Differential Expression of Serum microRNAs in Brain Metastasis of Breast Cancer

Nathaniel Lesperance

1. Introduction

MicroRNAs (miRNAs) are short non-coding RNA that regulate gene expression where the canonical mechanism of action is repression of translation by binding to the 3' untranslated region of target mRNAs and thus miRNA expression is typically associated with a down-regulation of their target genes (Saliminejad et al., 2019). However, there is recent evidence there are non-standard mechanisms that could increase expression of their target mRNAs or otherwise alter their expression in unique ways (Saliminejad et al., 2019). It has been estimated that more than half of genes are regulated by miRNAs in humans and they have been shown to play critical roles in carcinogenesis (Jonas & Izaurralde, 2015). They also are found in blood serum so unique miRNA sera profiles can be useful for minimally invasive diagnostics and targets for therapy (Saliminejad et al., 2019). Profiles of serum miRNAs are also informative for prediction of breast cancer outcomes and interestingly, in a unique way compared to intracellular miRNA profiles from tumour samples (Zhu et al., 2014). Given their short length, their ambiguity about impact across many biological pathways and because novel miRNAs are still being detected regularly, many traditional software tools are not optimized for analysis of miRNAs (Da Sacco et al., 2012). Determining analytical choices that are appropriate for effective differential expression (DE) analysis of miRNA can therefore have drastic implications in clinical decisions and open avenues for successful diagnostics.

There is ambiguity about analytical tools, such as alignment tools and DE analysis tools commonly used for RNA-seq analysis regarding their applicability to analysis of miRNA data. Many tools provide suggestions about analytical settings and model choices that are more optimal for RNA-seq data compared to DNA-seq data with intention for use with longer and coding RNA reads, often without mention of adaptations needed in the analysis of shorter, non-coding RNA (Da Sacco et al., 2012). One such challenge in DE analysis is the use of negative binomial distributions and fitting general linear models with Naïve Bayes Estimation of dispersion for those distributions. This is followed in DE analysis pipelines by an exact test to determine group differences in the means of these tagwise (aka gene-wise, but that term is not appropriate here for miRNA counts) distributions. Adjusting for covariates can improve fitting of general linear models and lesson or remove the impact of random and batch effects in the data and this method is used widely in randomized controls to narrow down the effect of treatment while adjusting for covariates that contribute to outcome but their application is still case-specific and up for debate (Etminan et al., 2021).

The analysis presented here is an investigation into analytical methods that have implications for the precision and accuracy of differential expression analysis of miRNAs. The primary analytical step scrutinized is the ways in which differential expression results are impacted by incorporating covariate adjustments when using Naïve Bayes estimation of dispersion for negative binomial distributions of each miRNA count and their effect on the fitting of downstream general linear models for exact testing of differences in the means of these distributions. Calculating principal components (PCs) by Principal Component Analysis (PCA) as potential covariates allows the comparison of results using models fit without covariate adjustment, models fit with batch effect adjustment (i.e sequencing lanes) and models fit with batch effects and calculated covariate (PCs) adjustment. The objective of this project is to evaluate the accuracy and precision differences that come with adjusting for covariates when fitting models of differentially expressed miRNAs.

2. Description of Data

The dataset used in this project are miRNA raw read counts collected from an investigation of brain-metastasis of breast cancer, however the study is yet to be published but the data is available on the Gene Expression Omnibus (GEO). The count data is publicly available as supplementary files from GEO (Accession GSE216934). Supplementary files for this study contain counts of 2374 miRNAs that were calculated from RNA-seq reads of 42 blood sera samples, where 21 patients have breast cancer that has metastasized but not to the brain (called *control* in this project) and 21 patients have breast cancer that has metastasized to the brain (called *case* in this project). Using Bowtie with default settings, reads were mapped to human genome (hg19) and miRNAs were counted using a ‘custom script’ against miRNAs from the miRBase v21 database. There is also a supplementary file containing metadata that has been included in the project. Sequencer ‘lanes’ were included in the sample labels and provide a clear batch effect that provides an opportunity for covariate adjustment. Control samples are matched with age, time from diagnosis, and HER2/ER/PR tumour status to case samples, but this data was not provided in the metadata. All files were initially accessed on Nov 24th, 2023 but the script provided below performs a fresh download and unzip of the supplementary files relevant for this investigation using ftp links to GEO. The dataset was selected not only because of the interest I have in the role of miRNA in cancer but because the case and control conditions are so biologically similar that batch and random effects can have a greater impact on masking the detectable differential expression of miRNAs.

3. Data Acquisition, Exploration, Filtering, and Quality Control

```r
library(stringr)
library(tidyverse)
library(R.utils)
library(limma)
library(Glimma)
library(edgeR)
library(knitr)
library(ggplot2)
library(factoextra)
library(lattice)
library(gridExtra)
```

```r
# Supplementary data files from GSE216934
matrix_url <- "https://ftp.ncbi.nlm.nih.gov/geo/series/GSE216nnn/GSE216934/matrix/GSE216934_series_matri
suppl_1_url <- "https://ftp.ncbi.nlm.nih.gov/geo/series/GSE216nnn/GSE216934/suppl/GSE216934_miRNA_13071
suppl_2_url <- "https://ftp.ncbi.nlm.nih.gov/geo/series/GSE216nnn/GSE216934/suppl/GSE216934_miRNA_13082
suppl_3_url <- "https://ftp.ncbi.nlm.nih.gov/geo/series/GSE216nnn/GSE216934/suppl/GSE216934_miRNA_13101
suppl_4_url <- "https://ftp.ncbi.nlm.nih.gov/geo/series/GSE216nnn/GSE216934/suppl/GSE216934_miRNA_14031

numPCs <- 9  # Number of PCs used
sig_thresh <- 0.05  # Significance threshold for p-values used
```

```r
utils::download.file(matrix_url, destfile = "GSE216934_metamatrix.txt.gz")
metamatrix_file <- R.utils::gunzip("GSE216934_metamatrix.txt.gz", overwrite = T)

start_line <- 30
end_line <- 43
selected_lines <- readLines(metamatrix_file)
```

```r
# Create a data frame from the selected lines
df <- data.frame(do.call(rbind, strsplit(selected_lines[start_line:end_line], "\t")),
    stringsAsFactors = FALSE)

# Remove double quotes from all entries
meta_df <- data.frame(lapply(df, function(x) gsub("\"", "", x)))

# Only extract unique/useful rows (Sample names, GEO Accession, Disease State)
meta_df <- meta_df[c(1, 2, 13), -c(1:5)]
colnames(meta_df) <- meta_df[1, ]  # Make column names the sample names
meta_df <- meta_df[-1, ]  # Remove sample names
meta_df[2, ] <- gsub("disease state: ", "", meta_df[2, ])  # Process disease string
lanes <- sub(".*_L00([1-6]).*", "\\1", colnames(meta_df))  # Extract lane
meta_df <- rbind(meta_df, lanes)
rownames(meta_df) <- c("GEO_Accession", "Group", "Lane")
# Order metadata df so matches ordered miRNA count df
meta_df <- meta_df[, order(colnames(meta_df))]
```

```r
# Supplement files 1-4 are the miRNA counts from serum samples of 42 patients
urls <- c(suppl_1_url, suppl_2_url, suppl_3_url, suppl_4_url)

# Initialize raw counts df to build from files
raw_counts <- data.frame(X = character(0))

# Loop through each supplementary file, unzip and add sample miRNA counts to df
for (link in urls) {
    filename <- str_extract(link, "[A-z0-9_]*.txt.gz")
    utils::download.file(link, destfile = filename)
    gzipped_file <- R.utils::gunzip(filename, overwrite = T)
    counts <- read.delim(gsub("\\.gz", "", filename))
    raw_counts <- merge(counts, raw_counts, by = "X", all = TRUE)
}

# Take row names from miRNA name and remove that column
rownames(raw_counts) <- raw_counts[, 1]
miRNA_counts <- raw_counts[, -1]

# Since outer join, will get NAs and need to replace Replaced with 0s here
miRNA_counts <- replace(miRNA_counts, is.na(miRNA_counts), 0)

# Remove duplicates
mi_filt <- miRNA_counts[, -c(10, 13, 14, 17, 19, 22, 23, 26, 27, 30, 31, 34)]
```

```r
# Order miRNA count df so matches ordered metadata df
mi_filt <- mi_filt[, order(colnames(mi_filt))]

# Assign case and control to different groups
groups <- unlist(unname(meta_df[2, ]))
names(groups) <- colnames(mi_filt)

# Calculate counts per million and log2
cpm_log = cpm(mi_filt, log = TRUE)
```

```r
# Investigating expression by histogram to determine minimal expression cutoff
# Counts per million in log2 values
median_log2_cpm <- apply(cpm_log, 1, median)
mean_log2_cpm <- apply(cpm_log, 1, mean)
hist(median_log2_cpm)
```

```r
hist(mean_log2_cpm)
```

```r
expr_cutoff <- 0

# Low Expression Filtering by histogram inspection - low expression cutoff of 0
counts_clean_mean <- mi_filt[mean_log2_cpm > expr_cutoff, ]
counts_clean_med <- mi_filt[median_log2_cpm > expr_cutoff, ]
# Investigate filtering results
dim(counts_clean_mean)
dim(counts_clean_med)
summary(counts_clean_mean)
summary(counts_clean_med)

# Used median log cpm above 0 to define minimal expression
counts_clean <- mi_filt[median_log2_cpm > expr_cutoff, ]
nrow(counts_clean)  # Investigate

# Create DGElist object from counts of miRNAs
diff_ex <- DGEList(counts = counts_clean, group = groups)

# TMM Normalization
diff_ex <- calcNormFactors(diff_ex, method = "TMM")
diff_ex$samples
# Looking at Norm factors, outliers are 148665, 226907 and 304618, All of which
# are samples run on lane 6

# Counts per million of miRNAs with at least minimal expression
cpm_log_normed = cpm(diff_ex, log = TRUE, normalized.lib.sizes = TRUE)  # with normalization
```

```r
# Investigating raw counts
str(mi_filt)
summary(mi_filt)
# 5704, 5785, and 5786 have 3rd quartile counts of 0

# Histogram of 0 counts for each sample, 3 outliers (5704, 5785, 5786)
hist(colSums(mi_filt == 0))
```

```r
colnames(mi_filt[colSums(mi_filt == 0) > 1800])

# PCA of log and normed count data
PCA <- prcomp(t(cpm_log_normed), scale = TRUE, center = TRUE)
PCs <- PCA$x

# Assign each PC to its own variable
for (i in seq(1:numPCs)) {
    assign(paste0("PC", i), PCA$x[, i])
}

# Investigate PCA results with scree plot of variance explained
fviz_eig(PCA, geom = "bar", main = "Variance Explained by PC", xlab = "PC")
```

```
# PC1 - 27% variance explained and PC2 - 13%

# Investigate outliers by density plots of PCs
plot_pc1 <- densityplot(PC1, pch = 19, col = "blue", main = "PC1 Distribution")
plot_pc2 <- densityplot(PC2, pch = 19, col = "blue", main = "PC2 Distribution")
# Lane 6 samples identified again by PCA as outliers Other PCs showed no clear
# outliers so are omitted here
```

```
# RERUN above with new dataset
mi_filt_new <- mi_filt[, -c(3, 11, 17)]
groups_new <- unlist(unname(meta_df[2, ]))[-c(3, 11, 17)]
names(groups_new) <- colnames(mi_filt[, -c(3, 11, 17)])

# Calculate counts per million and log2
cpm_log_new <- cpm(mi_filt_new, log = TRUE)
# Used median log2 cpm above 0 to define minimal expression
median_log2_cpm_new <- apply(cpm_log_new, 1, median)
counts_clean_new <- mi_filt_new[median_log2_cpm_new > expr_cutoff, ]
nrow(counts_clean_new)  # Investigate results of filtering (756 miRNAs retained)

# Create DGElist object from raw counts
diff_ex <- DGEList(counts = counts_clean_new, group = groups_new, remove.zeros = FALSE)

# TMM Normalization
diff_ex <- calcNormFactors(diff_ex, method = "TMM")
diff_ex$samples  # No clear outliers in norm factors

# Counts per million of miRNAs with at least minimal expression
cpm_log <- cpm(counts_clean_new, log = TRUE, normalized.lib.sizes = FALSE)  # without normalization
cpm_log_normed_new <- cpm(diff_ex, log = TRUE, normalized.lib.sizes = TRUE)  # with normalization

# Redo PC to see if outliers removed
PCA <- prcomp(t(cpm_log_normed_new), scale = TRUE, center = TRUE)
PCs <- PCA$x

# Assign each PC to its own variable
for (i in seq(1:9)) {
    assign(paste0("PC", i), PCA$x[, i])
}

# Investigate PCA results with scree plot of variance explained
fviz_eig(PCA, geom = "bar", main = "Variance Explained by PC", xlab = "PC")
```

```
# PC1 - 28% variance explained and PC2 - 12.5% (similar to before filtering)

# Investigate outliers by density plots of PCs
plot_pc1new <- densityplot(PC1, pch = 19, col = "blue", main = "PC1 Distribution After Filtering")
plot_pc2new <- densityplot(PC2, pch = 19, col = "blue", main = "PC2 Distribution After Filtering")
```

```
# Look for +- 2.5 SD away from mean
a <- subset(PC1, PC1 > (mean(PC1) + 2.5 * sd(PC1)))
b <- subset(PC1, PC1 < (mean(PC1) - 2.5 * sd(PC1)))
outliers <- c(a, b)
outliers
# shows 5785 and 5786 still may be outliers but not the same library size
# issues so will proceed with these 39

table(groups)  # Started with 21 case and 21 control
table(groups_new)  # Filtered has 19 case and 20 control
```

```
# Plot PC outlier investigation in one grid
grid.arrange(plot_pc1, plot_pc2, plot_pc1new, plot_pc2new, ncol = 2, nrow = 2)
```

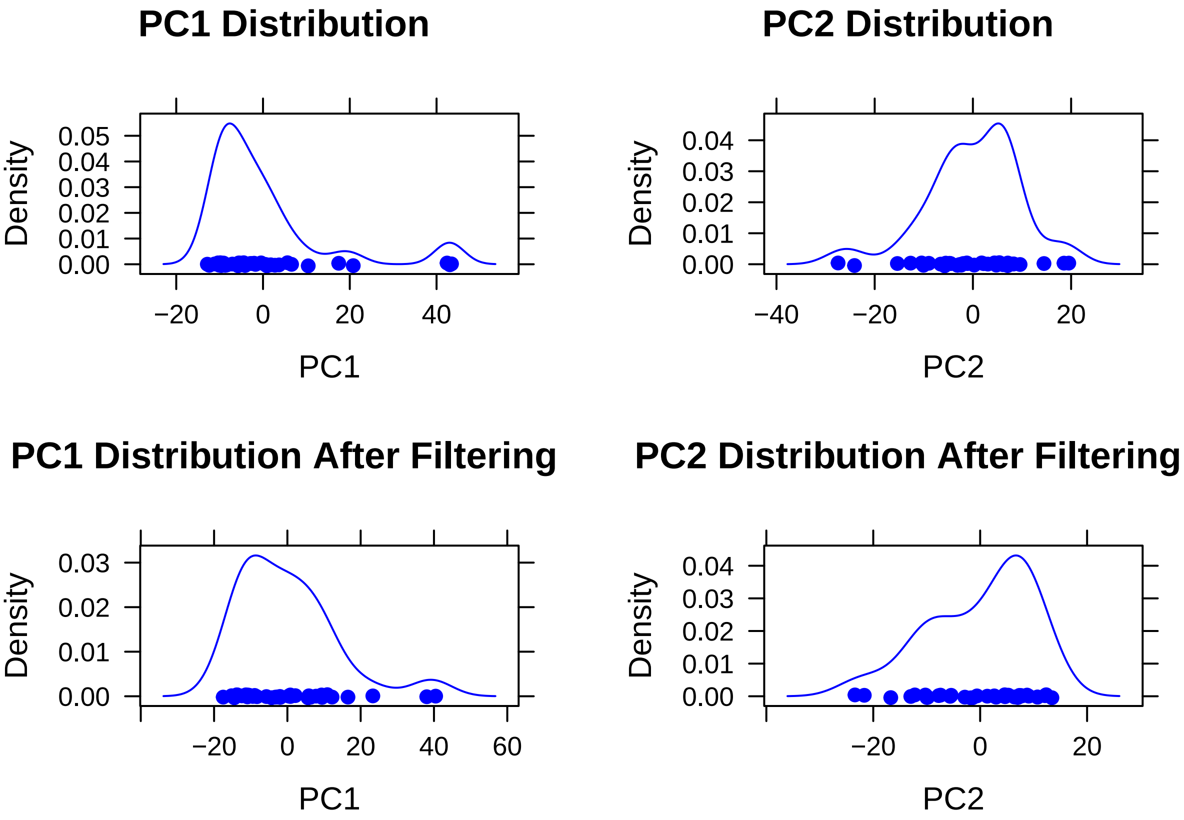

Figure 1: Distributions of PC1 and PC2 from PCA of 749 miRNA counts for all 42 samples before filtering, and 756 miRNA counts for 39 samples after filtering outliers. Dots represent PC values of specific samples and the line above represents the distribution of that principal component.

```
# Prepare dfs for graphing
raw_gathered <- gather(as.data.frame(cpm_log), key = "Sample", value = "cpm_counts")

normed_gathered <- gather(as.data.frame(cpm_log_normed_new), key = "Sample", value = "cpm_counts")
# Plot count distribution before normalization
b4_plot <- ggplot(raw_gathered, aes(x = cpm_counts, fill = Sample)) + geom_density(alpha = 0.15) +
    scale_x_continuous(limits = c(-2, 20)) + labs(x = "Counts per Million (log2)",
    y = "Density", title = "Before TMM Normalization") + theme_bw() + theme(legend.position = "none",
    plot.title = element_text(hjust = 0.5))

# Plot count distribution before normalization
af_plot <- ggplot(normed_gathered, aes(x = cpm_counts, fill = Sample)) + geom_density(alpha = 0.15) +
    scale_x_continuous(limits = c(-2, 20)) + labs(x = "Counts per Million (log2)",
    y = "Density", title = "After TMM Normalization") + theme_bw() + theme(legend.position = "none",
    plot.title = element_text(hjust = 0.5))

# Plot normalization investigation in one grid
grid.arrange(b4_plot, af_plot, nrow = 2)
```

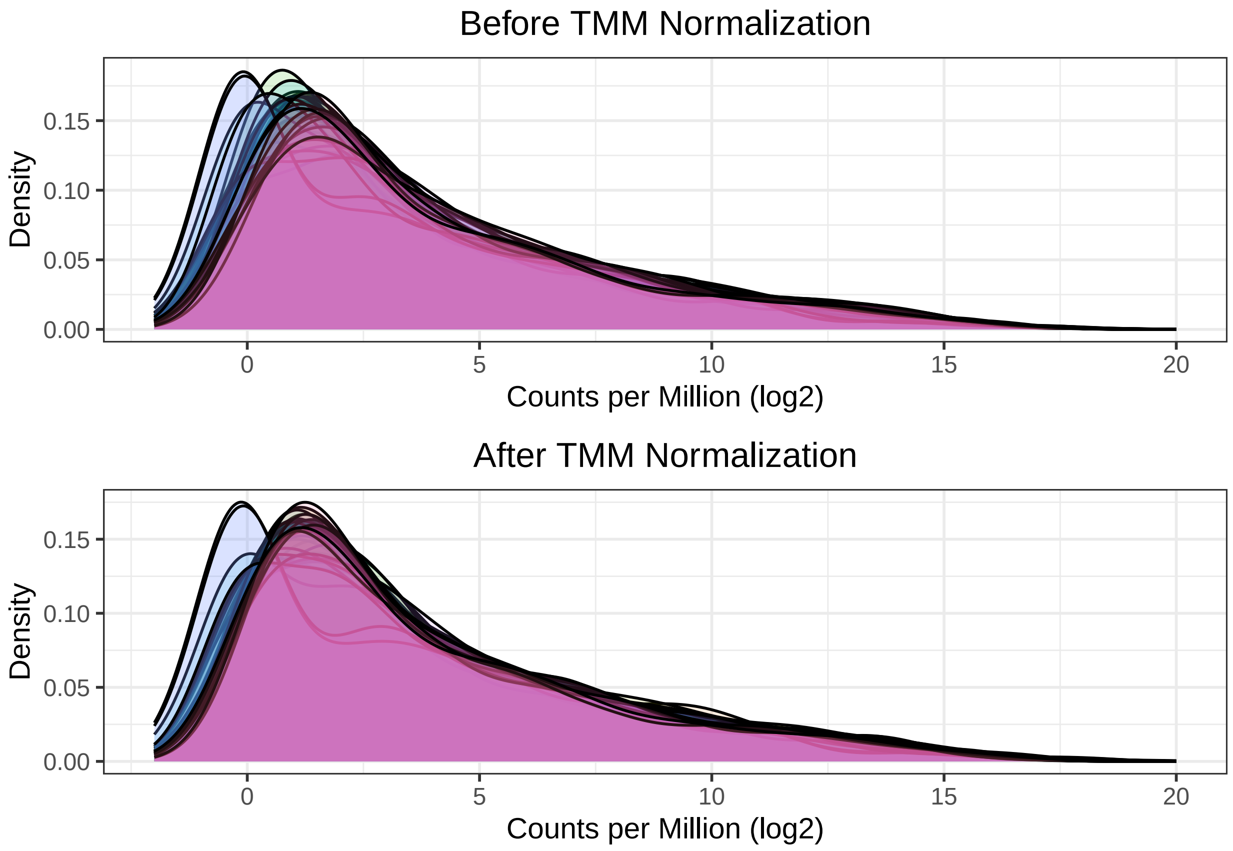

Figure 2: Log-transformed miRNA count distributions before and after TMM normalization where each sample is coloured differently.

4. Main Software Tool

The primary software tool I will use to tackle the objective of this project is the *edgeR* package (v4.0.2) (Robinson et al., 2010) to perform differential expression analysis and the *factoextra* package (v1.0.7) (Kassambara & Mundt, 2020) to perform PCA. Having explored both packages minimally in previous projects and seen *edgeR* mentioned in the literature multiple times, it is clear both are popular packages for their respective analyses. *DESeq* and *edgeR* are also highlighted in the literature as packages for DE analysis of non-coding RNA (He et al., 2018). The user guide vignette for *edgeR* also has a short section about designing and using additive models (models with adjustments for additional covariates). The extensive vignette about RNA-seq analysis using *edgeR* presents many different applications of the tools with concise explanations of analytical choices, however, the addition of PCs as covariates nor the differential expression of miRNA data are not explored in the vignette, leaving a gap to be investigated here. *DESeq2* was an alternative to *edgeR* that I considered and there are some alternatives to *factoextra* that I considered, primarily *mixOmics*, which can be used for other clustering and dimensionality reduction methods such as sparse partial least squares regression but the extensive vignette, my familiarity with *factoextra* and *edgeR* as well as their common use in bioinformatic pipelines (and lecture notes in the case of *edgeR*) made them appealing choices for this project.

5. Differential Expression Analysis

```r
# Create lane covariate
lanes <- unlist(unname(meta_df[3, ]))[-c(3, 11, 17)]
names(lanes) <- colnames(mi_filt[, -c(3, 11, 17)])
# Create covariate factors
group <- as.factor(groups_new)
lane <- as.factor(lanes)

# Build different models with various covariates
model1 = model.matrix(~0 + group)
model2 = model.matrix(~0 + group + lane)
model3 = model.matrix(~0 + group + lane + PC1)
model4 = model.matrix(~0 + group + lane + PC1 + PC2)
model5 = model.matrix(~0 + group + lane + PC1 + PC2 + PC3)
model6 = model.matrix(~0 + group + lane + PC1 + PC2 + PC3 + PC4 + PC5 + PC6)

design <- matrix(list(model1, model2, model3, model4, model5, model6))

diff_expre_test <- function(dge_list, design_matrix) {
    # Function to perform differential expression analysis of a DGE_list object
    # using the provided design matrix First, estimate tagwise dispersion for
    # exact test that follows
    disp <- estimateDisp(dge_list, design = design_matrix)
    # Exact test for group difference by comparing tagwise means of
    # distributions
    et <- exactTest(disp)
    # Extract df for returning, ordered by p-value
    results_edgeR <- as.data.frame(topTags(et, n = nrow(dge_list$counts), sort.by = "p.value"))
    return(results_edgeR)
}

# Initialize p-value results matrix for visualization
pvals <- matrix(ncol = 0, nrow = nrow(diff_ex$counts))
```

```r
# Using Bonferroni multiple test correction to define significant results
Ntests <- nrow(diff_ex$counts)
adj_thresh <- sig_thresh/Ntests

# Name different models
model_names <- c("None", "Lane (L)", "L+PC1", "L+PC1+PC2", "L+PC1+PC2+PC3", "L+PC1 to PC6")

# Loop through each design matrix, performing DE analysis for each and
# reporting significant results
for (i in seq(1:length(design))) {
    # Perform DE analysis
    results_edgeR <- diff_expre_test(diff_ex, design[[i]])
    # Add to pval matrix for use in downstream visualization
    pvals <- cbind(pvals, results_edgeR$PValue[order(rownames(results_edgeR))])
    # Filter to only significant results
    sig_results_edgeR <- subset(results_edgeR, results_edgeR$PValue < adj_thresh)
    # Report significance results
    message(paste(nrow(sig_results_edgeR), "significantly differentially expressed miRNAs identified usi
                model with",
        model_names[i], "covariates"))
}
```

```
## 0 significantly differentially expressed miRNAs identified using
##                  model with None covariates
```

```
## 2 significantly differentially expressed miRNAs identified using
##                  model with Lane (L) covariates
```

```
## 8 significantly differentially expressed miRNAs identified using
##                  model with L+PC1 covariates
```

```
## 17 significantly differentially expressed miRNAs identified using
##                  model with L+PC1+PC2 covariates
```

```
## 26 significantly differentially expressed miRNAs identified using
##                  model with L+PC1+PC2+PC3 covariates
```

```
## 41 significantly differentially expressed miRNAs identified using
##                  model with L+PC1 to PC6 covariates
```

```r
# Build variables for visualization
log_pvals <- -log10(t(pvals))
data_long <- as.data.frame(log_pvals)
data_long$Model <- model_names
data_long <- gather(data_long, key = "Test", value = "LogPValue", -Model)
data_long$Colour <- ifelse(data_long$LogPValue < -log10(adj_thresh), "Notsig", "Sig")  # Colour signifi

# Create plot of each model's p-values with significant threshold line
ggplot(data_long, aes(x = factor(Model, levels = model_names), y = LogPValue, color = Colour)) +
    geom_point(position = position_jitter(width = 0.15, height = 0), size = 0.8) +
    geom_hline(yintercept = -log10(adj_thresh), linetype = "dashed", color = "red") +
    labs(title = "Distribution of P-Values", x = "Model Covariates", y = "-log(P-Value)") +
    scale_color_manual(values = c("black", "#F781BF")) + theme_minimal() + theme(legend.position = "non
    axis.text.x = element_text(size = 8), plot.title = element_text(hjust = 0.5))
```

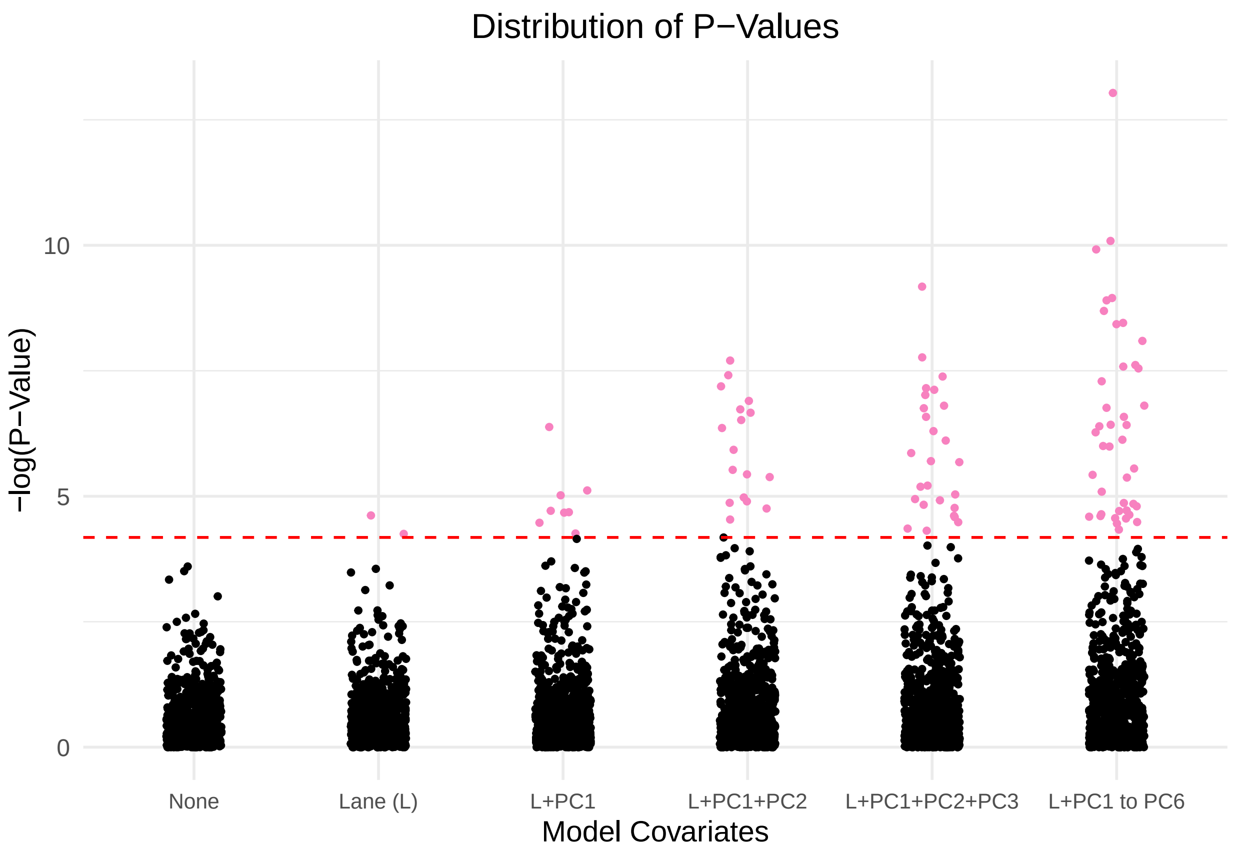

Figure 3: Each models resulting -log(p-value) after exact testing for differences in the tagwise mean between negative binomial distributions of 756 miRNAs. The significance threshold (dashed red line) is 0.05 and is Bonferroni multiple test corrected. In the respective order of models on the x-axis, there were 0, 2, 8, 18, 26 and 41 significant miRNAs with p-values passed the significance threshold (coloured pink).

```
create_qq_plot <- function(data, main) {
  # Function to create QQ-plot object from DE resulting df
  # Already sorted by p-value but remove any p = 1
  refined <- data[data$PValue < 1,]
  pvals <- refined$PValue[refined$PValue < 1] # only pvalues
  # Calculate p-values expected by chance alone
  expected_pvals <- ppoints(nrow(refined))

  qq_data <- data.frame(Expected = -log10(expected_pvals),
                        Observed = -log10(pvals))

  # Create QQ-plot
  qqplot <- ggplot(qq_data, aes(x = Expected, y = Observed)) +
    geom_point() +
    geom_abline(intercept = 0, slope = 1, color = "red") +
    labs(title = paste(main, "Covariates Model"),
         x = "Expected (-log10)", y = "Observed (-log10)") +
    theme_bw() + theme(axis.title.x = element_text(size = 9),
                       axis.title.y = element_text(size = 9),
                       plot.title = element_text(size = 12, hjust = 0.5))
  return(qqplot)
}
```

```
# Loop through each design matrix and create a QQ-plot of the DE analysis
for (j in seq(1:length(design))){
  assign(paste0("qqplot", j), # Name and assign plot objects
         create_qq_plot(diff_expre_test(diff_ex, design[[j]]), model_names[j]))
}
```

```
# Create grid of QQ-plots
grid.arrange(qqplot1, qqplot2, qqplot3, qqplot4, qqplot5, qqplot6, nrow = 3, ncol = 2)
```

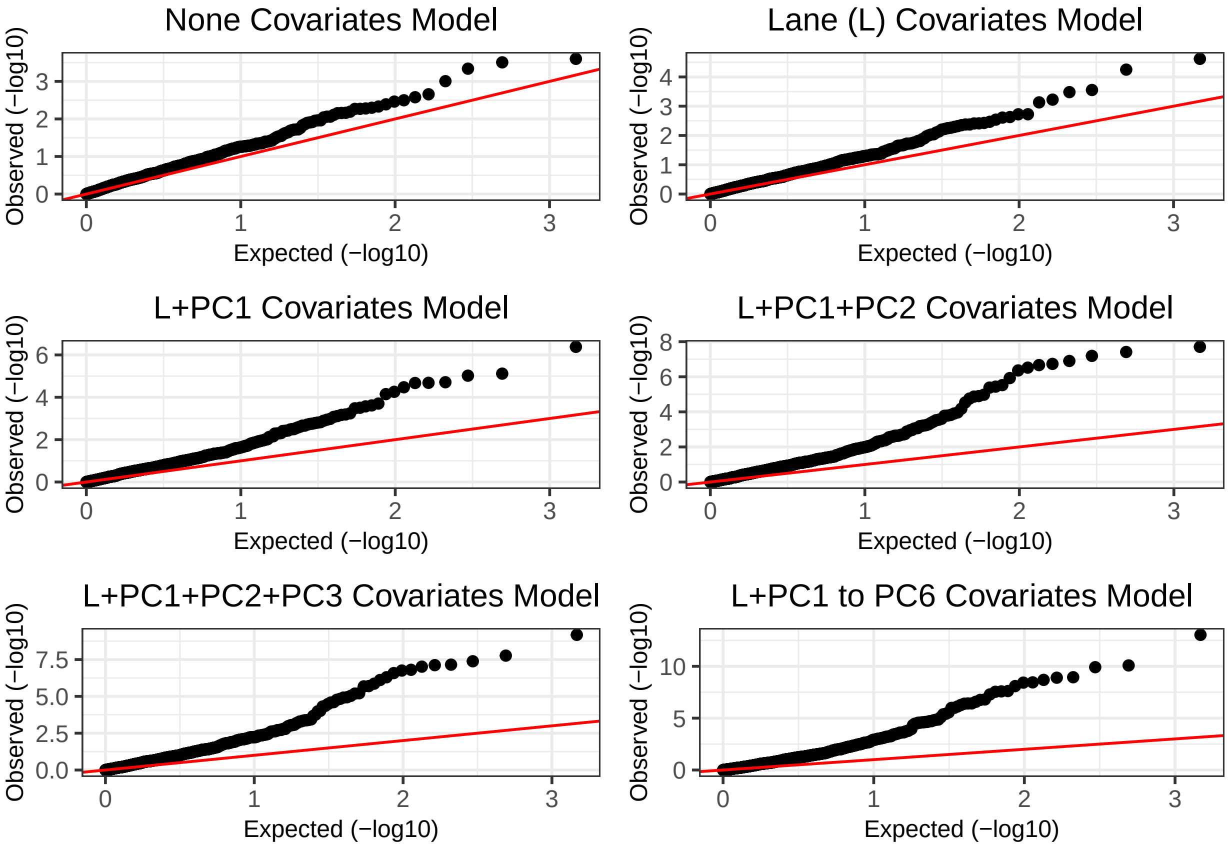

Figure 4: Quantile-Quantile plots (QQ-plots) of expected vs observed -log(p-values) for each model. The red line signifies the expected values if the p-value represented chance alone and the black points represent the p-values from each exact test for significant difference in means of negative binomial miRNA count distributions.

6. Results and Discussion

Several samples were run multiple times on different sequencer lanes, so initial filtering removed duplicate samples without preference for sequencing lane. Figure 1 and investigation of underlying PC1 and PC2 values as well as investigation of normalization factors showed that samples run on sequencing lane 6 were extreme outliers, possibly pointing to a batch effect unique to lane 6, so they were removed from downstream analysis while maintaining approximately equal ratio of case and control. Though Figure 1 shows there were still outliers in PC1 (samples numbered 5785 and 5786) after outlier filtering, their library sizes did not show the same extremes as the initial outliers and they were not outliers in PC2 after filtering so they were not discarded. Figure 2 shows TMM normalization marked an improvement in uniformity of count distributions across all samples and filtering of low expression, removing miRNAs with median count that are not greater than 1 read per million, retained 756 miRNAs for downstream analysis. The significant miRNAs detected by each statistical model (Figure 3) shows the precision progressively increased as covariates were included in the model's design. However, progressing forward into functional analysis with these significant miRNAs may be a mistake. The slope of observed -log(p-values) increasing away from the expected line as covariates are added in Figure 4 demonstrates the apparent increase in precision may be spurious and come at the cost of accuracy. If the precision was not sacrificing accuracy, the QQ-plots in Figure 4 would have shown a tight association to the expected line with a more significant increase in slope at the 'tail' for the identified significant miRNAs.

There are several caveats to be considered when interpreting the results from this project. Initially, the data used was subject to analytical methods that were not made clear and choices that were outside of my control, namely the counting of miRNA reads using a 'custom script' that was not provided by the original authors and the choice of alignment tool and settings, which has been shown to have great impact on downstream DE analysis (Ziemann et al., 2016). The replacement of NA count values with zeroes is likely also impacting downstream results and imputation could be an avenue to explore further. Additionally, using QQ-plots as a valid measure of accuracy is only upheld if the data are expected to follow the distribution (Pliel, 2016) and there is further investigation needed to determine if the assumption that miRNA count data would be normally distributed holds true. Had the metadata contained age, tumour status and time since diagnosis, there could have been more covariates to adjust for in each model and the results may have been more clear about precision and accuracy as well as more reliable.

This project could be extended in many ways, both to make these results more robust and to increase the scope of analytical choices that could impact miRNA DE analysis. One future extension could be performing gene target prediction for the top significant miRNAs followed by functional enrichment analysis to determine if the significant miRNAs identified are implicated in brain metastasis. This would help to clarify if the accuracy is truly being sacrificed with the increased significance afforded by adjusting for covariates. The statistical test used here can also be replaced with quasi-likelihood F-tests, shown to generate less type 1 errors and resulting in more conservative results (Lun et al., 2016), which is especially relevant for miRNA analysis given each miRNAs propensity to be involved in numerous pathways. Additionally, performing similar analysis on simulated count dataset could help to with this 'precision over accuracy' clarification as it would allow the ground truth to be known and the methodological choices could be evaluated more objectively. There are many other important considerations for DE analysis of miRNA count data, such as the impact of various normalization techniques and the challenge of gene target prediction (Da Sacco et al., 2012, Jonas & Izaurralde, 2015) that are worthy of exploring to increase the impact these studies can have on disease progression prediction and diagnostics.

References:

Da Sacco, L., Baldassarre, A., & Masotti, A. (2012). Bioinformatics tools and novel challenges in long non-coding RNAs (lncRNAs) functional analysis. *International journal of molecular sciences*, 13(1), 97–114. https://doi.org/10.3390/ijms13010097

Etminan, M., Brophy, J. M., Collins, G., Nazemipour, M., & Mansournia, M. A. (2021). To Adjust or Not to Adjust: The Role of Different Covariates in Cardiovascular Observational Studies. *American heart journal*, 237, 62–67. https://doi.org/10.1016/j.ahj.2021.03.008

He, Q., Liu, Y., & Sun, W. (2018). Statistical analysis of non-coding RNA data. *Cancer letters*, 417, 161–167. https://doi.org/10.1016/j.canlet.2017.12.029

Jonas, S., & Izaurralde, E. (2015). Towards a molecular understanding of microRNA-mediated gene silencing. Nature reviews. *Genetics*, 16(7), 421–433. https://doi.org/10.1038/nrg3965

Kassambara A, Mundt F (2020). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.7, https://CRAN.R-project.org/package=factoextra.

Lun, ATL, Chen, Y, and Smyth, GK (2016). It's DE-licious: a recipe for differential expression analyses of RNA-seq experiments using quasi-likelihood methods in edgeR. *Methods in Molecular Biology(Clifton, N.J.)*, 1418, 391–416. https://doi.org/10.1007/978-1-4939-3578-9_19

Pleil J. D. (2016). QQ-plots for assessing distributions of biomarker measurements and generating defensible summary statistics. *Journal of breath research*, 10(3), 035001. https://doi.org/10.1088/1752-7155/10/3/035001

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics (Oxford, England)*, 26(1), 139–140. https://doi.org/10.1093/bioinformatics/btp616

Saliminejad, K., Khorram Khorshid, H. R., Soleymani Fard, S., & Ghaffari, S. H. (2019). An overview of microRNAs: Biology, functions, therapeutics, and analysis methods. *Journal of cellular physiology*, 234(5), 5451–5465. https://doi.org/10.1002/jcp.27486

Zhu, J., Zheng, Z., Wang, J., Sun, J., Wang, P., Cheng, X., Fu, L., Zhang, L., Wang, Z., & Li, Z. (2014). Different miRNA expression profiles between human breast cancer tumors and serum. *Frontiers in genetics*, 5, 149. https://doi.org/10.3389/fgene.2014.00149

Ziemann, M., Kaspi, A., & El-Osta, A. (2016). Evaluation of microRNA alignment techniques. *RNA (New York, N.Y.)*, 22(8), 1120–1138. https://doi.org/10.1261/rna.055509.115