# 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
   1. This file should have the experimental conditions of each array saved within it
   2. 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
   3. 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 |

1. Now, open R-Studio and create a new project for the analysis
2. 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"))

1. 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")

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

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

* 1. To see your current working directory:

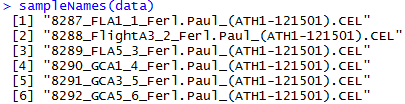
getwd()

1. 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)



|  |
| --- |
| **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 |

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

AEsetnorm <- affy::rma(data)

**OR**

AEsetnorm <- affy::mas5(data)

1. Create MA plots to evaluate the normalization:

for (i in 1:length(sampleNames(data)))

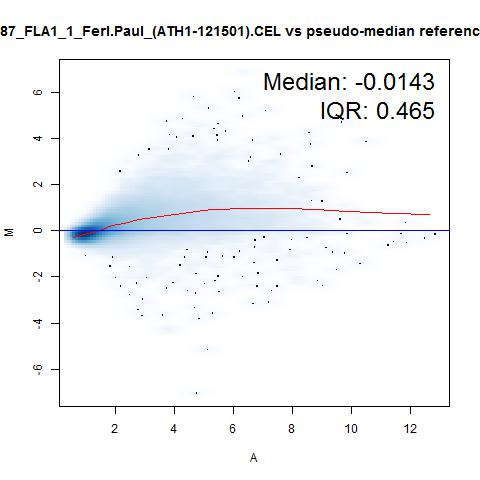
{

name = paste("MAplotnorm",i,".jpg",sep="")

jpeg(name)

oligo::MAplot(AEsetnorm,which=i)

dev.off()}



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

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

1. Fit a robust linear model to the probe level data of the raw microarray data (used to analyze overall quality of the microarrays):
   1. 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)

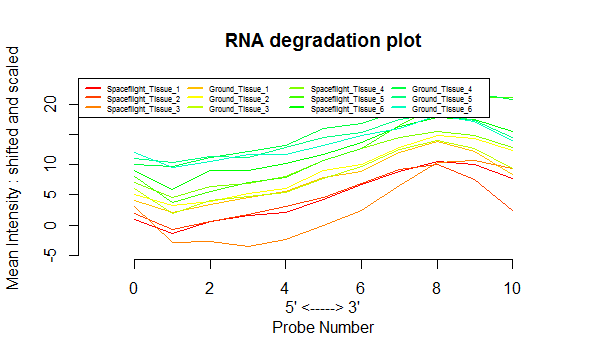
1. Assess RNA degradation (differentially degraded RNA will produce less accurate results):

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)



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

RNADegSummary <- summaryAffyRNAdeg(RNAdeg)

write.csv(RNADegSummary, file="RNADegSummary.csv")

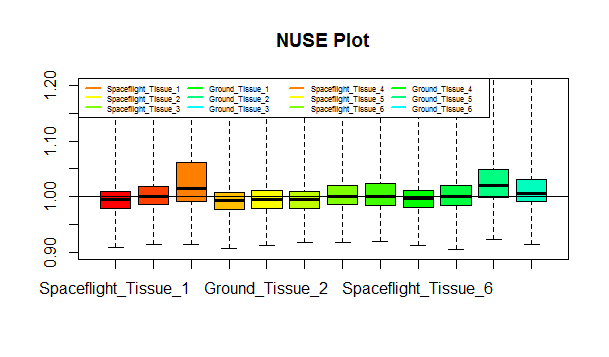
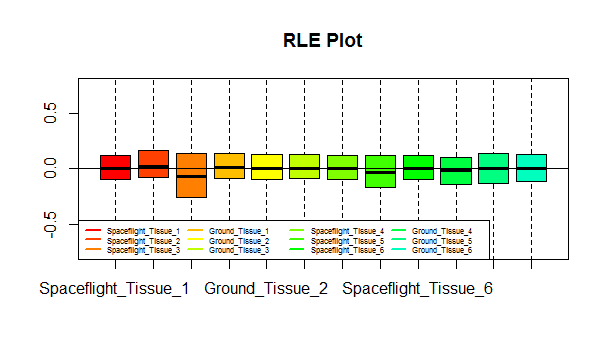
1. Analyze the quality of the Relative Log Expression (RLE) and Normalized Unscaled Standard Error (NUSE) plots:

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)



1. Extract the experimental factors used in the study:

facs <- targets[,c(1,3)]

facs = paste(facs[,2], sep="")

f = factor(facs)

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

design = model.matrix(~0+f)

colnames(design) = levels(f)

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

cont.matrix = makeContrasts(Tissue\_SpaceflightvsNorm = Spaceflight\_Tissue-Ground\_Tissue, levels=design)

1. 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)

1. 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)

1. 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")

1. 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

1. 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

1. **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)

1. 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") +

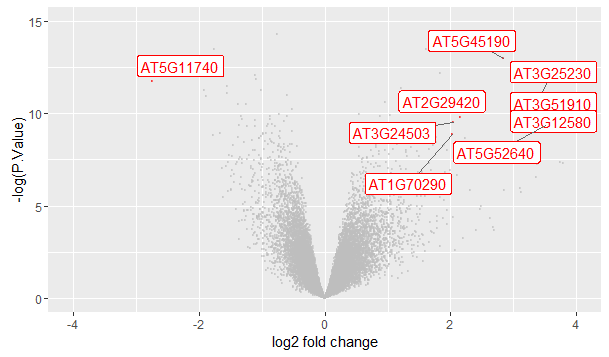
xlim(c(-4, 4)) + ylim(c(0, 15)) +

xlab("log2 fold change") + ylab("-log(P.Value)")

g

1. 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))



1. 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:**