R

Blast2GFF parser

```
args <- commandArgs(trailingOnly = TRUE)
blastFileName <- args[1]

blast = read.table(blastFileName, header=FALSE, sep="\t",as.is=TRUE)

colnames(blast) = c("query", "subject", "identity",
                  "alignment_length", "mismatches", "gap_opens",
                  "q_start", "q_end", "s_start", "s_end", "evalue",
                  "bit_score")

seqname <- paste('Chr',blast$subject,sep="")
attribute <- paste('gene=',blast$query,sep="")
strand <- ifelse(blast$s_start < blast$s_end,"+","-")
start_end <- cbind(blast$s_start,blast$s_end)
start <- apply(start_end, 1, min)
end <- apply(start_end, 1, max)

gff <- cbind(seqname,'blast','exon',start,end,'.',strand,'.',attribute)

GFFFileName = paste(blastFileName,".gff",sep="")
write.table(gff,file=GFFFileName, sep="\t", quote=F, col.names=F, row.names=F)
```

Anscombe quartet

```
data(anscombe)
mean(anscombe[,1])
mean(anscombe$x1)
mean(anscombe$x2)
mean(anscombe$x3)
mean(anscombe$x4)
sapply(anscombe,mean)
sapply(anscombe,var)
cor(anscombe$x1,anscombe$y1)
cor(anscombe$x2,anscombe$y2)
cor(anscombe$x3,anscombe$y3)
cor(anscombe$x4,anscombe$y4)
coefficients(lm(anscombe$y1~anscombe$x1))
coefficients(lm(anscombe$y2~anscombe$x2))
coefficients(lm(anscombe$y3~anscombe$x3))
coefficients(lm(anscombe$y4~anscombe$x4))
example(anscombe)
```

Iris dataset

```
data(iris)
View(iris)
iris[1,2]
summary(iris)
plot(iris)
setosa <- iris[iris$Species == "setosa",]
versicolor <- iris[iris$Species == "versicolor",]
iris.test <- t.test(setosa$Sepal.Length, versicolor$Sepal.Length)
print(iris.test)
paste("p-value:", iris.test$p.value)
```

Bioconductor

https://www.bioconductor.org/

"Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data."

Tutorial

```
http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf
```