Arkas: Repetitive Elements Quantification In Much Less Time: PCA Plotting

Timothy J. Triche, Jr, Anthony R. Colombo, Harold Pimentel

10 April, 2016

Contents

1 Introduction

Principal component analysis are eigenvalues which solves an equation involving expression assay data, an eigenvector of the expression assay associated with an eigenvalue. The eigenvectors of the assay represent coordinates of a linear transformation of data where the variables are uncorrelated.

1.1 Loading arkasData For Preparation

The PCA plotting supports the assays of a given KallistoExperiment, but also can plot any matrix class data. The matrix of expression values is cast into a data.frame using pcaGGFrame.R. The inputs are the assay data, the first component of interest (first-sixth), the second component of interest (first-sixth), and the assay type i.e. cpm, tpm, length ,etc.
the plot is called using pcaPlot(dataFrame). For this vignette, we randomly choose two integers 1-4, and output the PCA plot of the randomly selected principal components of interest.

```
library(arkas)
library(arkasData)
jsonFile <- system.file("extdata", "NS.JSON", package="arkas")
appSession <- fetchAppSession(jsonFile) ## autofill APPSESSION in paths

## samples <- c("n1","n2","n4","s1","s2","s4") ## set by appSession
names(appSession$samples) <- appSession$samples ##
 appSession$outputPath <- system.file("extdata", "", package="arkasData")
  kexp <- mergeKallisto(appSession$samples,
                        outputPath=appSession$outputPath)
```

```
## Setting transcriptome automatically from Kallisto call string.
```

```
firstInterest<-sample.int(4,1)

    if (firstInterest==1){
    firstInterest<-"first"
    }
    if (firstInterest==2){
    firstInterest<-"second"
```

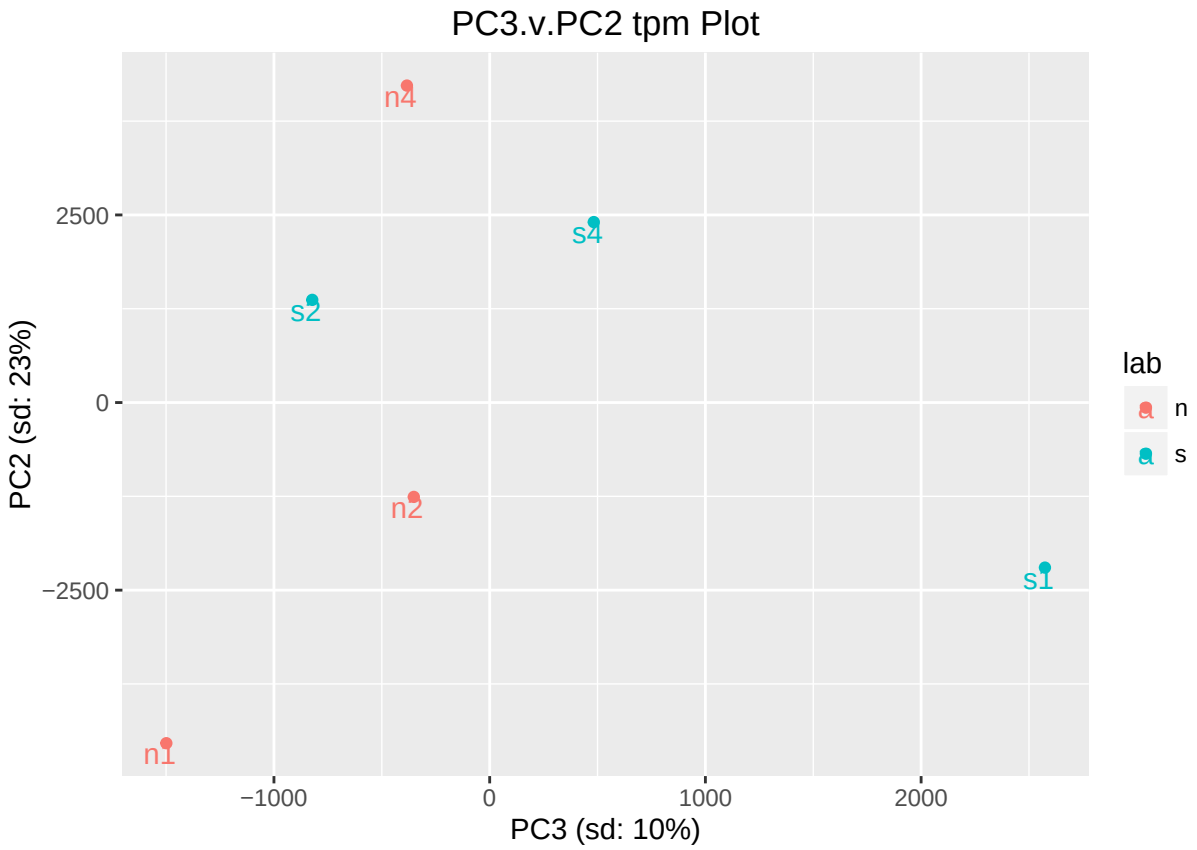
```
    }
    if (firstInterest==3){
    firstInterest<-"third"
   }
   if(firstInterest==4){
    firstInterest<-"fourth"
   }

secondInterest<-sample.int(4,1)

    if (secondInterest==1){
    secondInterest<-"first"
    }
    if (secondInterest==2){
    secondInterest<-"second"
    }
    if (secondInterest==3){
    secondInterest<-"third"
   }
   if(secondInterest==4){
   secondInterest<-"fourth"
   }



ggOut<-pcaGGFrame(assays(kexp)$tpm,
                  firstComponent=firstInterest,
                  secondComponent=secondInterest,
                  assayInterested="tpm")
pcaPlot(ggOut)
```


PC3.v.PC2 tpm Plot
n4
s4
s2
n2
s1
n1
2500
0
−2500
−1000
0
1000
2000
PC2 (sd: 23%)
PC3 (sd: 10%)
lab
n
s