I. Materials and Methods

The meta-analysis was conducted using the R statistical programming language and several packages associated with Bioconductor, an open source bioinformatics software implemented through R.

Download and Install R - https://cran.r-project.org/
Good user interface for R (R-Studio) - https://www.rstudio.com/

Example workflow of a GeneLab Affymetrix microarray dataset (GLDS-17)
https://genelab-data.ndc.nasa.gov/genelab/accession/GLDS-17/

1. Download the raw data set and save it to a known location on your computer
2. Unzip and extract the raw .CEL files
3. Create a "targets" file (easiest to use Excel) and save as targets.csv to same file location as raw .CEL files
    a. This file should have the experimental conditions of each array saved within it
    b. The first column should be the .CEL file name, followed by the name you'd like to refer to that specific file as and then the condition that the microarray is testing
    c. This information can be found by browsing the links attached to the study on GeneLab or to download the ISA-TAB Metadata File and using ISA-Creator to explore the Metadata

File	Name	Condition
8287_FLA1_1_Ferl.Paul_(ATH1-121501).CEL	Spaceflight_Seedlings_1	Spaceflight_Seedlings
8288_FlightA3_2_Ferl.Paul_(ATH1-121501).CEL	Spaceflight_Seedlings_2	Spaceflight_Seedlings
8289_FLA5_3_Ferl.Paul_(ATH1-121501).CEL	Spaceflight_Seedlings_3	Spaceflight_Seedlings
8290_GCA1_4_Ferl.Paul_(ATH1-121501).CEL	Ground_Seedlings_1	Ground_Seedlings
8291_GCA3_5_Ferl.Paul_(ATH1-121501).CEL	Ground_Seedlings_2	Ground_Seedlings
8292_GCA5_6_Ferl.Paul_(ATH1-121501).CEL	Ground_Seedlings_3	Ground_Seedlings

4. Now, open R-Studio and create a new project for the analysis
5. Install the following packages needed for analysis (these packages only need to be installed once and then can be subsequently loaded by using the R command "library("Package Name")":

```
source("https://bioconductor.org/biocLite.R")
biocLite()
biocLite(c("affy","limma","oligo","affyPLM","annotate","GO.db"))
install.packages(c("dplyr","ggplot2","ggrepel"))
```

6. Load the packages into your current workspace:

```
library("affy")
library("limma")
library("oligo")
library("affyPLM")
library("annotate")
library("dplyr")
library("GO.db")
library("ggplot2")
library("ggrepel")
```

7. Set the working directory to the location of the raw .CEL files:

```
setwd("F:/NASA/R Analysis/Arabidopsis thaliana/E-MTAB-1009-Tissue/RawData")
```

a. To see your current working directory:

getwd()

8. Once the working directory is set to the location with the raw data, read in the .CEL files to an AffyBatch object:

data <- ReadAffy()

Important: Make sure that the rows in the targets file align with how the raw files were read into R:

sampleNames(data)

```
> sampleNames(data)
[1] "8287_FLA1_1_Ferl.Paul_(ATH1-121501).CEL"
[2] "8288_FlightA3_2_Ferl.Paul_(ATH1-121501).CEL"
[3] "8289_FLA5_3_Ferl.Paul_(ATH1-121501).CEL"
[4] "8290_GCA1_4_Ferl.Paul_(ATH1-121501).CEL"
[5] "8291_GCA3_5_Ferl.Paul_(ATH1-121501).CEL"
[6] "8292_GCA5_6_Ferl.Paul_(ATH1-121501).CEL"
```

File
8287_FLA1_1_Ferl.Paul_(ATH1-121501).CEL
8288_FlightA3_2_Ferl.Paul_(ATH1-121501).CEL
8289_FLA5_3_Ferl.Paul_(ATH1-121501).CEL
8290_GCA1_4_Ferl.Paul_(ATH1-121501).CEL
8291_GCA3_5_Ferl.Paul_(ATH1-121501).CEL
8292_GCA5_6_Ferl.Paul_(ATH1-121501).CEL

9. Normalize the data using the RMA normalization **OR** Mas5 procedure:

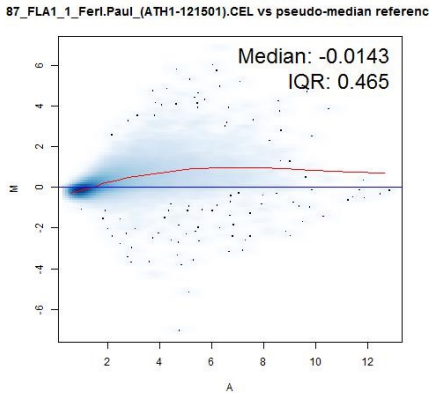
AEsetnorm <- affy::rma(data)

OR

AEsetnorm <- affy::mas5(data)

10. Create MA plots to evaluate the normalization:

```
for (i in 1:length(sampleNames(data)))
{
name = paste("MAplotnorm",i,".jpg",sep="")
jpeg(name)
oligo::MAplot(AEsetnorm,which=i)
dev.off()}
```



11. Read in the targets file in order to match the data to corresponding conditions:

```
targets <- read.csv(file="targets.csv", stringsAsFactors=FALSE)
```

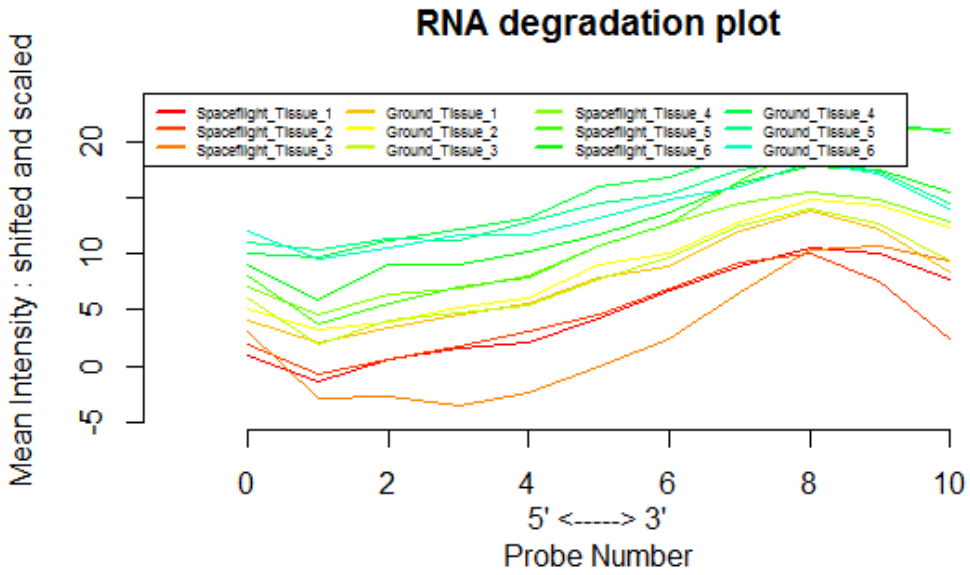
12. Fit a robust linear model to the probe level data of the raw microarray data (used to analyze overall quality of the microarrays):
    a. Subset arrays of interest:

```
data.Tissue <- data[,c(7,8,9,10,11,12,19,20,21,22,23,24)]
sampleNames(data.Tissue) <- targets$Name[c(7,8,9,10,11,12,19,20,21,22,23,24)]

data.PLM <- fitPLM(data.Tissue)
```

13. Assess RNA degradation (differentially degraded RNA will produce less accurate results):

```
RNAdeg <- AffyRNAdeg(data.Tissue)
colors <- palette(rainbow(12))
plotAffyRNAdeg(RNAdeg, col=colors)
legend("topleft", sampleNames(data.Tissue),lty=1,col=colors,lwd=2, cex=0.5, ncol = 4)
```



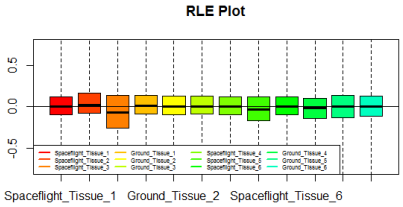
14. Save the RNA degradation measurements to a readable file:

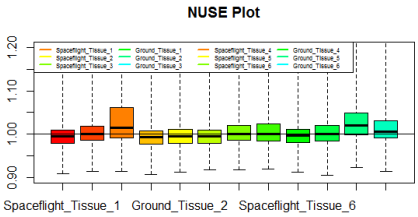
```
RNADegSummary <- summaryAffyRNAdeg(RNAdeg)
write.csv(RNADegSummary, file="RNADegSummary.csv")
```

15. Analyze the quality of the Relative Log Expression (RLE) and Normalized Unscaled Standard Error (NUSE) plots:

```
RLE(data.PLM, main="RLE Plot", col=colors)
legend("bottomleft", sampleNames(data.Tissue),lty=1,col=colors,lwd=2, cex=0.5, ncol = 4)

NUSE(data.PLM, main="NUSE Plot",col=colors)
legend("topleft", sampleNames(data.Tissue),lty=1,col=c(2,3,4,5,6,7),lwd=2, cex=0.5, ncol = 4)
```





16. Extract the experimental factors used in the study:

```
facs <- targets[,c(1,3)]
facs = paste(facs[,2], sep="")
f = factor(facs)
```

17. Create a design matrix, which matches .CEL files to their corresponding conditions:

```
design = model.matrix(~0+f)
colnames(design) = levels(f)
```

18. Create a contrast matrix for comparison of Spaceflight and Control microarrays:

```
cont.matrix = makeContrasts(Tissue_SpaceflightvsNorm = Spaceflight_Tissue-Ground_Tissue, levels=design)
```

19. Compute estimated coefficients and standard errors and compare expression levels by running an Empirical Bayes modified T-test with a Benjamini-Hochberg multiple hypothesis correction:

```
fit = lmFit(AEsetnorm, design)
fit2 = contrasts.fit(fit, cont.matrix)
fit2 = eBayes(fit2)
res = topTable(fit2, coef = "Tissue_SpaceflightvsNorm", adjust = "BH", number = Inf)
```

20. Create an optional threshold, to highlight genes with a log fold-change > 2 and p-value < 0.05:

```
res$threshold = as.factor(abs(res$logFC) > 2 & res$adj.P.Val < 0.05)
```

21. Download and load the correct annotation database for the microarrays used (all annotation files can be found at https://www.bioconductor.org/packages/release/data/annotation/):

```
source("https://bioconductor.org/biocLite.R")
biocLite("ath1121501.db ")
library("ath1121501.db")
library("org.At.tair.db")
library("GO.db")
```

22. Link each probe to its corresponding gene symbol and Entrez ID number:

```
TAIR_Link <- read.table(
"https://www.arabidopsis.org/download_files/Genes/TAIR10_genome_release/TAIR10_NCBI_mapping_files/
TAIR10_NCBI_GENEID_mapping",
sep="\t", header=FALSE, col.names= c("EntrezID", "LocusID"))

gene.ID <- mget(rownames(res),ath1121501ACCNUM, ifnotfound = NA)
ID <- sapply(gene.ID, paste, collapse=",")
res$LocusID <- ID
```

```
gene.Symbol <- mget(rownames(res),ath1121501SYMBOL, ifnotfound = NA)
Symbol <- sapply(gene.Symbol, paste, collapse=",")
res$Symbol <- Symbol

Locus <- as.character(res$LocusID)
LocusLink <- TAIR_Link$LocusID
EntrezLink <- TAIR_Link$EntrezID
x <- match(Locus, LocusLink)
y <- 1:length(rownames(res))
for (i in 1:length(x)){
y[i] <- EntrezLink[x[i]]
}
res$EntrezID <- y
```

23. Correct NA values for gene symbol and Entrez IDs:

```
FixSymbol <- res$Symbol
for (i in 1:length(FixSymbol)){
if (FixSymbol[i] == "NA"){
FixSymbol[i] <- NA
}
}
res$Symbol <- FixSymbol

FixLocus <- res$LocusID
for (i in 1:length(FixLocus)){
if (FixLocus[i] == "NA"){
FixLocus[i] <- NA
}
}
res$LocusID <- FixLocus
```

24. **OPTIONAL:** Link Genes to correct Homology Group via Homologene 68

```
homologene <- read.table(
"ftp://ftp.ncbi.nih.gov/pub/HomoloGene/build68/homologene.data",
sep="\t", quote= "", header=FALSE, col.names= c("HomologyGroup", "taxonomy_id",
"EntrezID", "symbol", "protein_gi", "protein_accession"))

#Load Packages for Parallel Processing
library(parallel)
library(dplyr)

# Calculate the number of cores
no_cores <- detectCores() - 1

# Initiate cluster
cl <- makeCluster(no_cores)

#Subset Homologene table to only contain genes from Arabidopsis
homologene1 <- subset(homologene, taxonomy_id == 3702)
```

```
# Send variables and non-base functions to the cluster
clusterExport(cl=cl, varlist=c("homologene", "res", "homologene1"), envir=environment())
clusterEvalQ(cl, {library(dplyr)})

# Parallel functions to build groups table
Link <- parLapply(cl, res$EntrezID, function(x) {
subtable <- subset(homologene1, EntrezID == x)
ifelse(nrow(subtable) == 0, return(NA), return((subtable$HomologyGroup)))})

res$HomologyGroup <- Link

#Stop the current cluster
stopCluster(cl)

#un-list column
res$HomologyGroup <- sapply(res$HomologyGroup, paste, collapse="")
res$HomologyGroup <- as.numeric(res$HomologyGroup)
```

25. Create a Volcano Plot to visualize differentially expressed genes (Genes highlighted in red passed the threshold that was set previously):

```
g = ggplot(data=res, aes(x=logFC, y=-log(P.Value), colour=threshold)) + scale_color_manual(values =
c("grey", "red")) +
geom_point(alpha=0.5, size=.5) +
theme(legend.position = "none") +
xlim(c(-4, 4)) + ylim(c(0, 15)) +
xlab("log2 fold change") + ylab("-log(P.Value)")
g
```

26. Add labels to top differentially expressed genes:

```
g + geom_label_repel(data=filter(res, adj.P.Val<0.05 & abs(logFC)>2), aes(label=LocusID))
```

27. Save Top Table results:

```
write.table(res,file="GLDS-17-Results.txt", sep = "\t")
write.csv(res,file="GLDS-17-Results.csv")
```

Final Table Layout:

	logFC	AveExpr	t	P.Value	adj.P.Val	threshold	B	LocusID	Symbol	EntrezID	HomologyGroup
264005_at	-1.81594	1.942487	-8.45243	1.32E-08	0.000262	FALSE	9.051565	AT2G22470	AGP2,ATAGP2	816779	NA
256964_at	-2.80853	7.59198	-8.19603	2.30E-08	0.000262	TRUE	8.602009	AT3G13520	AGP12,ATAGP1	820554	NA
251482_s_a	1.233989	1.616325	7.648756	7.65E-08	0.000446	FALSE	7.605378	NA	NA	NA	NA
265179_at	2.744925	7.064346	7.638569	7.82E-08	0.000446	TRUE	7.586347	AT1G23650	NA	838975	NA
263268_at	2.121249	7.880648	7.277752	1.77E-07	0.000807	TRUE	6.90108	NA	NA	NA	NA
245504_at	-0.75253	1.004657	-6.72833	6.32E-07	0.002404	FALSE	5.816352	AT4G15660	NA	827243	117552
257697_at	-1.75523	4.318688	-6.39968	1.38E-06	0.003979	FALSE	5.144607	AT3G12700	NANA	820452	91827
245277_at	1.605171	6.71488	6.394901	1.40E-06	0.003979	FALSE	5.134716	AT4G15550	IAGLU	827229	116182
248988_at	2.838965	5.218799	6.18935	2.29E-06	0.005799	TRUE	4.706184	AT5G45190	NA	834555	135548
264486_at	1.840646	5.938247	5.856098	5.15E-06	0.011747	FALSE	3.998831	AT1G77180	SKIP	844055	56557
260068_at	-1.10905	1.428425	-5.81289	5.73E-06	0.011873	FALSE	3.906039	AT1G73805	SARD1	843716	121800
250455_at	-1.09436	1.399494	-5.7193	7.21E-06	0.013701	FALSE	3.70426	AT5G09980	PROPEP4	830859	NA
250358_at	-2.74858	7.011322	-5.67399	8.06E-06	0.014142	TRUE	3.606204	AT5G11740	AGP15,ATAGP1	831046	NA
260913_at	-1.55441	6.557605	-5.55798	1.07E-05	0.017324	FALSE	3.35403	AT1G02500	AT1,METK1,SA	839501	68057
246305_at	1.214805	4.762209	5.53414	1.14E-05	0.017324	FALSE	3.302013	AT3G51890	CLC3	824352	119713
267070_at	-1.01579	1.257574	-5.49142	1.27E-05	0.017886	FALSE	3.208656	AT2G41000	NA	818700	NA