Analysis of miRNA data in HNSCC

Elana J. Fertig
ejfertig@jhmi.edu

February 1, 2012

Contents

Chapter 1

Analysis Code

1.1 Package Requirements

```
> library('Biobase')
> library('ddCt') # package for processing RT-PCR data
> library('simpleaffy')
> library('ClassDiscovery')
> library('biomaRt') # to extract location of miRNA along chromosome
> library('limma')
> library('AnnotationDbi')
> library('gtools')
> library('xtable')
> library('IRanges')
> library('Hmisc')
> library('mirbase.db')
```

1.2 File structure

```
> dataDir <- function(x='') {
+   return(file.path(getwd(),'Data',x))
+ }
> resultsDir <- figsDir <- figDir <- function(x='') {
+     return(file.path(getwd(),'Figs','CorrectedNorm',Sys.Date(),x))
+ }
> if (!file.exists(figDir())) {
+   dir.create(figDir(), recursive=T)
+ }
>
```

1.3 Preprocessing

We start by reading in the raw RT-PCR data with the functions provided in the `ddCt` and `Biobase` packages.

```
> file.names <- c(dataDir('Head & Neck 4controls TLDA 5-11-09.txt'),
+   dataDir('head&neck1.txt'), dataDir('head&neck2.txt'), dataDir('head&neck3.txt'),
+   dataDir('head&neck4.txt'))
> rawData <- SDMFrame(file.names)
> detectorNames <- unique(rawData$Detector)
> sampleNames <- unique(rawData$Sample)
```

Once processed, we reformat the raw RT-PCR data into a matrix of Ct data for each miRNA (row) and tumor (column).

```
> sample.idx <- sapply(sampleNames,function(x){which(rawData$Sample==x)})
> ct.matrix <- ct.undetected.matrix<- matrix(0,nrow=length(detectorNames),ncol=length(sampleNames))
> row.names(ct.matrix) <- row.names(ct.undetected.matrix) <- detectorNames
> colnames(ct.matrix) <- colnames(ct.undetected.matrix) <- sampleNames
> for (sample in sampleNames) {
+   ct.data <- ct.undetected <- rawData$Ct[sample.idx[[sample]]]
+   ct.data[which(is.na(ct.data))] <- 40
+   detectors <- rawData$Detector[sample.idx[[sample]]]
+   ct.matrix[,sample] <- sapply(detectorNames, function(x){
+       detectorIdx <- grep(x, detectors)
+       return(mean(ct.data[detectorIdx],na.rm=T))
+     })
+   ct.undetected.matrix[,sample] <- sapply(detectorNames, function(x) {
+     detectorIdx <- grep(x, detectors)
+     if (all(is.na(ct.undetected[detectorIdx]))) return(NA)
+     return(mean(ct.undetected[detectorIdx],na.rm=F))
+   })
+ }
```

In order to compare miRNA expression in tumors and normals in subsequent analyses, we must link the data to further information about the tumor samples provided in an additional annotation file.

```
> rttab=as.matrix(read.table(dataDir(’HPV miRNA small table.txt’), header=T, as.is=T, sep=’\t’))
> rownames(rttab)=rttab[,’VMSR.ID’]
> SampleType <- rep(’Tumor’,nrow(rttab))
> rttab <- cbind(rttab,SampleType)
> # get the file names for each sample
> sampleFile <- unique(cbind(rawData$Sample,rawData$Platename))
> row.names(sampleFile) <-sampleFile[,1]
> sampleFile <- sampleFile[colnames(ct.matrix),]
> # append information about normals
> normalSamples <- sampleFile[which(sampleFile[,2]=="Head & Neck 4controls TLDA 5-11-09.txt"),1]
> normalInfo <- matrix(NA,nrow=length(normalSamples),ncol=ncol(rttab))
> row.names(normalInfo) <- normalSamples
> colnames(normalInfo) <- colnames(rttab)
> normalInfo[,’VMSR.ID’] <- row.names(normalInfo)
> normalInfo[,’SampleType’] <- rep(’Normal’,nrow(normalInfo))
> rttab <- rbind(rttab,normalInfo)
> # append information about files (i.e., batches)
> RTPCR.file <- sampleFile[intersect(row.names(rttab),sampleFile[,1]),2]
> rttab <- cbind(rttab,RTPCR.file)
> # remaining samples are replicates, indicate that
> replicate <- rep(1,nrow(rttab))
> rttab <- cbind(rttab,replicate)
> replicateSample <- setdiff(sampleFile[,1], rttab[,’VMSR.ID’])
> # only indicate replicate samples for which reference gene
> # was detected
> ref.gene <- c(’MammU6-4395470’)
> validSamples <- sampleNames[which(!is.nan(ct.matrix[ref.gene,]))]
> replicateSample <- intersect(validSamples,replicateSample)
> # indicate replicate information in annotation
> replicateID <- rttab[grep(replicateSample,rttab[,’multiple.samples’]),
+                       'VMSR.ID']
> replicateInfo <- rttab[paste(replicateID,sep='.'),]
> replicateInfo['RTPCR.file'] <- sampleFile[replicateSample,2]
> replicateInfo['replicate'] <- 2
> rttab <- rbind(rttab,replicateInfo)
> row.names(rttab)[nrow(rttab)] <- replicateSample
```

1.3.1 Distribution of miRNA expression

Distribution of raw Ct counts, substituting the maximum value of 40 when undetectable. This analysis shows that sample 519CC26 has limited range of detection and should be disregarded from the analysis.

```
> pdf(figsDir('miRNADistributionDetected.pdf'))
> sampleColors <- rep(gray(0.6), ncol(ct.matrix))
> names(sampleColors) <- colnames(ct.matrix)
> sampleColors[normalSamples] <- 'white'
> boxplot(ct.matrix,las=2,col=sampleColors[colnames(ct.matrix)],
+        ylab='Ct',
+        ylim=c(max(ct.matrix,na.rm=T), min(ct.matrix,na.rm=T)))
> points(x=1:ncol(ct.matrix),y=ct.matrix[ref.gene,],
+       pch=18,col='red')
> dev.off()

null device
          1

> pdf(figsDir('miRNADistributionLegend.pdf'))
> par(mfrow=c(1,2))
> plot(c(0,1),c(0,1),col='white')
> legend('topleft',col=c('red'), pch=c(18), legend=c(ref.gene))
> plot(c(0,1),c(0,1),col='white')
> legend('topleft',fill=c('white',gray(0.6)), border='black', legend=c('normal','HNSCC'))
> dev.off()

null device
          1
```

We observe that miRNA are underexpressed (and often undetected) in normals relative to cancers. We also note that the C_t count is notably higher in the normal samples than the tumor samples.

```
> validSamples <-setdiff(colnames(ct.matrix),c(replicateSample,'519CC26','519CC74'))
> tumorSamples <- validSamples[which(!is.na(rttab[validSamples,'HPV.status']))]
> tumorRef <- median(ct.matrix[ref.gene,tumorSamples], na.rm = T)
> del.ct.data <- sweep(ct.matrix[setdiff(row.names(ct.matrix),ref.gene),],
+                     2,ct.matrix[ref.gene,])
> del.ct.undetected.data <-
+     sweep(ct.undetected.matrix[setdiff(row.names(ct.undetected.matrix),
+                                        ref.gene),],2, ct.undetected.matrix[ref.gene,])
> del.ct.renorm <- del.ct.data
> del.ct.undetected.renorm <- del.ct.undetected.data
```

We will also average replicate samples for VMSR ID 519CC63 and remove the sample 519CC26 as an outlier.

```
> validGenes <- names(which(apply(del.ct.data[,validSamples],1,function(x){all(!is.na(x))})))
> correctReplicate <- function(data) {
```

```
+   outputData <- data[,setdiff(colnames(data),replicateSample)]
+   outputData[,replicateID] <- apply(data[,c(replicateID,replicateSample)],1,mean)
+   outputData <- outputData[validGenes,intersect(colnames(outputData),validSamples)]
+   return(outputData)
+ }
> replicate.ct.data <- correctReplicate(ct.matrix)
> replicate.ct.undetected <- correctReplicate(ct.undetected.matrix)
> replicate.del.ct.data <- correctReplicate(del.ct.data)
> replicate.del.ct.undetected <- correctReplicate(del.ct.undetected.data)
> replicate.del.ct.renorm <- correctReplicate(del.ct.renorm)
> replicate.del.ct.renorm.undetected <- correctReplicate(del.ct.undetected.renorm)
```

Clustering of resulting, normalized miRNA ΔC_t values.

```
> pdf(figsDir('miRNADistributionDelCt.pdf'))
> sampleColors <- rep(gray(0.6), ncol(replicate.del.ct.renorm))
> names(sampleColors) <- colnames(replicate.del.ct.renorm)
> sampleColors[normalSamples] <- 'white'
> boxplot(replicate.del.ct.renorm,las=2,
+           col=sampleColors[colnames(replicate.del.ct.renorm)],
+           ylab=expression(Delta * 'Ct'),
+         ylim=c(max(replicate.del.ct.renorm,
+                    na.rm=T),
+                min(replicate.del.ct.renorm,
+                    na.rm=T)))
> dev.off()

null device
          1
```

1.4 Differential miRNA expression

We use the limma package to determine differential expression of miRNA between tumors and normals. These statistics and the corresponding Benjamini-Hotchberg [1] adjusted p-values are output into /Users/ejfertig/Documents/Data/F Analysis_Projects/ChristineChung/RTPCRmiRNA/RTPCRmiRNA/Figs/CorrectedNorm/2012-02-01/miRNATumorDiffExp

```
> tumorsKept <- setdiff(colnames(replicate.del.ct.renorm),
+                         normalSamples)
> tumorType <- rep('normal',ncol(replicate.del.ct.renorm))
> names(tumorType) <- colnames(replicate.del.ct.renorm)
> tumorType[tumorsKept] <- 'tumor'
> tumorType.design <- model.matrix(~0+tumorType)
> tumorType.fit <- lmFit(replicate.del.ct.renorm, tumorType.design)
> tumorType.contrasts <- makeContrasts(tumorTypenormal-tumorTypetumor,
+       levels = tumorType.design)
> tumorType.contrasts.fit <- eBayes(contrasts.fit(tumorType.fit,
+                                                  tumorType.contrasts))
> tumorType.results <- topTable(tumorType.contrasts.fit,
+                                 number=nrow(tumorType.contrasts.fit),
+                               p.value=1,
+                                 sort.by="logFC")[,
+                               c('ID','logFC','adj.P.Val')]
> tumorType.results.out <- cbind(tumorType.results[,'ID'],
+                                round(tumorType.results[,'logFC'],2),
+                                format(signif(tumorType.results[,'adj.P.Val'],2),scientific=T))
```

```
> per.normal.NA <- apply(ct.undetected.matrix[tumorType.results.out[,
+                                                    1],
+                             normalSamples], 1,
+                   function(x){length(which(is.na(x))) / length(x)})
> per.tumor.NA <- apply(ct.undetected.matrix[tumorType.results.out[,
+                                                    1],
+                             which(tumorType==’tumor’)], 1,
+                   function(x){length(which(is.na(x))) / length(x)})
> tumorType.results.out <- cbind(tumorType.results.out,
+                                format(signif(per.normal.NA,2)),
+                                format(signif(per.tumor.NA,2)))
> colnames(tumorType.results.out) <- c(’miR ID’, ’Delta Delta Ct’,
+                                      ’Adj p-value’,
+                                      ’Percentage of undetected normals’,
+                                      ’Percentage of undetected tumors’)
> write.table(tumorType.results.out, row.names=F, sep="\t",
+             file=figsDir(’miRNATumorDiffExprs.txt’),
+             quote=F)
> miRNA05Thresh <- tumorType.results[which(tumorType.results[,
+                                    ’adj.P.Val’] < 0.05),’ID’]
> downreg <- miRNA05Thresh[which(tumorType.results[match(miRNA05Thresh,
+                                                  tumorType.results[,’ID’]),
+                           ’logFC’]<0)]
> upreg <- setdiff(miRNA05Thresh,downreg)
```

Plotting the distribution of differentially-expressed miRNA in Figure 3.3.

```
> pdf(figsDir(’SignatureExprsBoxplot.pdf’))
> boxplot(t(replicate.del.ct.renorm[tumorType.results[which(tumorType.results[,’adj.P.Val’]<=0.05),
+                                                      ’ID’],])˜tumorType,
+         names=c(’Normal’,’HNSCC’),
+         ylab=expression(Delta * ’Ct’),
+         ylim=c(max(replicate.del.ct.renorm),
+                min(replicate.del.ct.renorm)))
> dev.off()
```

```
null device
          1
```

1.4.1 HPV negative vs HPV positive

```
> tumorStatus <- (rttab[colnames(replicate.del.ct.renorm),’HPV.status’])
> tumorStatus[which(is.na(tumorStatus))] <- ’normal’
> tumorStatus <- factor(tumorStatus)
> HPVTumor.design <- model.matrix(˜0+tumorStatus)
> HPVTumor.fit <- lmFit(replicate.del.ct.renorm,
+                       HPVTumor.design)
> HPVTumor.contrasts <- makeContrasts(tumorStatusneg-tumorStatuspos,
+       levels = HPVTumor.design)
> HPVTumor.contrasts.fit <- eBayes(contrasts.fit(HPVTumor.fit,
+                                                HPVTumor.contrasts))
>  HPVSigResults <- topTable(HPVTumor.contrasts.fit,
+                            number=nrow(HPVTumor.contrasts.fit),
+                            p.value=0.05,adjust.method=’BH’)
>
```

1.5 miRNA families

```
> detectorMirs <- sapply(strsplit(tumorType.results[,’ID’],split=’-’),function(x){
+   if (x[length(x)-1]!=’3p’ && x[length(x)-1]!=’5p’) {
+     return(paste(x[1:(length(x)-1)],collapse=’-’))
+   } else {
+     return(paste(x[1:(length(x)-2)],collapse=’-’))
+   } })
> names(detectorMirs) <- tumorType.results[,’ID’]
> miRNAID <- names(as.list(mirbaseID2ACC))
> lapply(detectorMirs,function(x){setdiff(grep(x,miRNAID,ignore.case=T,value=T),grep(paste(x,’[0-9]’,sep
> mirbase2detector <- revmap(detector2mirbase)
> mirtypemaped <- lapply(mirbase2detector,function(x){sapply(strsplit(x,
+                         split=’-’), function(y){y[3]})})
> for (m in names(mirtypemaped)) {
+   mt <- strsplit(m,split=’-’)[[1]][3]
+   if (is.na(mt)) next
+   mirbase2detector[[m]] <- mirbase2detector[[m]][which(mirtypemaped[[m]]==mt)]
+ }
> mirbase2detector <- mirbase2detector[which(sapply(mirbase2detector,
+                                                    length)>0)]
> detector2mirbase <- revmap(mirbase2detector)
> mirnaFamily <- revmap(as.list(mirbaseFAMILY)[unlist(detector2mirbase)])
> lapply(mirnaFamily,unique) -> mirnaFamily
> mirnaFamily[which(sapply(mirnaFamily,length)>1)] -> mirnaFamily
> mirnaFamilyDetector <- lapply(mirnaFamily, function(x){unique(unlist(mirbase2detector[x]))})
> mirnaFamilyDetector[which(sapply(mirnaFamilyDetector,length)>1)] -> mirnaFamilyDetector
> mirnaFamilyStats <- cbind(names(mirnaFamilyDetector),
+                            sapply(mirnaFamilyDetector,length),
+                            sapply(mirnaFamilyDetector,
+                                   function(x){length(intersect(upreg,x))}),
+                            sapply(mirnaFamilyDetector,
+                                   function(x){length(intersect(downreg,
+                                                                x))}),
+                            sapply(mirnaFamilyDetector,
+                                   function(x){paste(x, collapse=’, ’)}),
+                            sapply(mirnaFamilyDetector,
+                                   function(x){paste(intersect(x,upreg),
+                                                     collapse=’, ’)}),
+                            sapply(mirnaFamilyDetector,
+                                   function(x){paste(intersect(x,downreg),
+                                                     collapse=’, ’)}))
> colnames(mirnaFamilyStats) <- c(’mirbase family’,
+                                  ’number on detector and in family’,
+                                  ’number upregulated’,
+                                  ’number downregulated’,
+                                  ’miRNA on detector and in family’,
+                                  ’upregulated miRNA’,
+                                  ’downregulated miRNA’)
> write.table(mirnaFamilyStats,file=figDir(’mirbaseFamilyStats.txt’),
+             sep="\t", row.names=F)
> mircluster <- lapply(as.list(mirbaseCLUSTER),
+                       function(x){paste(sort(x),collapse=’,’)})
> mircluster <- unique(unlist(unique(unlist(mircluster))))
> mircluster <- mircluster[grep(',',mircluster)]
> mircluster <- strsplit(mircluster,split=',')
> mirnaClusterDetector <- lapply(mircluster,
+   function(x){unique(unlist(mirbase2detector[x]))})
> mirnaClusterDetector <- mirnaClusterDetector[which(!sapply(mirnaClusterDetector,is.null))]
> mirnaClusterDetector <- mirnaClusterDetector[which(sapply(mirnaClusterDetector,length)>1)]
> mirnaClusterStats <- cbind(sapply(mirnaClusterDetector,length),
+                           sapply(mirnaClusterDetector,
+                                  function(x){length(intersect(upreg,x))}),
+                           sapply(mirnaClusterDetector,
+                                  function(x){length(intersect(downreg,
+                                                               x))}),
+                           sapply(mirnaClusterDetector,
+                                  function(x){paste(x, collapse=', ')}),
+                           sapply(mirnaClusterDetector,
+                                  function(x){paste(intersect(x,upreg),
+                                                    collapse=', ')}),
+                           sapply(mirnaClusterDetector,
+                                  function(x){paste(intersect(x,downreg),
+                                                    collapse=', ')}))
> colnames(mirnaClusterStats) <- c('number on detector and in cluster',
+                                  'number upregulated',
+                                  'number downregulated',
+                                  'miRNA on detector and in cluster',
+                                  'upregulated miRNA',
+                                  'downregulated miRNA')
> write.table(mirnaClusterStats,file=figDir('mirnaClusterStats.txt'),
+             sep="\t", row.names=F)
>
```

1.6 Analysis summary

```
> sessionInfo()

R version 2.14.0 (2011-10-31)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] splines   stats     graphics  grDevices utils     datasets  methods  
[8] base     

other attached packages:
 [1] mirbase.db_0.7.0      RSQLite_0.11.1        DBI_0.2-5            
 [4] Hmisc_3.9-1           survival_2.36-10      IRanges_1.12.5       
 [7] gtools_2.6.2          AnnotationDbi_1.16.11 limma_3.10.2         
[10] biomaRt_2.10.0        ClassDiscovery_2.13.3 PreProcess_2.12.1    
[13] oompaBase_2.14.0      mclust_3.4.11         cluster_1.14.1       
[16] simpleaffy_2.30.0     gcrma_2.26.0          genefilter_1.36.0    
[19] affy_1.32.0           ddCt_1.8.0            lattice_0.20-0       
[22] xtable_1.6-0          RColorBrewer_1.0-5    Biobase_2.14.0       
```

```
loaded via a namespace (and not attached):
 [1] affyio_1.22.0         annotate_1.32.1       BiocInstaller_1.2.1
 [4] Biostrings_2.22.0     grid_2.14.0           preprocessCore_1.16.0
 [7] RCurl_1.9-5           tools_2.14.0          XML_3.9-2
[10] zlibbioc_1.0.0
```

```
> save(list=ls(),
+      file=figDir(paste(’HPVAnalysis’,format(Sys.time(),’%d%b%Y’),
+                        ’Rda’,sep=’.’)))
```

Chapter 2

Methods

2.1 RT-PCR miRNA normalization

Prior to normalization, Figure 3.2(a) plots the distribution of C_t counts and value of the reference gene. In this plot and subsequent analysis, any value indicated as not detected is replaced with the maximum C_t count of 40. This boxplot reveals that the majority of miRNA for sample 519CC26 have this maximum C_t value of 40, and so this sample is also removed from the analysis as an outlier. We also remove sample 519CC74 from the analysis because MammU6-4395470 is not detected.

We then normalize the RT-PCR miRNA data by subtracting the C_t counts of the endogenous control gene MammU6-4395470 from the raw C_t counts to obtain ΔC_t values consistent with [2, 4]. This normalization results in the distribution of ΔC_t counts in Figure 3.2(b). For these final ΔC_t values, replicate samples for a single patient (519CC63 and 519CC64) are averaged for subsequent analyses.

2.2 Differential miRNA expression analysis

Empirical Bayes moderated t-statistics were used to assess differential expression between $\Delta\Delta C_t$ of miRNA associated with HNSCC or HPV status. The Benjamini-Hotchberg correction [1] was applied to the resulting p-values to account for multiple hypothesis testing. All analyses were performed with the LIMMA Bioconductor package [3] implemented in R. All analyses are documented in a Sweave report, and included as a Supplemental File in the paper.

2.3 Annotation of miRNA families and clusters

MicroRNA families and clusters are identified from the miRBase [5, 6, 7] released in the Bioconductor package mirbase.db (version 0.7.0) [8]. In this package, sets of miRNA within a 10kb window are assigned to genomic clusters.

To identifying these families or clusters from mirbase for the miRNA measured on the RT-PCR, we first map the probe names to mirbase miRNA names. In this analysis, both the miRNA on the 3p and 5p arms measured with separate probes in the RT-PCR array are assigned to a single miRNA name listed in the mirbase database. Finally, a single detector for a miRNA can be assigned to multiple miRNA named in the mirbase database (e.g., hsa-miR-9-4373285 is referenced to mirbase miRNA hsa-mir-9-1, hsa-mir-9-2, hsa-mir-9-3.

Chapter 3

Results

3.1 Results of differential miRNA analysis

The results presented in this section summarize the miRNA that are differentially expressed between the specified categories with Benjamini-Hotchberg adjusted p-value less than 0.05 (see Methods section).

Downregulated in HNSCC vs normal: 103, distribution of ΔC_t values in Figure 3.3.

Upregulted in HNSCC vs normal: 154, distribution of ΔC_t values in Figure 3.3.

Upregulated in HPV+ vs HPV-: hsa-miR-449a-4373207, $\Delta\Delta C_t$ = 2.76, adj. p-value 0.0262.

Upregulated in HPV- vs HPV+: hsa-miR-129-3p-4373297, $\Delta\Delta C_t$ = -3.82, adj. p-value 0.0262.

hsa-miR-205-4373093, $\Delta\Delta C_t$ = 4.58173, adj. p-value 0.00104424.

3.2 Figures

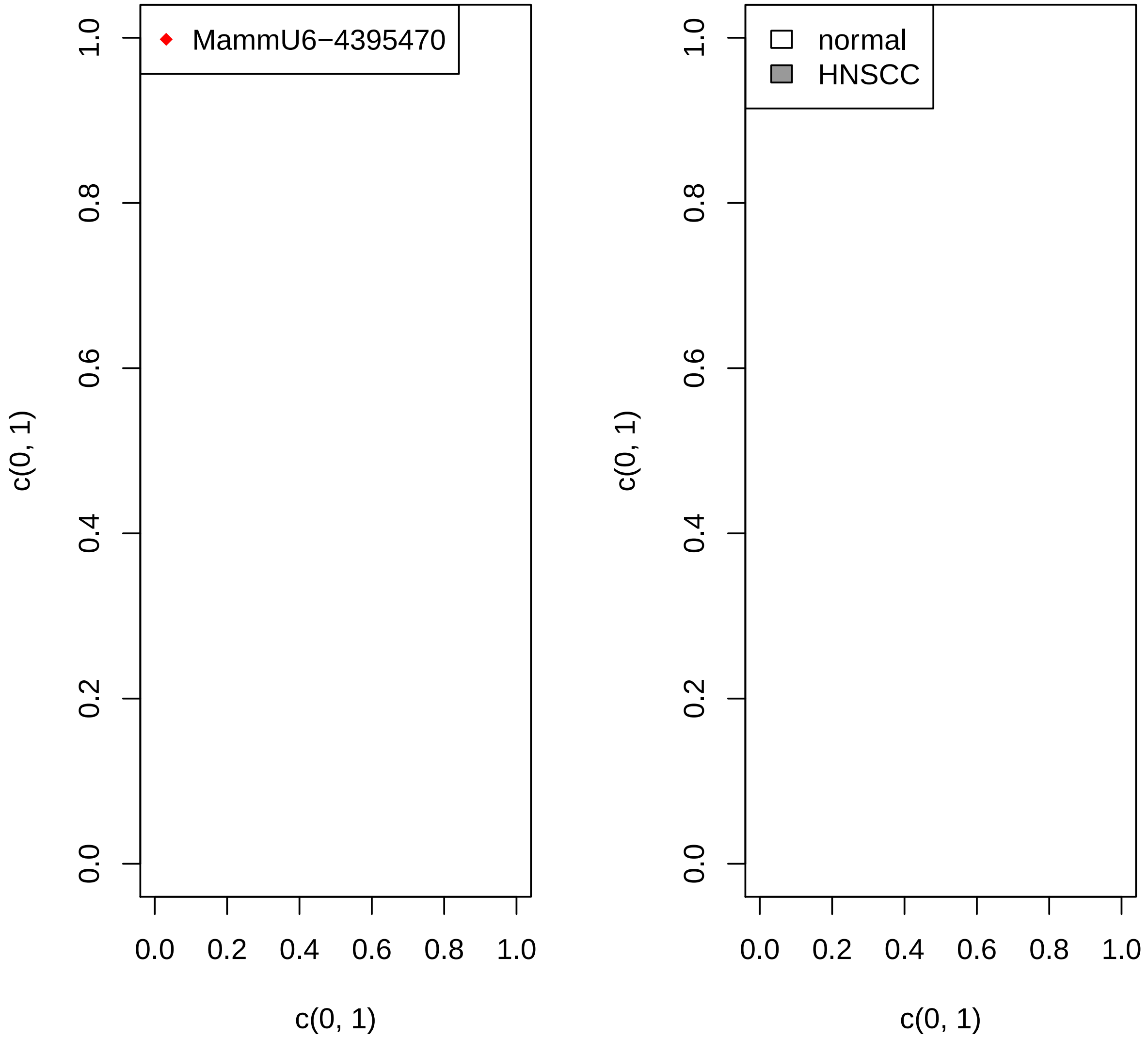

Figure 3.1: Legends for Figure 3.2.

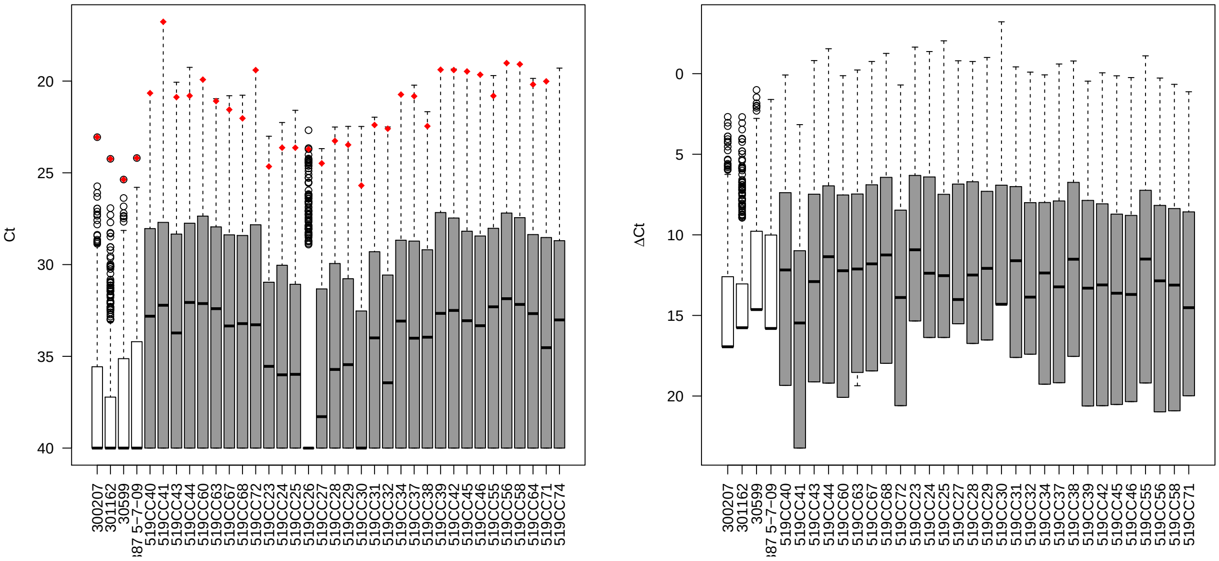

Figure 3.2: (a) Distribution of C_t counts in normal (white) and HNSCC (grey) samples. The C_t value for the endogenous control is plotted in each sample for which it is detected (red). A maximum value of 40 is assigned to any miRNA not detected by the assay. (b) Distribution of ΔC_t counts normalized to the value of the endogenous control reference MammU6-4395470.

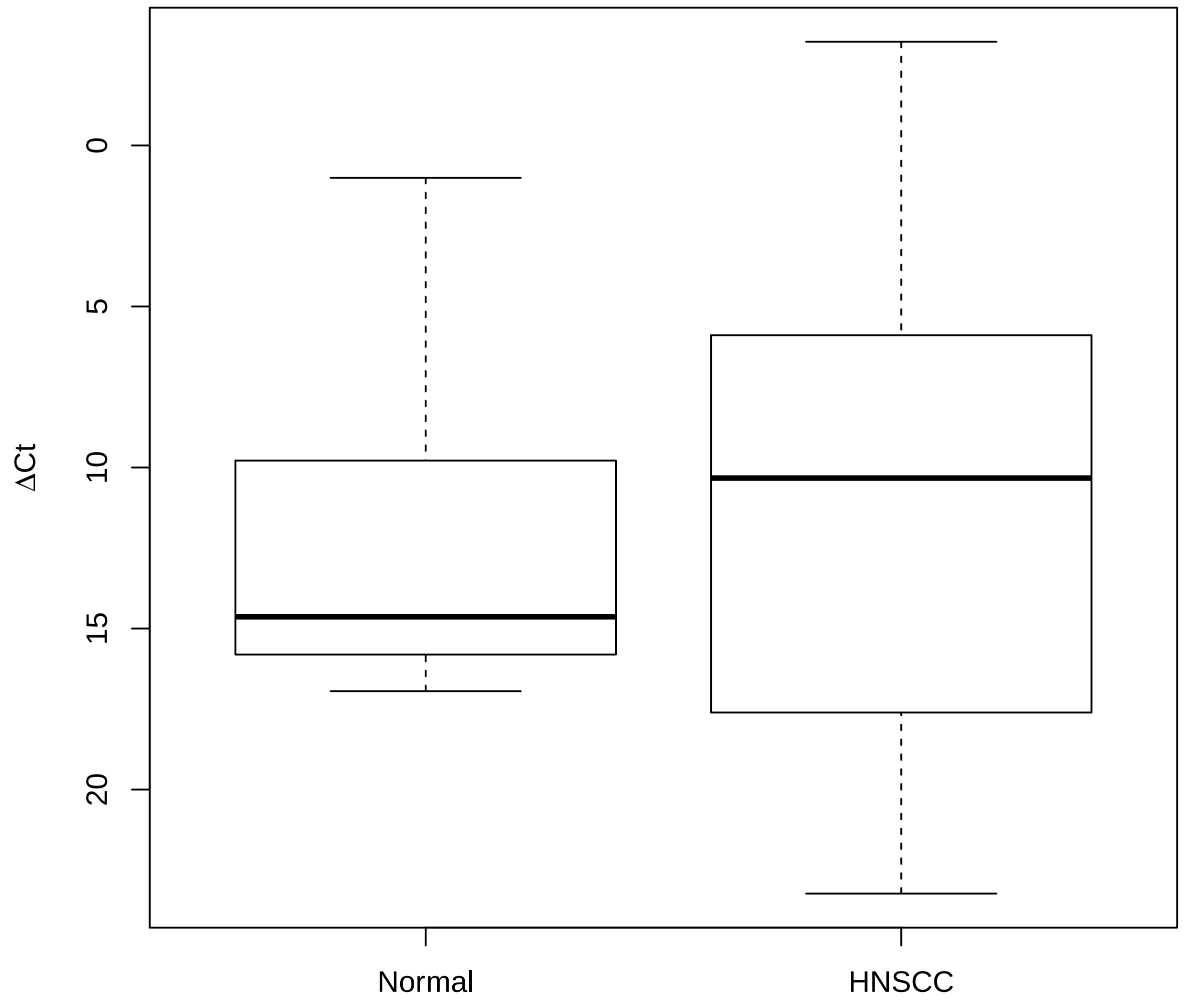

Figure 3.3: Distribution of the ΔC_t counts of miRNA differentially expressed between normal control (left) and HNSCC tumor (right) with an adjusted p-value below 0.05.

Chapter 4

Discussion

We have observed that miRNA measured with an RT-PCR array are generally upregulated in HNSCC tumors relative to normal controls. This trend extends to the C_t value of the endogenous control gene MammU6-4395470. As a result, normalization of these samples by the reference gene may falsely decrease the global upregulation of miRNA in normals. Nonetheless, after this conservative normalization, we observe that 154 miRNA are upregulated in HNSCC tumors at a Benjamini-Hotchberg adjusted p-value of 0.05, as compared to 103 significantly downregulated miRNA. However, we cannot eliminate the possibility that the observed differences in miRNA expression arise from running all normal samples on an isolated batch.

Chapter 5

Acknowledgements

M Considine, MF Ochs, L Marchioni

Bibliography

[1] Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B, 57, 289-300.

[2] Livak, KJ and Schmittgen, TD (2001). Analysis of relative gene expression data using real-time quantitative PCR and the $2^{-\Delta\Delta C_t}$ method. Methods **25**:402-408.

[3] Smyth, GK (2004) Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology 3, No. 1, Article 3.

[4] Schmittgen, TD and Livak, KJ (2008). Analyzing real-time PCR by the comparative C_t method. Nature Protocols **3**:1101-1108.

[5] Griffiths-Jones, S (2004) The microRNA registry. NAR **32**:D109-D111.

[6] Griffiths-Jones S, Grocock RJ, van Dongen S, Bateman A, Enright AJ. miRBase: microRNA sequences, targets and gene nomenclature. NAR 2006 34(Database Issue):D140-D144

[7] Kozomara A, Griffiths-Jones S. miRBase: integrating microRNA annotation and deep-sequencing data. NAR 2011 39(Database Issue):D152-D157

[8] James F. Reid. mirbase.db: miRBase: the microRNA database. R package version 0.7.0.