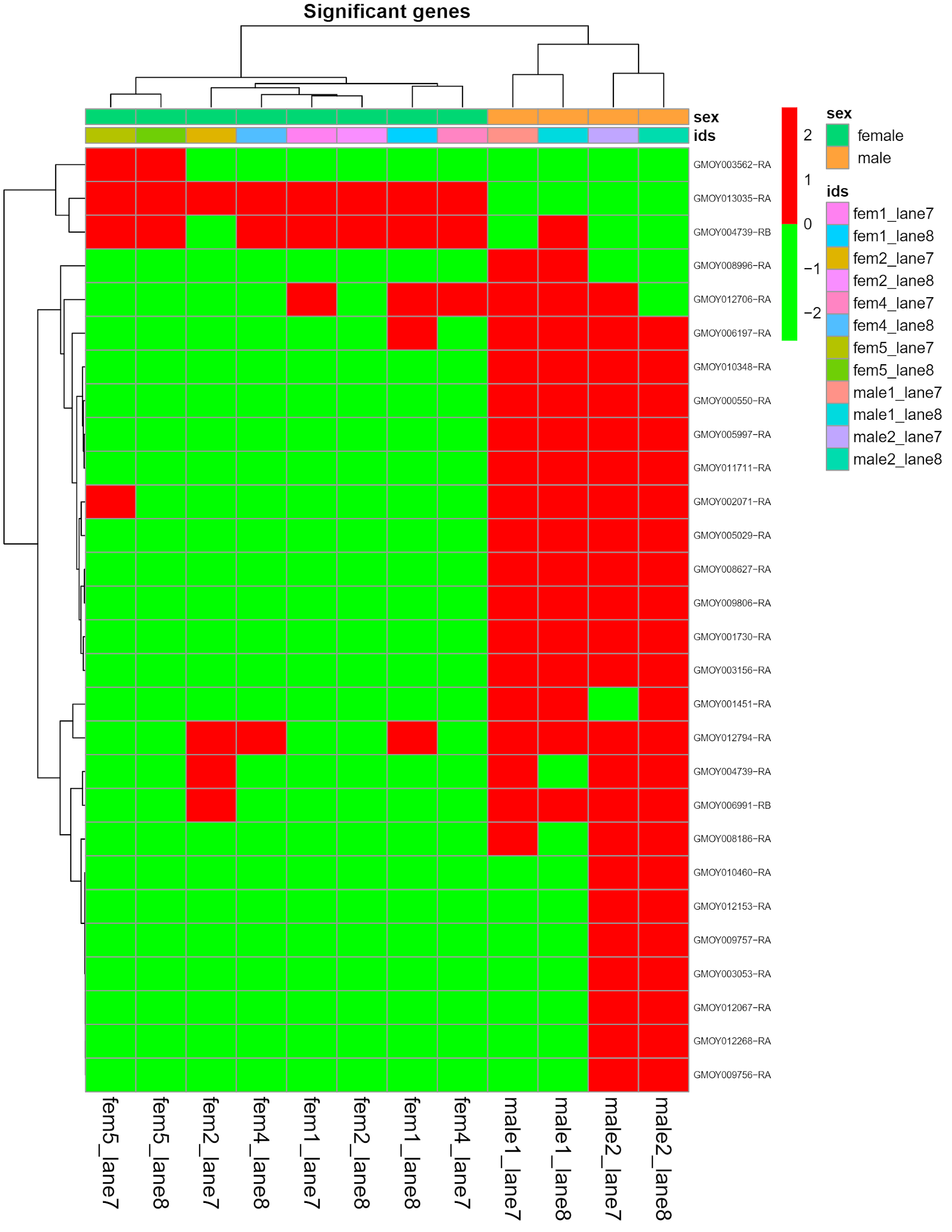


**Figure 4. Distribution plot.** Log2 transformed FPKM values collected based on transcript data compared by sample. Overall our data set was consistent. The overall profiles of all of the samples were consistent.



Our overall message from the above analysis suggested that there were few overall differences between sample sets. We decided to further explore specific genes that were found to be significantly expressed differently and then use that information to see if the overall samples cluster together using a PCA analysis. We used the Pheatmap package to build a heat map off of specific genes we had identified as significant based on a few metrics (SIGDE.txt). These significant genes were clustered using a specific argument in pheatmap. The resulting clustered samples were then graphed as a PCA plot based on significance metrics.

To determine potential functional characterizations of differentially expressed genes, several genes that showed consistent higher expression in both males relative to females (GMOY010348, GMOY000550, GMOY005997, GMOY011711, GMOY002071, GMOY005029, GMOY008627, GMOY009806, GMOY001730, and GMOY003156) were searched on the VectorBase BioMart by their gene IDs to retrieve annotations. All of these transcripts represented protein-coding genes with several being involved in the membrane or in membrane transport. GMOY000550 has several known functions in sulfate transfer, and GMOY005029 was listed as a serine protease. However, gaps in the annotation of the genome made specific functional analysis of such a small subset of genes difficult, as several genes had not been assigned any names or putative functions.



**Figure 5. Heatmap of genes that had significantly different expression values between samples.** This map was made by sorting the transcripts significance data by overall differential expression. We then captured those genes back to the original samples. Overall this shows that the 4 male runs had a few genes that were expressed differently than the female runs.



**Figure 6. PCA plot built from the most differentially expressed plot.** This emphasizes the difference between the two sets of samples when compared to there most different features. Further it shows that there was consistency associated with the technical runs, as the lane 7 and 8 replicates consistently cluster together.

***3. Discussion***

The tsetse fly is an obligate blood feeders with females reproducing viviparously. We predicted that there would be considerable transcriptional differences due to this extreme difference in physiology. We found that not to be the case. Only 29 genes had a significant difference transcriptional level by fpkm, qvalue, and pvalue. Suggesting that it is reasonable to consider individuals within a population the same male and female as teneral flies.

While some differences were identified, this small number of transcripts made specific functional analyses difficult, especially given the gaps in the annotation of the reference genome. Further pathway analysis of these genes could still be possible, but a more robust dataset will be created by comparing males and females across *Glossina* species and different tissues (bacteriome vs gut, where the gut is also colonized by a secondary symbiont). Naturally, analysis of pregnant flies is a reasonable next step in reaching our goal of characterizing tsetse interactions with the primary endosymbiont during symbiont transmission, and a larger sample size (especially for males) may make further transcriptomic differences identifiable by the statistical methods used.

Besides these issues, which are more or less intrinsic to the chosen data, several obstacles came up during the pipeline itself. Identifying an issue with HISAT2 that did not cause the program to terminate consumed considerable time and went against the general idea that if the program continued to run, something must be working. Ultimately, this was remedied by revisiting a previous step. In the future, more readily taking a whole-pipeline examination of anything that appears unusual and ensuring that runtimes are meeting the expectations for datasets of similar sizes could prevent this issue.

Additionally, this issue corroborates comments made by the reviewers about distribution of labor. A compromise between the two opposing team structures presented could be to have multiple individuals run preliminary versions of a given step in parallel, which likely would have resulted in two or more very different looking runs in this case and thus identified the problem immediately. Subsequently, final runs for each step could be conducted as one job to retain consistency.

Despite the issues this posed, it allowed for us to remedy a proposed issue with the group structure by providing troubleshooting support between team members across the different steps, allowing for overall growth in competence with the specific programs, the use of the Spruce cluster, and general bioinformatics skills.

***4. Pipeline Code***

***FastQC***

$fastqc Rio-Gmbacfem1\_S97\_L007\_R1\_001.fastq.gz Rio-Gmbacfem1\_S97\_L007\_R2\_001.fastq.gz Rio-Gmbacfem1\_S97\_L008\_R1\_001.fastq.gz Rio-Gmbacfem1\_S97\_L008\_R2\_001.fastq.gz Rio-Gmbacfem2\_S98\_L007\_R1\_001.fastq.gz Rio-Gmbacfem2\_S98\_L007\_R2\_001.fastq.gz Rio-Gmbacfem2\_S98\_L008\_R1\_001.fastq.gz Rio-Gmbacfem2\_S98\_L008\_R2\_001.fastq.gz Rio-Gmbacfem4\_S99\_L007\_R1\_001.fastq.gz Rio-Gmbacfem4\_S99\_L007\_R2\_001.fastq.gz Rio-Gmbacfem4\_S99\_L008\_R1\_001.fastq.gz Rio-Gmbacfem4\_S99\_L008\_R2\_001.fastq.gz Rio-Gmbacfem5\_S100\_L007\_R1\_001.fastq.gz Rio-Gmbacfem5\_S100\_L007\_R2\_001.fastq.gz Rio-Gmbacfem5\_S100\_L008\_R1\_001.fastq.gz Rio-Gmbacfem5\_S100\_L008\_R2\_001.fastq.gz Rio-Gmbacmale1\_S105\_L007\_R1\_001.fastq.gz Rio-Gmbacmale1\_S105\_L007\_R2\_001.fastq.gz Rio-Gmbacmale1\_S105\_L008\_R1\_001.fastq.gz Rio-Gmbacmale1\_S105\_L008\_R2\_001.fastq.gz Rio-Gmbacmale2\_S106\_L007\_R1\_001.fastq.gz Rio-Gmbacmale2\_S106\_L007\_R2\_001.fastq.gz Rio-Gmbacmale2\_S106\_L008\_R1\_001.fastq.gz Rio-Gmbacmale2\_S106\_L008\_R2\_001.fastq.gz

***HISAT2***

$gunzip Rio\*.gz

$gunzip Glossina\*.gz

$hisat2\_extract\_splice\_sites.py Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf >splice.ss

$hisat2\_extract\_exons.py Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf >exon.exon

$hisat2-build --ss splice.ss --exon exon.exon Glossina-morsitans-Yale\_SCAFFOLDS\_GmorY1.fa index\_tran

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem1\_S97\_L007\_R1\_001.fastq -2 Rio-Gmbacfem1\_S97\_L007\_R2\_001.fastq -S fem1\_lane7.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem2\_S98\_L007\_R1\_001.fastq -2 Rio-Gmbacfem2\_S98\_L007\_R2\_001.fastq -S fem2\_lane7.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem4\_S99\_L007\_R1\_001.fastq -2 Rio-Gmbacfem4\_S99\_L007\_R2\_001.fastq -S fem4\_lane7.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem5\_S100\_L007\_R1\_001.fastq -2 Rio-Gmbacfem5\_S100\_L007\_R2\_001.fastq -S fem5\_lane7.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacmale1\_S105\_L007\_R1\_001.fastq -2 Rio-Gmbacmale1\_S105\_L007\_R2\_001.fastq -S male1\_lane7.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacmale2\_S106\_L007\_R1\_001.fastq -2 Rio-Gmbacmale2\_S106\_L007\_R2\_001.fastq -S male2\_lane7.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem1\_S97\_L008\_R1\_001.fastq -2 Rio-Gmbacfem1\_S97\_L008\_R2\_001.fastq -S fem1\_lane8.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem2\_S98\_L008\_R1\_001.fastq -2 Rio-Gmbacfem2\_S98\_L008\_R2\_001.fastq -S fem2\_lane8.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem4\_S99\_L008\_R1\_001.fastq -2 Rio-Gmbacfem4\_S99\_L008\_R2\_001.fastq -S fem4\_lane8.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacfem5\_S100\_L008\_R1\_001.fastq -2 Rio-Gmbacfem5\_S100\_L008\_R2\_001.fastq -S fem5\_lane8.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacmale1\_S105\_L008\_R1\_001.fastq -2 Rio-Gmbacmale1\_S105\_L008\_R2\_001.fastq -S male1\_lane8.sam

$hisat2 -p 12 -x index\_tran -1 Rio-Gmbacmale2\_S106\_L008\_R1\_001.fastq -2 Rio-Gmbacmale2\_S106\_L008\_R2\_001.fastq -S male2\_lane8.sam

***StringTie***

**Conversion of .bam to .sam**

$samtools view -o fem1\_lane7.bam -bS fem1\_lane7.sam

$samtools sort fem1\_lane7.bam fem1\_lane7.sorted

$samtools view -o fem1\_lane8.bam -bS fem1\_lane8.sam

$samtools sort fem1\_lane8.bam fem1\_lane8.sorted

$samtools view -o fem2\_lane7.bam -bS fem2\_lane7.sam

$samtools sort fem2\_lane7.bam fem2\_lane7.sorted

$samtools view -o fem2\_lane8.bam -bS fem2\_lane8.sam

$samtools sort fem2\_lane8.bam fem2\_lane8.sorted

$samtools view -o fem4\_lane7.bam -bS fem4\_lane7.sam

$samtools sort fem4\_lane7.bam fem4\_lane7.sorted

$samtools view -o fem4\_lane8.bam -bS fem4\_lane8.sam

$samtools sort fem4\_lane8.bam fem4\_lane8.sorted

$samtools view -o fem5\_lane7.bam -bS fem5\_lane7.sam

$samtools sort fem5\_lane7.bam fem5\_lane7.sorted

$samtools view -o fem5\_lane8.bam -bS fem5\_lane8.sam

$samtools sort fem5\_lane8.bam fem5\_lane8.sorted

$samtools view -o male1\_lane7.bam -bS male1\_lane7.sam

$samtools sort male1\_lane7.bam male1\_lane7.sorted

$samtools view -o male1\_lane8.bam -bS male1\_lane8.sam

$samtools sort male1\_lane8.bam male1\_lane8.sorted

$samtools view -o male2\_lane7.bam -bS male2\_lane7.sam

$samtools sort male2\_lane7.bam male2\_lane7.sorted

$samtools view -o male2\_lane8.bam -bS male2\_lane8.sam

$samtools sort male2\_lane8.bam male2\_lane8.sorted

**Assemble and quantify expressed genes and transcripts**

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem1\_lane7.sorted.gtf fem1\_lane7.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem1\_lane8.sorted.gtf fem1\_lane8.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem2\_lane7.sorted.gtf fem2\_lane7.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem2\_lane8.sorted.gtf fem2\_lane8.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem4\_lane7.sorted.gtf fem4\_lane7.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem4\_lane8.sorted.gtf fem4\_lane8.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem5\_lane7.sorted.gtf fem5\_lane7.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem5\_lane8.sorted.gtf fem5\_lane8.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem6\_lane7.sorted.gtf fem6\_lane7.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o fem6\_lane8.sorted.gtf fem6\_lane8.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o male1\_lane7.sorted.gtf male1\_lane7.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o male1\_lane8.sorted.gtf male1\_lane8.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o male2\_lane7.sorted.gtf male2\_lane7.sorted.bam

$stringtie -p 8 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmorY1.9.gtf -o male2\_lane8.sorted.gtf male2\_lane8.sorted.bam

**Merge transcripts from all samples**

$stringtie --merge fem1\_lane7.sorted.gtf fem1\_lane8.sorted.gtf fem2\_lane7.sorted.gtf fem2\_lane8.sorted.gtf fem4\_lane7.sorted.gtf fem4\_lane8.sorted.gtf fem5\_lane7.sorted.gtf fem5\_lane8.sorted.gtf fem6\_lane7.sorted.gtf fem6\_lane8.sorted.gtf male1\_lane7.sorted.gtf male1\_lane8.sorted.gtf male2\_lane7.sorted.gtf male2\_lane8.sorted.gtf -p 12 -G Glossina-morsitans-Yale\_BASEFEATURES\_GmoreY1.9.gtf -o merged.gtf

**Estimate transcript abundances and create table counts for Ballgown**

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem1\_lane7/fem1\_lane7.sorted.gtf fem1\_lane7.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem1\_lane8/fem1\_lane8.sorted.gtf fem1\_lane8.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem2\_lane7/fem2\_lane7.sorted.gtf fem2\_lane7.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem2\_lane8/fem2\_lane8.sorted.gtf fem2\_lane8.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem4\_lane7/fem4\_lane7.sorted.gtf fem4\_lane7.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem4\_lane8/fem4\_lane8.sorted.gtf fem4\_lane8.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem5\_lane7/fem5\_lane7.sorted.gtf fem5\_lane7.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem5\_lane8/fem5\_lane8.sorted.gtf fem5\_lane8.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem6\_lane7/fem6\_lane7.sorted.gtf fem6\_lane7.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/fem6\_lane8/fem6\_lane8.sorted.gtf fem6\_lane8.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/male1\_lane7/male1\_lane7.sorted.gtf male1\_lane7.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/male1\_lane8/male1\_lane8.sorted.gtf male1\_lane8.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/male2\_lane7/male2\_lane7.sorted.gtf male2\_lane7.sorted.bam

$stringtie -e -B -p 12 -G merged.gtf -o ballgown/male2\_lane8/male2\_lane8.sorted.gtf male2\_lane8.sorted.bam

**BALLGOWN SCRIPT**

setwd("C:/Users/arpol/Not\_Bad/R\_packages/")

library(vctrs)

library(ballgown)

library(RSkittleBrewer)

library(genefilter)

library(dplyr)

library(devtools)

bg\_chrX=ballgown(dataDir="ballgown", samplePattern = "ane", meas='all')

sampleNames(bg\_chrX)

#read in the folder names with the read data

pheno\_data = read.csv("class\_work/run\_data.csv")

pData(bg\_chrX)=pheno\_data

tx\_table = texpr(bg\_chrX, "all")

fpkm = texpr(bg\_chrX,meas="FPKM")

coverage = texpr(bg\_chrX,meas = "cov")

**#extract transcript with both coverage and fpkm**

Whole\_bg\_chrx=texpr(bg\_chrX,'all')

gene\_expression=as.data.frame(Whole\_bg\_chrx)

gene\_expression=subset(gene\_expression, select = -c(1:5))

gene\_expression=subset(gene\_expression, select = -c(6))

gene\_expression=subset(gene\_expression, select = -c(7))

gene\_expression=subset(gene\_expression, select = -c(8))

gene\_expression=subset(gene\_expression, select = -c(9))

gene\_expression=subset(gene\_expression, select = -c(10))

gene\_expression=subset(gene\_expression, select = -c(11))

gene\_expression=subset(gene\_expression, select = -c(12))

gene\_expression=subset(gene\_expression, select = -c(13))

gene\_expression=subset(gene\_expression, select = -c(14))

gene\_expression=subset(gene\_expression, select = -c(15))

gene\_expression=subset(gene\_expression, select = -c(16))

gene\_expression=subset(gene\_expression, select = -c(17))

trans\_expression = row.names(gene\_expression)<gene\_expression$t\_name

row.names(gene\_expression) <- gene\_expression$t\_name

gene\_expression=subset(gene\_expression, select =-c(1))

trans\_expression = subset(gene\_expression,select=-c(1:4))

**#extract expression for genes in fpkm and coverage**

gene\_bg\_chrx=gexpr(bg\_chrX)

**#Look at the exon, intron and transcript data**

structure(bg\_chrX)$exon

**#differentially expressed transcripts**

det = stattest(bg\_chrX, feature="transcript", covariate="sex", getFC=TRUE, meas="FPKM")

#store mapping between transcripts and genes

transcript\_gene\_table=indexes(bg\_chrX)$t2g

**#check the unique number of genes in the transcript/genes table**

length(unique(transcript\_gene\_table[,"g\_id"]))

#plot average transcript length

hist(Whole\_bg\_chrx$length, breaks=50, xlab="Transcript length (bp)", main="Distribution of transcript lengths", col = "darkslategray1")

#how many transcripts are there per gene? count the number of genes and count the number of transcripts pere gene and plot it.

counts=table(transcript\_gene\_table[,"g\_id"])

#makes chart that summarizes the data

c\_one = length(which(counts == 1))

c\_more\_than\_one = length(which(counts > 1))

c\_max = max(counts)

hist(counts, breaks=50, col="bisque4", xlab="Transcripts per gene", main="Distribution of transcript count per gene")

legend\_text = c(paste("Genes with one transcript =", c\_one), paste("Genes with more than one transcript =", c\_more\_than\_one), paste("Max transcripts for single gene = ", c\_max))

legend("topright", legend\_text)

**#extract gene names and transcript names**

gene\_names=data.frame(SYMBOL=unique(rownames(trans\_expression)))

**# differential transcript expression**

results\_txns = stattest(bg\_chrX, feature='transcript', getFC = T, covariate='sex',meas='FPKM' )

**#add names and IDs**

t.ids=Whole\_bg\_chrx[,c(1,6)]

t\_names=unique(Whole\_bg\_chrx[,c(1,6)])

results\_txns\_merged = merge(results\_txns,t.ids,by.x=c("id"),by.y=c("t\_id"))

head(results\_txns\_merged)

**# Calculate differentially expressed genes and use FPKM in calculating** differential gene expression

results\_genes = stattest(bg\_chrX, feature="transcript", covariate="sex", getFC=TRUE, meas="FPKM")

**# Compare the data before and after normalization. boxplot with and** without log transformation

par(mfrow=c(1,2))

boxplot(trans\_expression, col=rainbow(6), las=2, ylab="log2(FPKM)", main="Distribution of FPKMs for all 6 samples")

boxplot(log2(trans\_expression+1), col= rainbow(12), las=2, ylab="log2(FPKM)", main="log transformed distribution of FPKMs for all 6 samples")

**# Log # fold changes and store it in logfc** columnresults\_genes[,"logfc"] = log2(results\_genes[,"fc"])

results\_txns\_merged[,"logfc"] = log2(results\_txns\_merged[,"fc"])

**# Identify the genes (rows) with adjusted p-value (i.eq-value) < 0.05**

qsig=which(results\_txns\_merged$qval<0.05)

**# draw histogram**

hist(results\_txns\_merged[qsig,"logfc"], breaks=50, col="seagreen", xlab="log2(Fold change) male vs female", main="Distribution of differential expression values")

abline(v=c(-2,2), col="black", lwd=2, lty=2)

legend("topright", "Fold change >2 and <-2", lwd=2, lty=2)

**# Convert the matrix to data**

write.table(trans\_expression, "gene\_expression.txt", sep="\t")

trans\_expression$female=rowMeans(trans\_expression[, c(1:8)])

trans\_expression$male=rowMeans(trans\_expression[,c(9:12)])

**#to avoid 0**

x=log2(trans\_expression[,"female"]+1)

y=log2(trans\_expression[,"male"]+1)

plot(x=x, y=y, pch=1, cex=2, xlab="Female FPKM (log2)", ylab="Male (log2)", main="Male vs Female FPKMs")

abline(a=0, b=1)

xqsig=x[qsig]

yqsig=y[qsig]

points(x=xqsig, y=yqsig, col="green", pch=19, cex=2)

fsig=which(abs(results\_txns\_merged$logfc)>4)

xfsig=x[fsig]

yfsig=y[fsig]

points(x=xfsig, y=yfsig, col="red", pch=1, cex=2)

legend\_text = c("Significant by Q value", "Significant by Fold change")

legend("topright", legend\_text,bty="n",pch = c(19,19), col=c("green","red"))

**# label the significant genes**

library(calibrate)

textxy(xfsig,yfsig, cex=0.8, labs=row.names(trans\_expression[fsig,]))

# add red line through 0

abline(v=0, col="red", lwd=3)

**# add red line through fold change 4 (log2,2)**

abline(v=c(4,-4), col="red", lwd=3)

abline(h=c(-4,4), col="red",lwd=3)

**#volcano plot**

**# Filter genes by log fold change by 16 fold**

fc\_sig\_results\_genes=which(abs(results\_txns\_merged$logfc)>4)

fc\_sig\_results\_genes\_plot=results\_genes[fc\_sig\_results\_genes,]

plot(results\_genes$logfc,results\_genes$qval, col="steelblue", pch=1)

**#abline**

abline(v=c(2,-2), col="red", lwd=3)

abline(h=0.05, col="red",lwd=3)

**# highlight the genes with color**

points(fc\_sig\_results\_genes\_plot$logfc,fc\_sig\_results\_genes\_plot$qval, col="green", pch=16)

**# label the significant genes**

textxy(fc\_sig\_results\_genes\_plot$logfc,fc\_sig\_results\_genes\_plot$qval, labs=fc\_sig\_results\_genes\_plot$id, cex=1.2)

colors = colorRampPalette(c("white", "blue","red","green","yellow"))

par(mfrow=c(1,2))

plot(x,y)

smoothScatter(x,y, colramp = colors)

**# Identify the genes (rows) below p-value 0.05**

sigpi = which(results\_txns\_merged[,"pval"]<0.05)

**# Extract p-significant genes in a separate object**

sigp = results\_txns\_merged[sigpi,]

sigde = which(abs(sigp[,"logfc"]) >= 2)

**# Extract and store the statistically significant genes (rows) that are upregulated/ downregulated by 4 fold**

sig\_tn\_de = sigp[sigde,]

**# Order by q value, followed by differential expression**

sorted\_sig\_tn\_de = order(sig\_tn\_de[,"qval"], -abs(sig\_tn\_de[,"logfc"]), decreasing=FALSE)

sig\_tn\_de[sorted\_sig\_tn\_de,c("t\_name","fc","pval","qval","logfc")]

write.table(output, file="ballgown/SigDE.txt", sep="\t", row.names=FALSE, quote=FALSE)

sig\_gene\_expression=trans\_expression[rownames(trans\_expression) %in% sig\_tn\_de$t\_name,]

**#remove female and male columns**

sig\_gene\_expression=sig\_gene\_expression[,-c(13:14)]

**# for pheatmap function, column names and row names of data and pdata mush be identical# change the row names**

rownames(pheno\_data)=pheno\_data[,1]

**# remove the id column**

phenotype\_table=subset(pheno\_data, select = -c(ids))

**# change the colnames to match with the sample names**

colnames(sig\_gene\_expression)=row.names(pheno\_data)

library(pheatmap)

pheatmap(as.matrix(sig\_gene\_expression), scale = "row", clustering\_distance\_rows = "correlation", clustering\_method = "complete",annotation\_col = pheno\_data , main="Significant genes",fontsize\_col=14, fontsize\_row = 6 ,color = c("green","red"))

**#PcA plot**

pca\_data=prcomp(t(sig\_gene\_expression))

**# Calculate PCA component percentages**

pca\_data\_perc=round(100\*pca\_data$sdev^2/sum(pca\_data$sdev^2),1)

df\_pca\_data = data.frame(PC1 = pca\_data$x[,1], PC2 = pca\_data$x[,2], sample = colnames(sig\_gene\_expression), condition = rep(c("Female","Male")))

library(ggplot2)

library(ggrepel)

ggplot(df\_pca\_data, aes(PC1,PC2, color = sample))+

geom\_point(size=8)+

labs(x=paste0("PC1 (",pca\_data\_perc[1],")"), y=paste0("PC2 (",pca\_data\_perc[2],")"))

ggplot(df\_pca\_data, aes(PC1,PC2, color = condition))+

geom\_point(size=8)+

labs(x=paste0("PC1 (",pca\_data\_perc[1],")"), y=paste0("PC2 (",pca\_data\_perc[2],")"))+

geom\_text\_repel(aes(label=sample),point.padding = 0.75)

***References***

1. R. V. M. Rio, A. K. S. Jozwick, A. F. Savage, A. Sabet, A. Vigneron, Y. Wu, S. Aksoy, B. L. Weiss, Mutualist-Provisioned Resources Impact Vector Competency. *mBio*. **10**, e00018-19 (2019).

2. J. B. Benoit, G. M. Attardo, A. A. Baumann, V. Michalkova, S. Aksoy, Adenotrophic Viviparity in Tsetse Flies: Potential for Population Control and as an Insect Model for Lactation. *Annu Rev Entomol*. **60**, 351–371 (2015).

3. R. V. M. Rio, R. E. Symula, J. Wang, C. Lohs, Y. Wu, A. K. Snyder, R. D. Bjornson, K. Oshima, B. S. Biehl, N. T. Perna, M. Hattori, S. Aksoy, Insight into the Transmission Biology and Species-Specific Functional Capabilities of Tsetse (Diptera: Glossinidae) Obligate Symbiont Wigglesworthia. *mBio*. **3**, e00240-11 (2012).

4. R. Pais, C. Lohs, Y. Wu, J. Wang, S. Aksoy, The Obligate Mutualist Wigglesworthia glossinidia Influences Reproduction, Digestion, and Immunity Processes of Its Host, the Tsetse Fly. *Appl. Environ. Microbiol.* **74**, 5965–5974 (2008).

5. J. R. Franco, P. P. Simarro, A. Diarra, J. G. Jannin, Epidemiology of human African trypanosomiasis. *Clin Epidemiol*. **6**, 257–275 (2014).