Differential Expression Analysis

First we will load the necessary libraries.

```
library(tidyverse)
library(DESeq2)
library(magrittr)
library(edgeR)
```

The data has been extracted from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE226134 from the GSE226134_CK_10___norm.xlsx file. Lauren Mock selected for pre-treatment samples and performed data quality control.

```
data <- read.csv("data/input/JoinedWide.csv")

property <- as.data.frame(cbind(data$SegmentDisplayName, data$METASTATIC))
names(property)=c("SegmentDisplayName","METASTATIC")
normCountData <- data[,59:ncol(data)]
row.names(normCountData) <- data$SegmentDisplayName
```

The data given is normalized, but the properties include a normalization factor that we can divide the data by to get to integer counts, which are required for DESeq.

```
intCountData <- normCountData / data$NormalizationFactor
intCountData <- data.frame(lapply(intCountData, as.integer))
```

DESeq

```
dds <- DESeqDataSetFromMatrix(countData = t(intCountData),
                              colData = property,
                              design = ~METASTATIC)
```

```
Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
design formula are characters, converting to factors
```

DESeq recommends that row counts are filtered to remove rows with very few reads, especially rows with less than 10 reads. Here it appears that we end up keeping all rows.

```
keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep,]
```

To distinguish between metastatic and non-metatstatic samples we will relevel the factor where the reference is non-metastatic, here coded as false.

```
dds$METASTATIC <- relevel(dds$METASTATIC, ref = "False")
```

Next we will run DESeq to get the differentially expressed genes.

```
dds <- DESeq(dds)
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

-- note: fitType='parametric', but the dispersion trend was not well captured by the
   function: y = a/x + b, and a local regression fit was automatically substituted.
   specify fitType='local' or 'mean' to avoid this message next time.

final dispersion estimates

fitting model and testing

-- replacing outliers and refitting for 116 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)

estimating dispersions

fitting model and testing
```

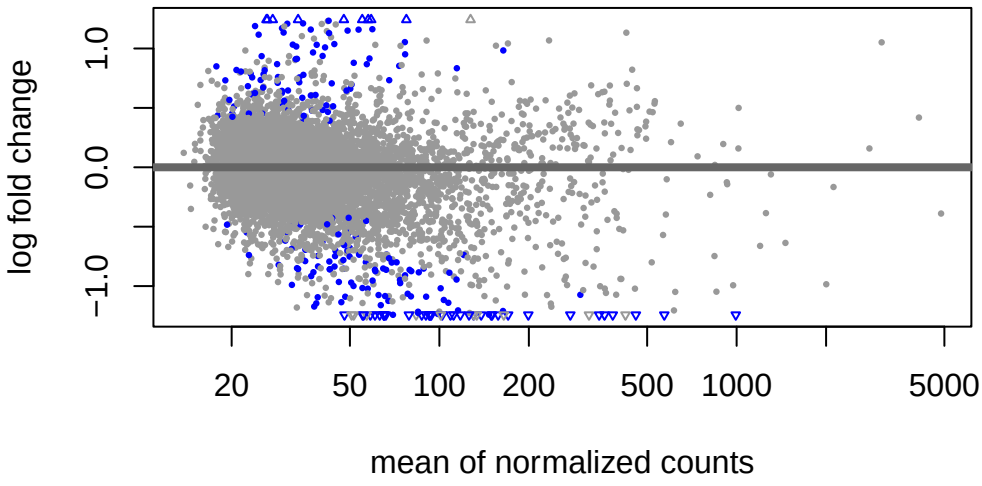
```r
adj_pval_threshold <- 0.01
res <- results(dds, alpha = adj_pval_threshold)
summary(res)
```

```
out of 9223 with nonzero total read count
adjusted p-value < 0.01
LFC > 0 (up)       : 89, 0.96%
LFC < 0 (down)     : 126, 1.4%
outliers [1]       : 0, 0%
low counts [2]     : 0, 0%
(mean count < 14)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

```r
resultsNames(dds)
```

```
[1] "Intercept"                 "METASTATIC_True_vs_False"
```

```r
DESeq2::plotMA(res)
```




```r
resOrdered <- res[order(res$padj),]
resSig <- subset(resOrdered, padj < 0.01)
gene_names <- rownames(resSig)
```

```r
for (gene in gene_names) {
  cat(gene, "\n")
} #running these genes through ShinyGO shows cancer and metabolism pathways http://bioinfo
```

```
TNC
MMP1
MT1G
DCXR
KRT1
F3
SERPINE1
FGFR1
TNFRSF12A
BAG4
ASH2L
INHBA
RAC3
LGR4
CXCL1
DUSP6
CBLB
ABCA1
PLAU
FLNA
PHLDA1
PMEPA1
SERPINE2
PODXL
PRSS23
GAGE1
DDHD2
LGALS1
PLPBP
SCAND1
AHCY
CTSB
DKK3
THBS1
CEP250
POLR1H
ROMO1
PDLIM7
```

FBXO32
KRT15
FHL2
NKX1.1
FNDC3B
MMP10
LPCAT2
ALDOC
NSD3
PLA2G4F
CPT1A
CAMSAP3
PTTG1IP
LIMA1
GLTP
ITGB1
ACSS2
MMP13
KAT2A
HEG1
TINAGL1
CHST15
APLP2
TES
DYNLT2B
FRAT2
LAMB3
JUN
SPTBN2
CADM4
SERINC2
GAGE10
FASN
CDC42EP3
TMEM132A
PDP1
B4GALT1
TGM3
ARL6IP5
EXT1
SORD
LSM1
DIS3

BMP1
VIM
ZYX
TNPO1
ADM
RNF152
ITGA5
CNN3
ERBB2
TPM1
JAG1
PPP6R3
PRODH
BRF2
NDUFV1
HGD
ETNK2
FAM83C
CT45A1
UQCC1
FAHD2B
MMP2
RAPGEFL1
RAPGEF1
FXYD5
ACTN1
TNFRSF21
LAMA3
NKD2
TPM4
BCAR1
PDXK
MARCKS
SEMA3C
FAM135A
CAV1
EFHD2
BPGM
ACSL4
MVD
FDXR
TMEM63C
MTX1

NDRG1
NDUFS8
SRD5A1
LAMC2
PLEKHA2
UBL7
TP53INP2
GSTA1
LLGL2
NR4A1
AKAP1
GAST
WDR1
SLFN5
CD47
CALU
LITAF
EPB41L1
TMEM39A
ASPH
PRNP
SIK1
EIF6
CAP1
IGFBP7
FKBP10
CAB39
HLA.DQB1
TGFBR1
MYO1B
CMTM6
SDC4
LCE2D
SULF2
CBFA2T2
PRMT2
ITGA3
ARHGAP21
AGFG1
RBM39
SERPINH1
SURF4
TRPC4AP

ELL2
CYRIB
LHFPL2
VOPP1
PTGFRN
SIRT6
CAPN2
GCLM
EFNA4
RBP1
CD68
EFNA3
RND3
BZW2
C3orf52
CKB
BAIAP2
CD24
CDH3
ANKRD35
ULK3
TLN1
ATP5MC2
SOGA1
KLC3
NLE1
GPR87
SYNPO
COL5A1
SH3TC1
CPNE1
TM2D2
LRP5
SAMM50
SCAMP1
MYADM
RHOC
ENAH
TUBB2B
SQSTM1
SUMO3
MRPL30
IVNS1ABP

```
B4GALNT3
PLOD2
CALD1
ADORA2B
THAP4
```

```
write.csv(as.data.frame(resOrdered),
          file="metastasis_results.csv")
```

EdgeR

Reference: https://web.stanford.edu/class/bios221/labs/rnaseq/lab_4_rnaseq.html

First we will prepare the data and calculate the dispersion so we will next be able to find the differential expression

```
d <- DGEList(counts=t(intCountData),group=property$METASTATIC)
dim(d)
```

```
[1] 9223   59
```

```
d.full <- d # keep the old one in case we mess up
head(d$counts)
```

```
       Sample1 Sample2 Sample3 Sample4 Sample5 Sample6 Sample7 Sample8 Sample9
AIP         67      12      22      24      58      39      27      26      35
AMY1A       39       6      21      22      45      26      20      66      36
PABIR1      89       9      23      22      66      32      30      33      30
POLR2F      91       9      19      40      94      84      39      47      46
MTA2        86       6      13      35     130      87      65      47      46
CDCA4       61       5      18      24      34      33      27      34      32
       Sample10 Sample11 Sample12 Sample13 Sample14 Sample15 Sample16 Sample17
AIP         153       64       42       14       38       49       23       19
AMY1A        59       74       48       10       51       58       41       26
PABIR1       59       69       57       16       40       37       31       12
POLR2F      119       66       89       30       29       63       30       33
MTA2        208      108      162       28       51       72       59       49
CDCA4       120       83       37       16       32       63       34       22
       Sample18 Sample19 Sample20 Sample21 Sample22 Sample23 Sample24 Sample25
```

AIP	13	33	68	8	4	26	94	123
AMY1A	19	34	75	52	13	39	132	124
PABIR1	19	45	65	12	10	25	95	138
POLR2F	43	56	107	12	25	53	265	201
MTA2	36	67	99	26	34	78	210	354
CDCA4	12	34	70	13	8	37	82	143
	Sample26	Sample27	Sample28	Sample29	Sample30	Sample31	Sample32	Sample33
AIP	38	15	67	29	33	1	4	49
AMY1A	46	15	83	23	20	4	7	31
PABIR1	31	7	125	29	31	1	7	45
POLR2F	87	30	210	36	68	6	3	112
MTA2	77	37	273	41	51	1	4	149
CDCA4	25	13	95	20	44	6	7	27
	Sample34	Sample35	Sample36	Sample37	Sample38	Sample39	Sample40	Sample41
AIP	8	33	22	78	13	30	44	120
AMY1A	3	38	28	201	14	14	45	51
PABIR1	8	46	35	102	9	17	61	93
POLR2F	6	61	20	187	12	26	63	195
MTA2	10	43	35	147	8	45	92	231
CDCA4	8	23	18	76	7	17	27	71
	Sample42	Sample43	Sample44	Sample45	Sample46	Sample47	Sample48	Sample49
AIP	27	25	4	31	49	48	17	1
AMY1A	40	31	5	28	46	79	15	13
PABIR1	35	23	7	29	46	57	15	4
POLR2F	31	29	4	32	124	88	17	13
MTA2	63	37	14	55	152	209	11	8
CDCA4	30	42	5	27	33	40	11	4
	Sample50	Sample51	Sample52	Sample53	Sample54	Sample55	Sample56	Sample57
AIP	10	40	28	63	232	48	24	19
AMY1A	9	63	29	54	134	11	24	19
PABIR1	9	30	31	52	105	22	22	15
POLR2F	8	30	55	132	362	50	20	19
MTA2	12	80	57	122	333	144	30	24
CDCA4	1	21	20	46	91	32	18	12
	Sample58	Sample59						
AIP	68	64						
AMY1A	47	93						
PABIR1	87	79						
POLR2F	122	62						
MTA2	185	117						
CDCA4	71	71						

```
apply(d$counts, 2, sum)
```

```
 Sample1  Sample2  Sample3  Sample4  Sample5  Sample6  Sample7  Sample8
  910619    49480   131443   348687   915774  1045122   649427   378320
 Sample9 Sample10 Sample11 Sample12 Sample13 Sample14 Sample15 Sample16
  380713  2014498   823838   802316   238557   883929   514514   496125
Sample17 Sample18 Sample19 Sample20 Sample21 Sample22 Sample23 Sample24
  361452   314568   442702  1115658   369882   156975   488535  1658825
Sample25 Sample26 Sample27 Sample28 Sample29 Sample30 Sample31 Sample32
 2145876   485456   291752  2437146   381860   524198    39357    42942
Sample33 Sample34 Sample35 Sample36 Sample37 Sample38 Sample39 Sample40
  970369    86043   883825   254777  1124356   122495   309227   567186
Sample41 Sample42 Sample43 Sample44 Sample45 Sample46 Sample47 Sample48
 2001258   517700   323444    57887   528169  1187802  1153804   211023
Sample49 Sample50 Sample51 Sample52 Sample53 Sample54 Sample55 Sample56
   64074   108979   528054   500160   802961  2543986   474835   303885
Sample57 Sample58 Sample59
  152894  1453165  1118159
```

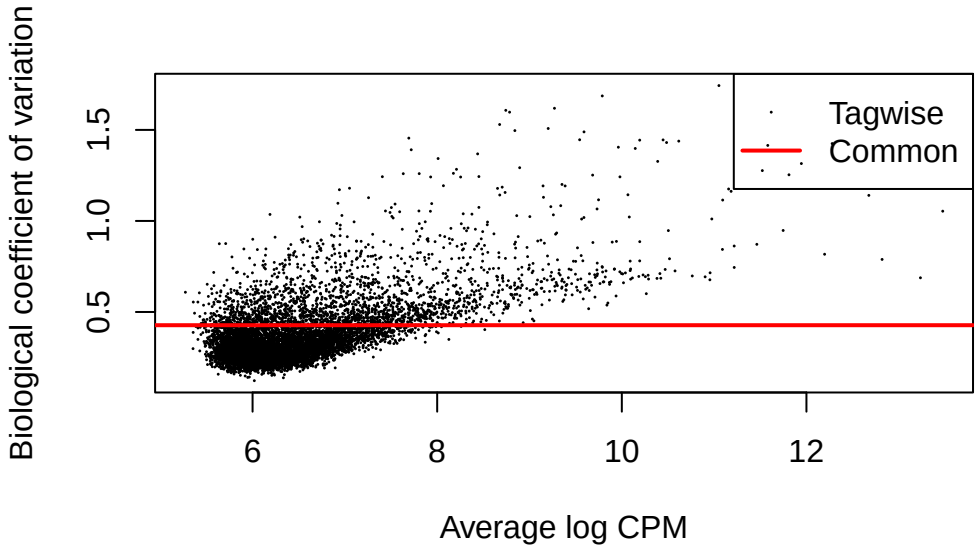
```
#filtering steps for DESeq
keep <- rowSums(cpm(d)>100) >= 2
d <- d[keep,]
dim(d) #cuts down about 600 genes
```

```
[1] 8680   59
```

```
d$samples$lib.size <- colSums(d$counts)
d <- calcNormFactors(d)
d1 <- estimateCommonDisp(d, verbose=T)
```

```
Disp = 0.18261 , BCV = 0.4273
```

```
d1 <- estimateTagwiseDisp(d1)
plotBCV(d1)
```



We will now use our information from the dispersion calculation to check for differential expression and then compare to the results from DESeq.

```
et12 <- exactTest(d1, pair=c(1,2))
topTags(et12, n=10)
```

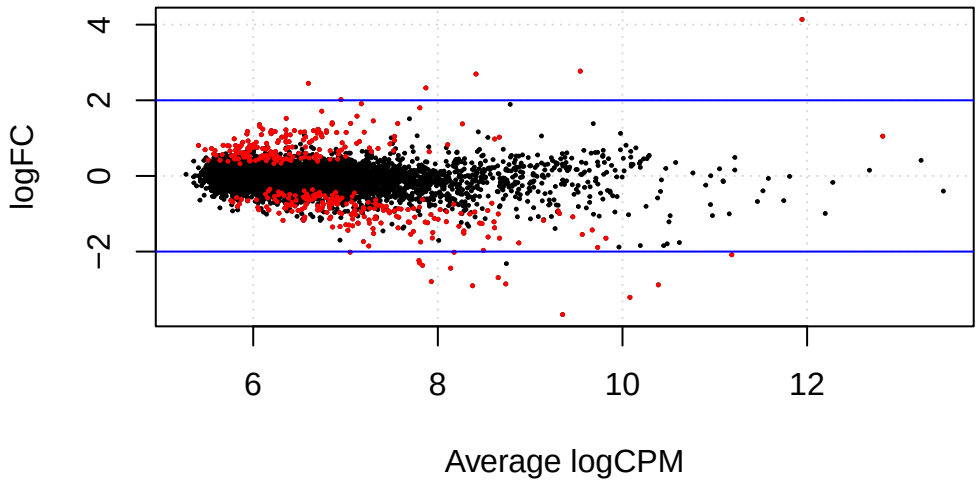
```
Comparison of groups:  True-False
           logFC    logCPM       PValue          FDR
PITX1  4.1377711 11.945164 1.817350e-15 1.577460e-11
MT1G   1.9075098  7.171275 5.273978e-13 1.806617e-09
ANK1   2.6948864  8.413772 6.244069e-13 1.806617e-09
GSTA1  2.4484216  6.597395 4.285723e-12 9.300019e-09
DCXR   1.1614314  6.191021 4.497128e-11 7.807015e-08
ASH2L  1.3051355  6.072951 1.168861e-09 1.690953e-06
TNC   -3.2101590 10.080863 1.991779e-09 2.438533e-06
BAG4   1.0346380  6.745474 2.247496e-09 2.438533e-06
LUM    2.3276325  7.870496 2.841152e-09 2.740133e-06
FGFR1  0.9251594  5.944759 4.159412e-09 3.610369e-06
```

```
de1 <- decideTestsDGE(et12, adjust.method="BH", p.value=0.05)
summary(de1)
```

```
       True-False
Down          211
```

```
NotSig        8296
Up             173
```

```
de1tags12 <- rownames(d1)[as.logical(de1)]
plotSmear(et12, de.tags=de1tags12)
abline(h = c(-2, 2), col = "blue")
```




```
tags <- topTags(et12, n=Inf)
top_genes <- rownames(tags$table)[tags$table$FDR < 0.05 & abs(tags$table$logFC) > 1]
```

```
sum(top_genes %in% rownames(resSig))
```

```
[1] 85
```

```
matching_genes <- top_genes[top_genes %in% rownames(resSig)]
for (gene in matching_genes) {
  cat(gene, "\n")
}
```

```
MT1G
GSTA1
DCXR
ASH2L
TNC
```

BAG4
GAGE1
RAC3
MMP1
SERPINE1
F3
DDHD2
PODXL
PLPBP
TNFRSF12A
AHCY
DUSP6
CPT1A
FLNA
INHBA
GLTP
PHLDA1
PLAU
NSD3
PMEPA1
CXCL1
TGM3
PRODH
JUN
PRSS23
LGALS1
CT45A1
ALDOC
CTSB
SERPINE2
THBS1
GAGE10
DKK3
ETNK2
PDLIM7
LSM1
NR4A1
FHL2
MMP13
RAPGEFL1
ITGB1
FBXO32
MMP10

PLEKHA2
LAMB3
KRT15
SERINC2
TM2D2
B4GALT1
TINAGL1
ITGA5
VIM
TPM1
TPM4
CNN3
ADM
FXYD5
MMP2
ACTN1
TNFRSF21
SEMA3C
CAV1
LAMA3
SIK1
ASPH
IGFBP7
GCLM
SULF2
ITGA3
LAMC2
SH3TC1
NDRG1
SDC4
SERPINH1
HLA.DQB1
CDH3
LCE2D
COL5A1
RBP1
CD24

We can see that there are 85 genes in common between the results from DESeq and from EdgeR. When we plug in the overlapping genes into ShinyGO, we see pathways enriched for receptor interactions, cancer, adhesion, and signaling pathways, which make sense given the biological basis of metastasis.