Supplementary Materials

Figure Legends

Figure S1. Here is a plot of the mean spectral count values of the cancerous data versus the mean spectral count values of the non-cancerous data for the QSpec statistical method, with p-values shown by shading of each data point (Bluer the point, the lower the p-value. Redder the point, higher the p-value). The left side graph is false discovery rate unadjusted while the right side graph is FDR adjusted.

Figure S2. The left side of this figure demonstrates the mean spectral count values for the quasi-Poisson method plotted against the variance, with a data point colored according to FDR adjust p-value, with redder being a higher p-value. The left side is the normal data and the right side is the cancerous data. The right side graph demonstrates the mean spectral count vales plotted against their coefficients of variance, with the left side being normal and the right side being cancerous.

Figure S3. The left side of this figure demonstrates the mean spectral count values for the negative binomial method plotted against the variance, with a data point colored according to FDR adjust p-value, with redder being a higher p-value. The left side is the normal data and the right side is the cancerous data. The right side graph demonstrates the mean spectral count vales plotted against their coefficients of variance, with the left side being normal and the right side being cancerous.

Figure S4. Comparisons of the p-values(fdr adjusted) for quasi-Poisson without the normalization and quasi-Poisson with normalization for the HNSCC dataset (neck squamous cell carcinomas and normal tonsillectomy tissues) with negative binomial p-values as the color label. The correlation coefficient is shown in the inset.

Figure S5. Comparisons of the p-values(fdr adjusted) for negative binomial without the normalization and negative binomial with normalization for the HNSCC dataset with quasi-Poisson p-values as the color label. The correlation coefficient is shown in the inset.

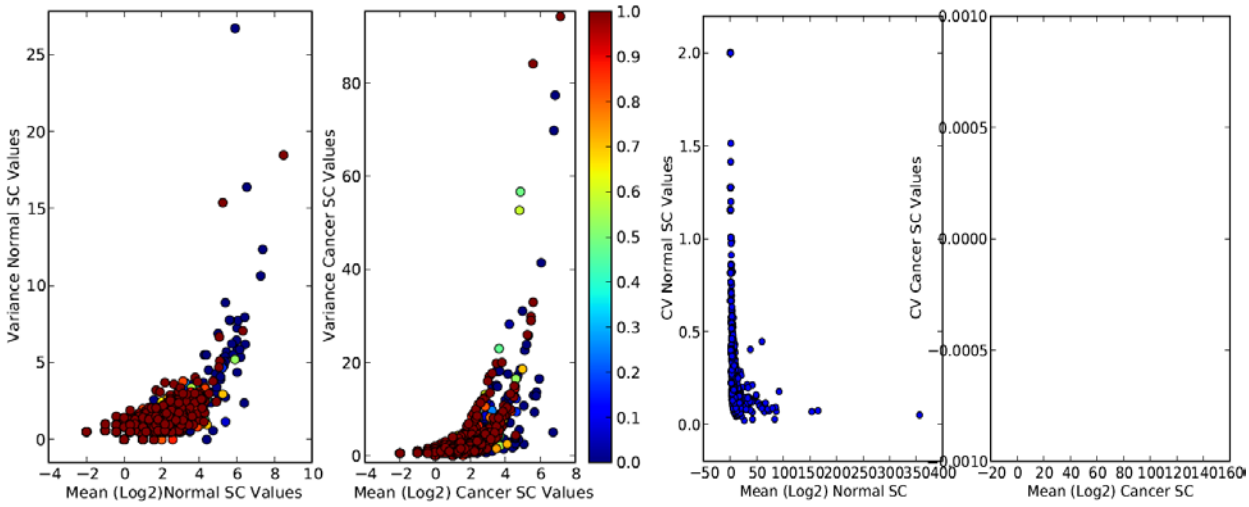
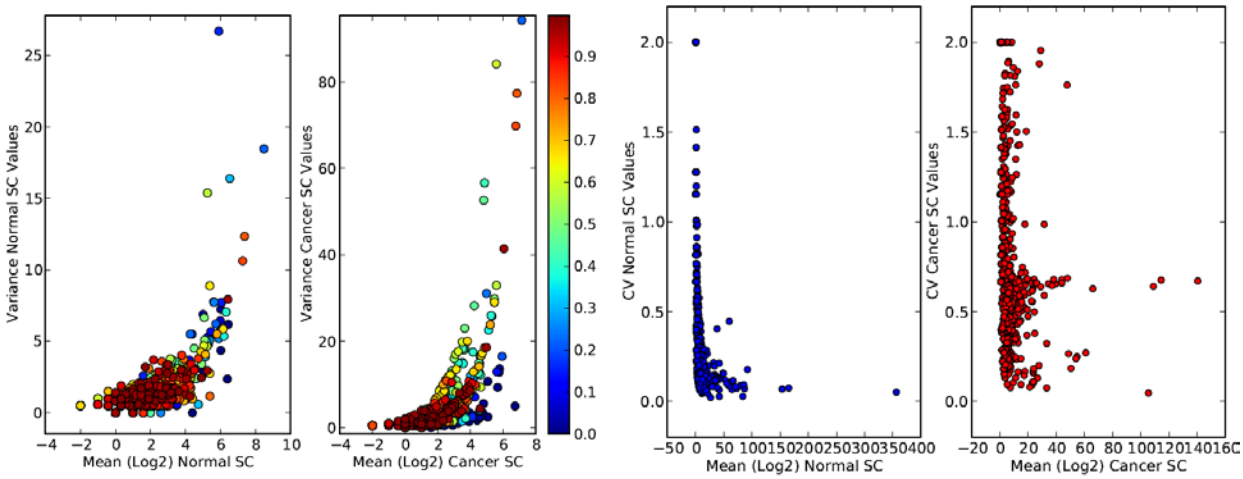
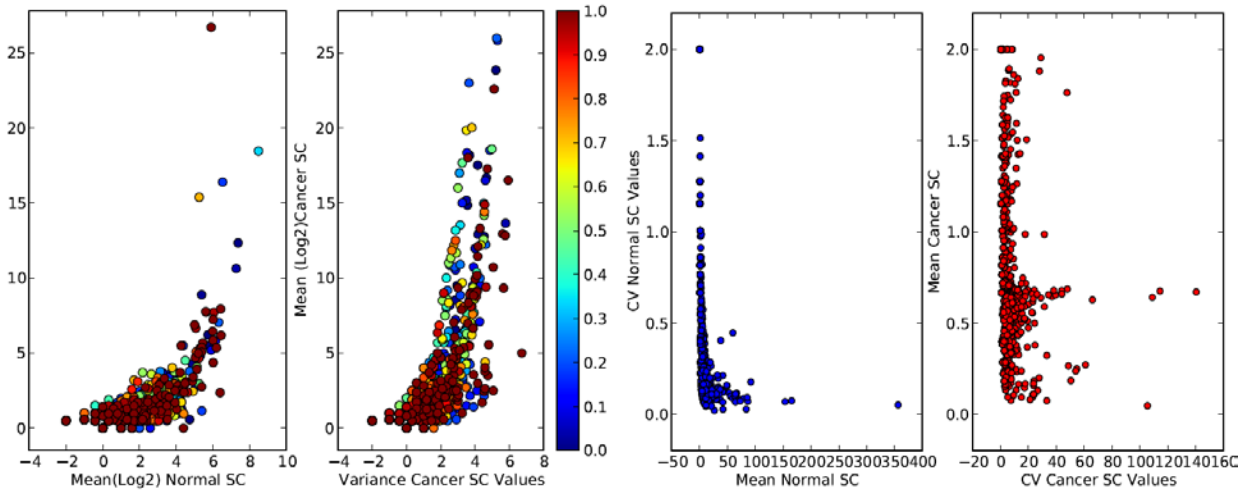

Variance Normal SC Values
Mean (Log2)Normal SC Values
Variance Cancer SC Values
Mean (Log2) Cancer SC Values
CV Normal SC Values
Mean (Log2) Normal SC
CV Cancer SC Values
Mean (Log2) Cancer SC

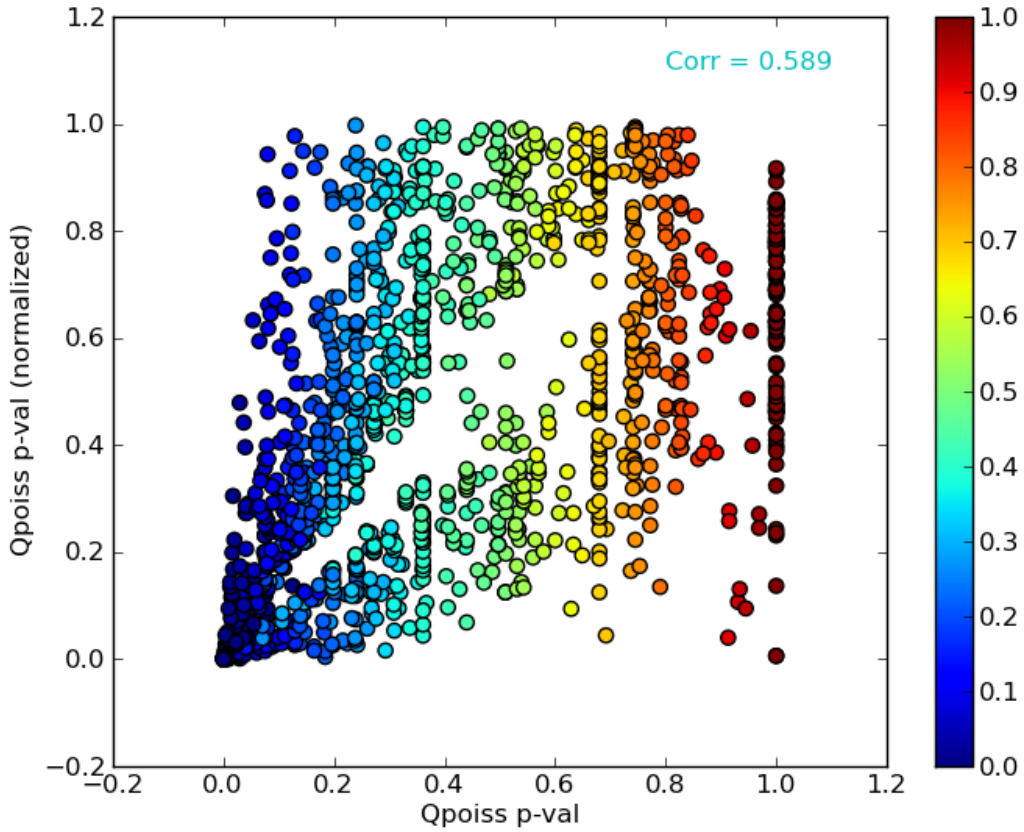
Figure S1.


Variance Normal SC Values
Mean (Log2) Normal SC
Variance Cancer SC Values
Mean (Log2) Cancer SC
CV Normal SC Values
Mean (Log2) Normal SC
CV Cancer SC Values
Mean (Log2) Cancer SC

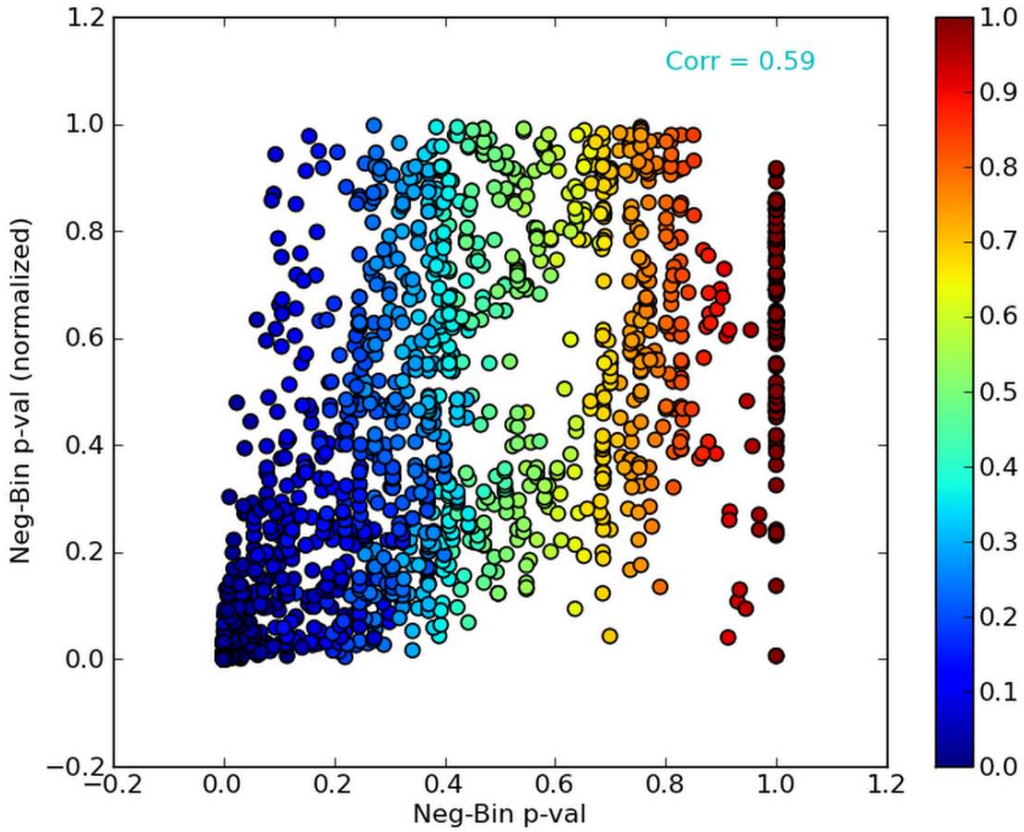
FigureS2.


Mean(Log2) Normal SC
Mean (Log2)Cancer SC
Variance Cancer SC Values
CV Normal SC Values
Mean Normal SC
Mean Cancer SC
CV Cancer SC Values

FigureS3.


Corr = 0.589
Qpoiss p-val (normalized)
Qpoiss p-val
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

FigureS4.


Corr = 0.59
Neg-Bin p-val (normalized)
Neg-Bin p-val
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

FigureS5.

Documented R Code:

```
#########################################################################

##Program in R which compute the GLMS' for quasiPoisson, Poisson , Negative
##Binomial and the corresponding p-values

###################loading the Libraries#########################

library(MASS)

library(stepNorm)

##############################The function#####################

### The function quasitelmod takes a data file of spectral count and evaluates the

### The different generalized linear models ( in this case Poisson, QuasiPoisson
and negve Binomial

### and  evaluates the corresponding p-values and fdr values for these
methods.....

#########################################################################

########the function takes the full data file as input, the datafile1

#####is the file only containg the spectral count( all along with headers)

quasitelmod <- function(data, data1, group1, group2, weight=NULL,
rm.SID=TRUE, rm.zero=FALSE, minavgcount=NULL) {

   require(stats) || stop("stats package is missing")


        ## creating the classes from the spectralcounts only  (to be used for nb)

        lib.size <- colSums(data1)

        classes <- c(1,1,1,1,2,2,2,2)

   # should pass these as arguments instead

   option.contrasts <- getOption('contrasts')

   options(contrasts=c('contr.SAS', 'contr.treatment'))


   # only keep SID==1
```

```
if (rm.SID) {
   #sids <- c("Sequence.ID", "SID")
   #sid <- sids[sids %in% colnames(data)][1]
   #data <- subset(data, with(data, eval(parse(text=paste(sid, "== 1")))))
   if ("Sequence.ID" %in% colnames(data)) {
      data <- subset(data, with(data, Sequence.ID == 1))
   } else if ("SID" %in% colnames(data)) {
      data <- subset(data, with(data, SID == 1))
   }
}
### the groups will only be either the control data or the treatment data
# make a grouping factor that will be applicable to the subsetted data
group <- character()
if (is.character(group1)) {
   group1 <- which(colnames(data) %in% group1)
}
if (is.character(group2)) {
   group2 <- which(colnames(data) %in% group2)
}
group[group1] <- "group1"
group[group2] <- "group2"
group <- group[!is.na(group)]
group <- factor(group)
gpl <- split(1:length(group), group)

# subset the data and groups
data <- subset(data, select=union(group1, group2))
```

```
group1 <- gpl$group1
group2 <- gpl$group2

# filter based on minimum count
if (!is.null(minavgcount)) {
    grp12count <- apply(data, 1, mean)
    data <- subset(data, grp12count >= minavgcount)
}

#grp1zero <- apply(subset(data, select=group1), 1, sum) == 0
#grp2zero <- apply(subset(data, select=group2), 1, sum) == 0
#if (rm.zero) {
#    # only keep features that are not both zero
#    data <- subset(data, !(grp1zero | grp2zero))
#} else {
#    # or add a single count
#    data[grp1zero, group1[1]] <- 1
#    data[grp2zero, group2[1]] <- 1
#}

# prepare the weight
if (is.null(weight)) {
    offset <- NULL
    wei <- c(length(group1), length(group2))
} else {
    weight <- weight[colnames(data)]
    offset <- log(weight)
```

```
    wei <- sapply(gpl, function(x) { sum(weight[x]) })
  }
  #### finding the number of proteins in the file
  Nprotein <- nrow(data)
      ####creating the results vector for storing the results
  result <- matrix(numeric(), nrow=Nprotein, ncol=14)

  for (i in 1:Nprotein) {
    count <- as.numeric(data[i,])

    # poisson p-value
    g1a <- glm(count ~ group, offset=offset, family=poisson)
    g1 <- glm(count ~ 1, offset=offset, family=poisson)
    anovaP <- data.frame(anova(g1, g1a, test="Chisq"))
    Pvalues <- ifelse(anovaP[2,4] < 0.1e-15, 1, anovaP[2,5])

    # quasi p-value
    gquasi1a <- glm(count ~ group, offset=offset, family=quasi(link=log, variance=mu))
    gquasi1 <- glm(count ~ 1, offset=offset, family=quasi(link=log, variance=mu))
    anovaPq <- data.frame(anova(gquasi1, gquasi1a, test="F"))
    Pvaluesq <- ifelse(anovaPq[2,4] < 0.1e-15, 1, anovaPq[2,6])

            #negb p-value
            fitneg <- glm.nb(count ~offset(log(lib.size))+classes)
            beta0 <- fitneg$coefficients[[1]]
            beta1 <- fitneg$coefficients[[2]]
            dispneg <- 1/fitneg$theta
```

```
            prop0 <- exp(beta0)
            prop1 <- exp(beta0+beta1)
            t.value <-summary(fitneg)$coefficients,"z value"][2]
            p.valueneg <- 2*pt(-abs(t.value),fitneg$df.residual)
            #negbpfdr <-p.adjust(p.valueneg, method='fdr')

            # ratios of poisson parameters for control and treatment
   lambda <- exp(rev(cumsum(as.numeric(g1a$coef))))
   totcot <- round(wei * lambda, 0)
   rateratio <- log2(lambda[1] / lambda[2])

            #evaluating the Coeffiecient of Variation for the two groups
   sdl <- sapply(gpl, function(x) { sd(count[x]) })
   meanl <- sapply(gpl, function(x) { mean(count[x]) })
   cvl <- mapply("/", sdl, meanl)
   ## the results from q-poisson, poisson and negative binomial
   result[i, ] <- c(   totcot,     # 2 items
                       lambda,     # 2 items
                       rateratio,
                       Pvalues,    NA,
                       Pvaluesq,   NA,
                                                   dispneg,
                       p.valueneg, NA,
                       cvl)        # 2 items
}

# fdr adjustment
```

```
        ## adjustment for fdr to quasipoisson p  and negv binom p values
    for (j in c(6, 8, 11)) {
       result[, j+1] <- p.adjust(result[, j], method="fdr")
    }
        ## row names are the corresponding proteins
    rownames(result) <- rownames(data)
        ## header names in the resultfile
    colnames(result) <- c(  "count1",   "count2",
                            "rates1",   "rates2",
                            "2log(rate1/rate2)",
                            "poisson.p", "poisson.fdr",
                            "quasi.p",  "quasi.fdr",
                                        "negbinomDisp",
                           "negbinomPvalue", "negbinfdr",
                            "cv1",      "cv2")
    options(contrasts=option.contrasts)
        ## return result
    result
}
```