Collate Region Counts

Input settings

```
gffFileInfo <- read.csv("un_bowtie_gffFileInfo.csv", stringsAsFactors = FALSE,
                        check.names = FALSE) #GFF file info

sampleInfo <- read.csv("un_bowtie_sampleInfo.csv", stringsAsFactors = FALSE,
                       check.names = FALSE) #Sample information

#name of input folder that contains region counts in csv format
inputFolder1 <- "output_gene_cds"

#name of output folder where the results will be saved
outputFolder1 <- "output_gene_cds_collated"

#name of output file
outputFileName1 <- "un_bowtie_region_COUNTS_ALL_Sum"

#Name of reference file from which annotation will be added
viralRef1 = read.csv(file = "Complete_Sequence_info.csv", header=T, stringsAsFactors = F)
```

Function definition - to collate results

```
funcCollateResults <- function(fileNames, sampleNames, inputFolder,
                               outputFolder, outputFileName, countType="sum") {
    #loop for every genome
    #browser()
    iCount1 <- 1
    finalM <- {}
    while(iCount1 <= length(fileNames)) {
        print(iCount1)
        oneGff <- gffFileInfo$gff.file.names[iCount1]

        if(file.exists(oneGff) == FALSE)

            #loop for every sample
            iCount2 <- 1
            tempM <- {}
            while(iCount2 <= length(sampleNames)) {
                #browser()
                #print(iCount2)
                oneSample <- sampleInfo$sample.names[iCount2]
                xx <- sub(pattern = ".gff",replacement = "",x = oneGff)
                fileName <- paste(inputFolder,"/",xx,
                                  "..",oneSample,"..counts.RData" ,sep ="" )

                if(file.exists(fileName) == FALSE) {
                    print(paste(fileName,":File does not exist",sep=""))


                } else {
```

```
                f1 <- load(fileName) #loads counts object
                temp2 <- as.data.frame(counts)
                uniq1 <- unique(row.names(temp2))
                temp1 <- temp2[uniq1,]
                row.names(temp1) <- uniq1

                #Count in.region + on.boundary
                if(countType == "sum") {
                    if(iCount2 == 1) {
                        # preserve sample names and virus names
                        x <- as.data.frame(temp1$in.region +
                                           temp1$on.boundary)
                        colnames(x) <- oneSample
                        row.names(x) <- row.names(temp1)

                        tempM <- x

                    } else {
                        x <- as.data.frame(temp1$in.region +
                                           temp1$on.boundary)
                        colnames(x) <- oneSample
                        row.names(x) <- row.names(temp1)

                        tempM <- cbind(tempM, x) #adding the "in.region" column
                    }
                    #Count in.region only
                } else if (countType == "inregion") {
                    if(iCount2 == 1) {
                        # preserve sample names and virus names
                        x <- as.data.frame(temp1$in.region)
                        colnames(x) <- oneSample
                        row.names(x) <- row.names(temp1)

                        tempM <- x

                    } else {
                        x <- as.data.frame(temp1$in.region)
                        colnames(x) <- oneSample
                        row.names(x) <- row.names(temp1)

                        tempM <- cbind(tempM, x) #adding the "in.region" column
                    }
                }
            }

        iCount2 <- iCount2 + 1
    } #end of sample loop
    #make a copy of tempM
    tempM2 <- tempM

    if(iCount1 == 1) {
        finalM <- tempM2
    } else {
```

```
            finalM <- rbind(finalM, tempM2) #adding regions for every virus together
        }


        iCount1 <- iCount1 + 1
    }#outer while loop

    f <- paste(outputFolder,"/" ,outputFileName,".csv", sep="")
    write.csv(x = finalM, file = f)
    return(finalM)
}
```

Function that adds annotation (virus name)

```
funcAddAnno <- function(results, viralRef,outputFileName) {
    #browser()
    iCount4 <- 1
    finalM <- {}
    while(iCount4 <= nrow(results)) {
        oneVirus <- row.names(results)[iCount4]
        z <- unlist(strsplit(oneVirus, "_", fixed = TRUE))
        ncId <- paste(z[1],"_",z[2],sep="")

        matchingid <- match(ncId,viralRef$Accession)
        annot.name <- as.character(unlist(viralRef[matchingid,2]))

        tempM <- {}
        tempM <- cbind(results[iCount4,])
        colnames(tempM) = colnames(finalMatrix) #appending counts
        #row.names(tempM) = row.names(finalMatrix)[iCount4] #appending region name
        tempM <- cbind(tempM, ncId, annot.name)
        finalM <- rbind(finalM, tempM)
        iCount4 <- iCount4+1
    }
    row.names(finalM) = row.names(results)
    write.csv(finalM, paste(outputFileName, "_withAnno.csv",sep=""), row.names = T)
    return(finalM)
}#end of funcAddAnno
```

Calling the function. The variable `countType` is set to `"sum"` to that it can count the `in.region + on.boundary` reads

```
#call function to collate results
finalMatrix = funcCollateResults(gffFileInfo$gff.file.names,
                  sampleInfo$sample.names,
                  inputFolder1,
                  outputFolder1,
                  outputFileName1,
                  "sum" )
#call function to add annotation
finalMatrix1 = funcAddAnno(finalMatrix, viralRef1,outputFileName1)
```