

Merge all outputs from Samtools idxstats

```
library(knitr)
library(dplyr)
```

This is an R Markdown document. Once you have aligned the unmapped (unaligned) BAM files to the viral reference genome, you will get a viral BAM file. You can then use `samtools idxstat` to find the genome level counts for each virus.

Set input parameters - to be set by the user.

```
#Save all "idxstats" files into one folder. Enter the complete path of this folder here:
pathName1 <- "../output_bowtie2/"

#Name of pattern of all these files
patternName1 <- "_un.bowtie2.idxstats.txt"

# Name of output file, where output will be saved
outputFileName1 <- "output_un_bowtie"

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

This is the function that does the merging

```
funcMergeIdxStatFiles <- function(pathNameB=NULL,
                                  patternNameB="_un.bowtie2.idxstats.txt",
                                  outputFileName="output.csv") {

  #browser()
  file_listB <- list.files(path = pathNameB,
                           pattern=patternNameB,
                           recursive = TRUE,
                           full.names = TRUE)

  #looping through every file, and putting them together as a matrix
  iCount2 <- 1
  finalMatrix <- {}
  while(iCount2 <= length(file_listB)){
    oneFile <- file_listB[iCount2]
    oneFileData <- as.data.frame(read.csv(oneFile, sep="\t", header=F))

    #extract virus name and aligned columns
    oneFileDataSmall <- dplyr::select(.data = oneFileData, 1,3)

    #sort , so that * is first row
    oneFileDataSmallSort <- dplyr::arrange(.data = oneFileDataSmall, V1 )

    x <- unlist(strsplit(oneFile, "/"))

    #get only file name - that last item
    oneFileName <- x[length(x)]
    colnames(oneFileDataSmallSort) <- c("ViralGenome",oneFileName )
```

```

    if(is.null(finalMatrix) == FALSE) {
      finalMatrix <- merge(x = finalMatrix,
                          y = oneFileDataSmallSort,
                          by.x="ViralGenome", by.y="ViralGenome")
    } else {
      #this will happen only for iCount2=1
      finalMatrix <- oneFileDataSmallSort
    }

    iCount2 <- iCount2 + 1
  }

  #remove the first row, since it is "*"
  finalMatrix1 = {}
  finalMatrix1 <- as.matrix(finalMatrix[2: nrow(finalMatrix),])

  #cleaning up column names
  cA = colnames(finalMatrix1)
  cB = gsub(x= cA, patternNameB, "", fixed = T)
  colnames(finalMatrix1) = cB

  write.csv(finalMatrix1, paste(outputFileName, ".csv", sep=""), row.names = F)
  return(finalMatrix1)
}

```

Function to add annotation

```

funcAddAnno <- function(results, viralRef, outputFileName) {
  #browser()
  iCount4 <- 1
  finalM <- {}
  while(iCount4 <= nrow(results)) {
    oneVirus <- results[iCount4, 1]
    z <- unlist(strsplit(oneVirus, "|", fixed = TRUE))
    ncId <- z[4]

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

    tempM <- {}
    tempM <- c(results[iCount4,], ncId, annot.name)

    finalM <- rbind(finalM, tempM)
    iCount4 <- iCount4+1
  }

  write.csv(finalM, paste(outputFileName, "_withAnno.csv", sep=""), row.names = F)
} #end of funcAddAnno

```

Calling functions

```
# Calling the function to merge all idx stats files
finalMatrix = funcMergeIdxStatFiles(pathNameB=pathName1,
                                     patternNameB=patternName1,
                                     outputFileName=outputFileName1)

#calling function to add annotation
funcAddAnno(finalMatrix, viralRef1,outputFileName1)
```