

Comparing the proportion of CIs

Loading Data

```
library(tximport)
source("helper_func.R")
load("out_1/sim_counts_matrix.rda")
dir <- "out"

gsFiles <- file.path(dir, c("ERR188297_GS", "sample_01_GS", "sample_01_GS_150", "sample_01_GS_200"), "quant.sf")
txiInfRepGS <- tximport(gsFiles, type = "salmon", txOut = TRUE)

bootFiles <- file.path(dir, c("ERR188297_B", "sample_01_B"), "quant.sf")
txiInfRepBoot <- tximport(bootFiles, type = "salmon", txOut = TRUE)

txiInfRepGS <- computeConfInt(txiInfRepGS)
txiInfRepBoot <- computeConfInt(txiInfRepBoot)

depthSim <- sum(counts_matrix[,1])
depthSal <- sum(txiInfRepBoot$counts[,2])

reqSim <- counts_matrix[,1]*(depthSal/depthSim)

print(abs(sum(reqSim) - sum(txiInfRepBoot$counts[,2])) < 1e-5)
```

```
## [1] TRUE
```

```
sum(rownames(txiInfRepBoot$conf[[2]]) != rownames(txiInfRepGS$conf[[2]])) == 0 ## Checking transcripts names match across BS and GS
```

```
## [1] TRUE
```

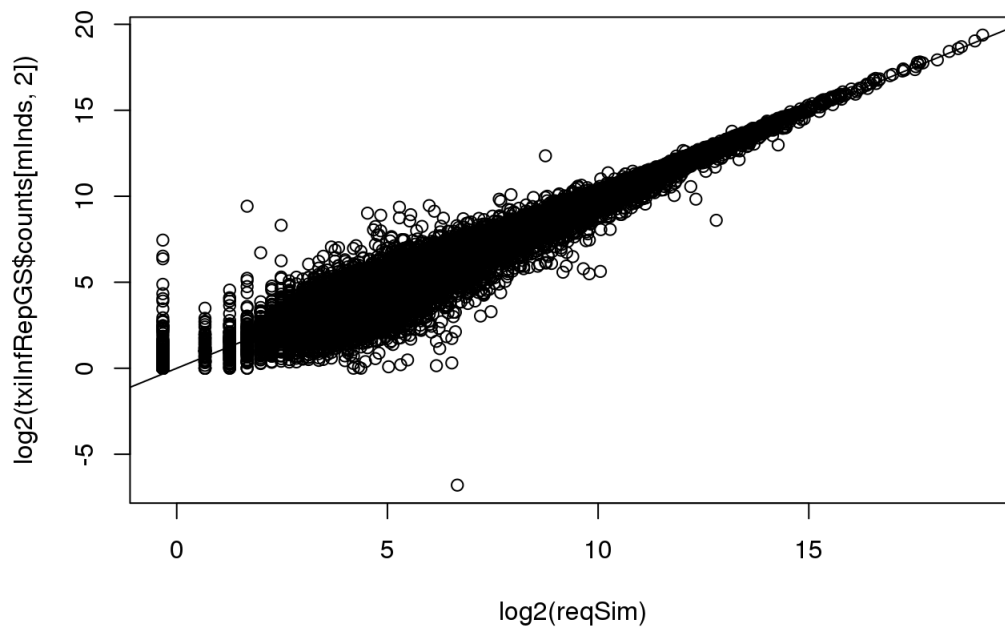
```
sum(rownames(txiInfRepGS$conf[[2]]) != rownames(txiInfRepGS$counts[,2])) == 0 ## Checking transcripts names match across the counts
```

```
## [1] TRUE
```

```
mInds <- match(rownames(counts_matrix), rownames(txiInfRepBoot$conf[[2]])) ##Indexes of transcripts of simulated data within txi
```

Plotting Simulated (True) vs Generated

```
plot(log2(reqSim), log2(txiInfRepGS$counts[mInds, 2])) + abline(coef = c(0,1))
```



```
## integer(0)
```

Computing the Coverage for the 1st sample of simulated data

```
confBootReq <- txiInfRepBoot$conf[[2]][mInds,]
confGSReq <- vector(mode = "list", 3)
names(confGSReq) <- c("GS100", "GS150", "GS200")
for(i in c(2:4))
  confGSReq[[i-1]] <- txiInfRepGS$conf[[i]][mInds,]

confMat <- list(confGSReq[[1]], confGSReq[[2]], confGSReq[[3]], confBootReq)
names(confMat) <- c("GS100", "GS150", "GS200", "BS100")

covOverall <- sapply(confMat, function(mat) computeCoverage(reqSim, mat, list(seq(nrow(counts_matrix)))))

cInds100 <- extractBinInds(reqSim, breaks = 100)
cov100 <- createCovDf(confList = confMat, counts = reqSim, cInds100)

cIndsAll <- extractBinInds(reqSim, breaks = NULL)
covAll <- createCovDf(confList = confMat, counts = reqSim, cIndsAll)
```

Overall Coverage

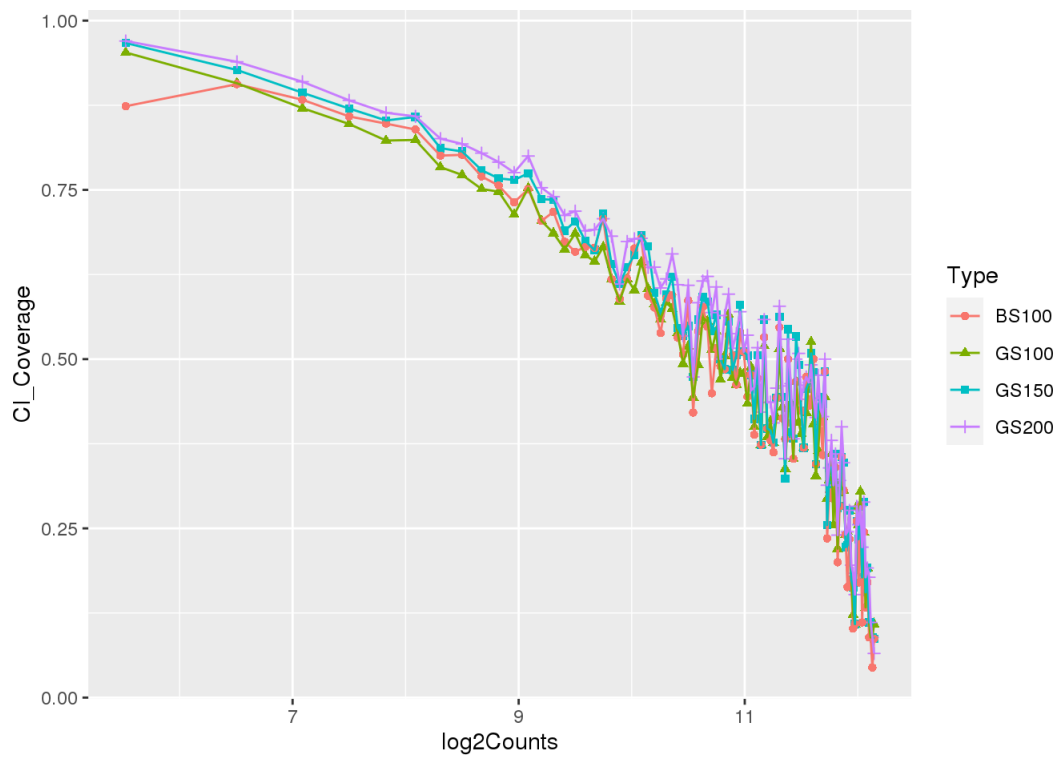
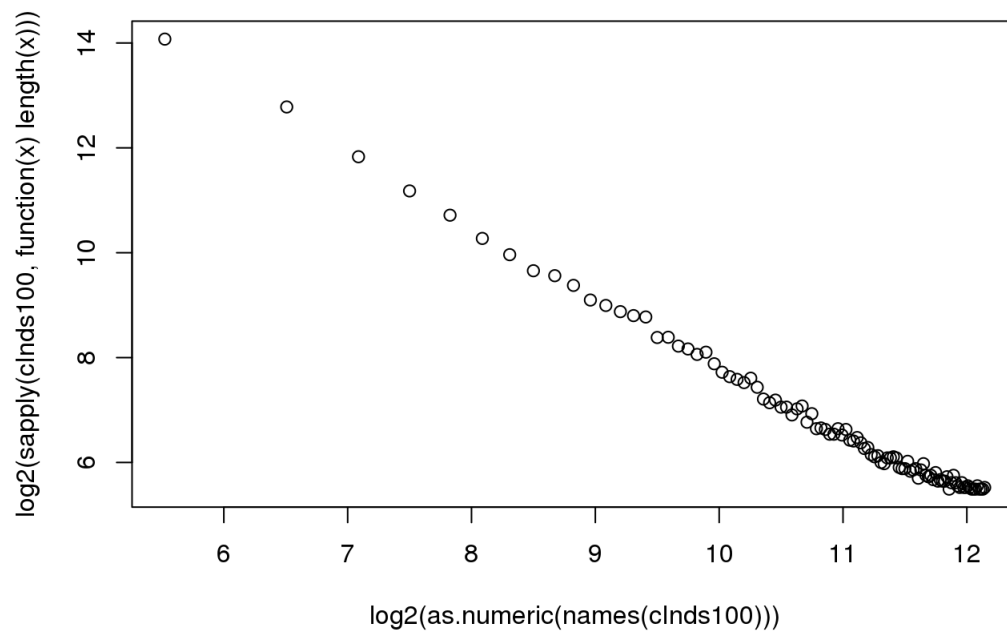
```
print(covOverall)
```

```
##      GS100      GS150      GS200      BS100
## 0.8190951 0.8402035 0.8509484 0.7953163
```

Plotting for 100 bins

First, unique counts across transcripts are computed and then partitioned into 100 bins. Accordingly transcripts arranged into bins. Plot 1 represents the number of transcripts (log) in each bin (log scale) Plot 2 shows the corresponding coverage across the bins

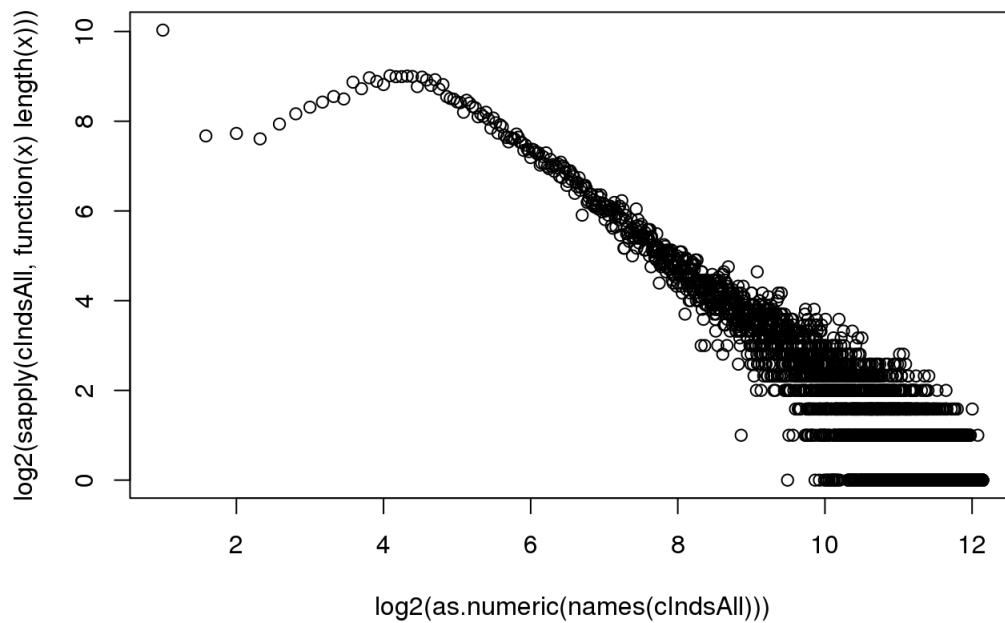
```
plot(x=log2(as.numeric(names(cInds100))), y = log2(sapply(cInds100, function(x) length(x))))
plotCovDf(cov100, line = T)
```



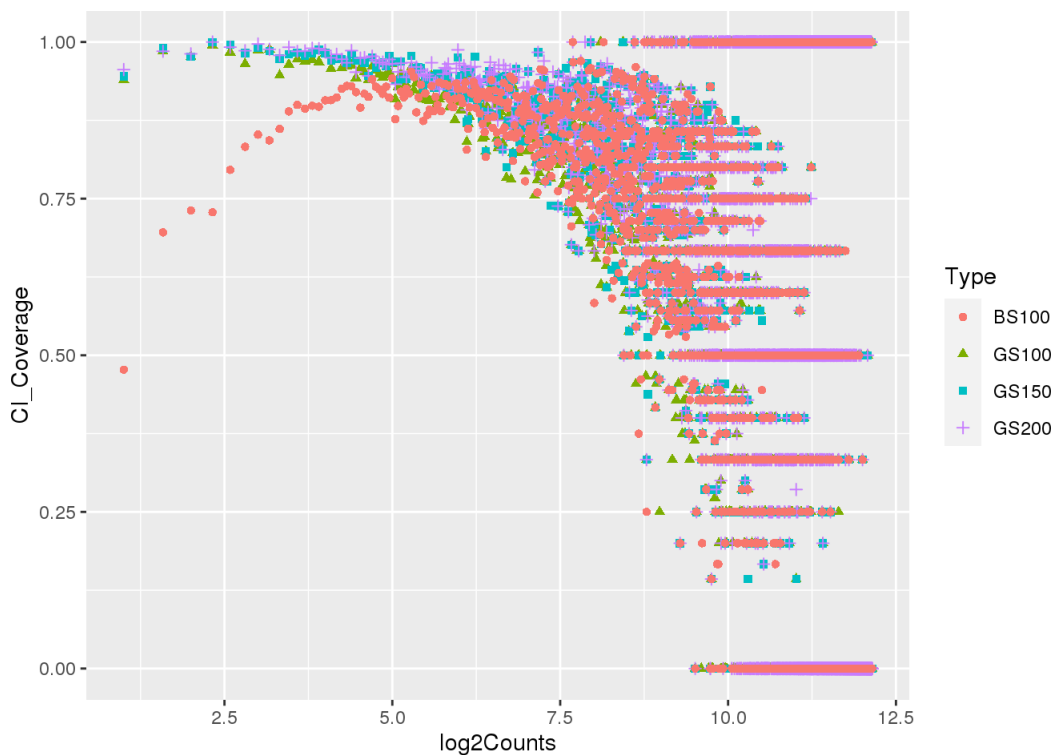
Plotting for all

Similar to above, however each bin represents a unique count

```
plot(x = log2(as.numeric(names(cIndsAll))), y = log2(supply(cIndsAll, function(x) length(x))))
```



```
plotCovDf(covAll)
```



InfReps

```
infRVs <- computeInfRep(confMat[c(1,4)])
quantGS <- quantile(infRVs[[1]], probs = c(0.10, 0.90))
quantBoot <- quantile(infRVs[[2]], probs = c(0.10, 0.90))

indsLowRV <- list("GS" = which(infRVs[[1]] <= quantGS[1]), "Boot" = which(infRVs[[2]] <= quantBoot[1]))
indsHighRV <- list("GS" = which(infRVs[[1]] >= quantGS[2]), "Boot" = which(infRVs[[2]] >= quantBoot[2]))

quantExp <- quantile(txInfRepGS$counts[mInds,2], probs = c(0.10, 0.90))
indsExp <- list("High" = which(txInfRepGS$counts[mInds,2] >= quantExp[2]), "Low" = which(txInfRepGS$counts[mInds,2] <= quantExp[1]))
```

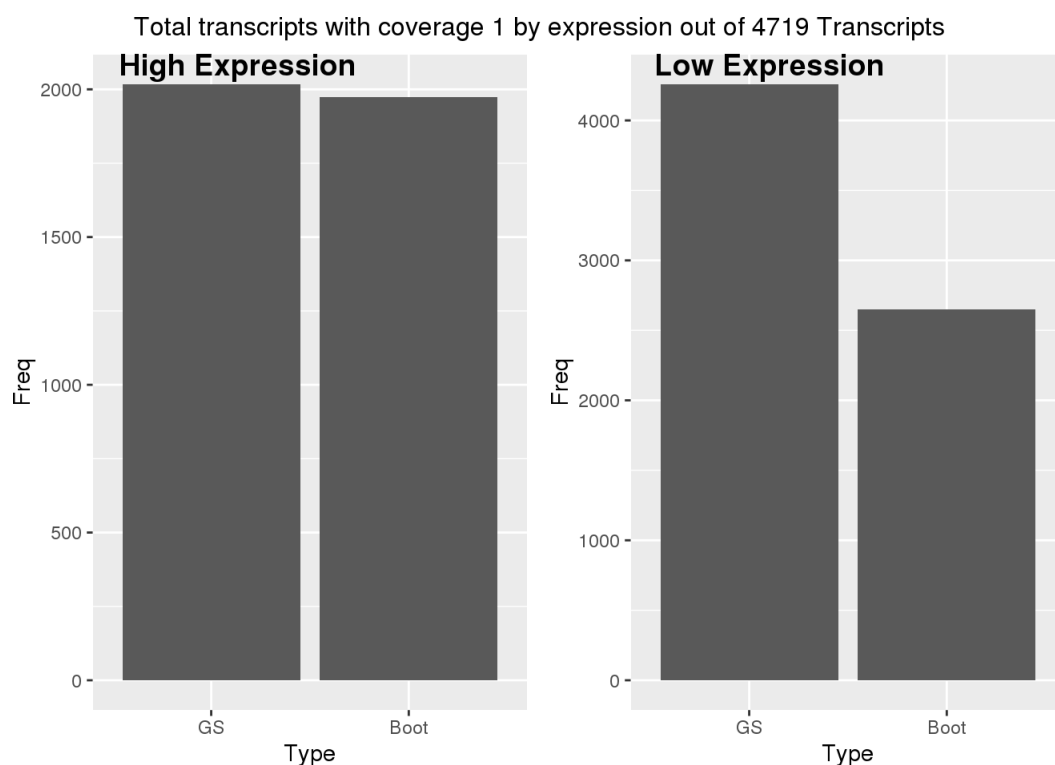
Plotting Coverage for Expression and InfReps (100 Inf Replicates)

```
library(ggpubr)
dfCov <- sapply(confMat[c(1,4)], function(mat) computeCoverage(reqSim, mat, list(seq(nrow(counts_matrix))),
F))

pHigh <- plotBarplot(list("GS"=dfCov[,1], "Boot"=dfCov[,2]), list("GS"=indsExp[["High"]], "Boot" = indsExp[["H
igh"]]))
```

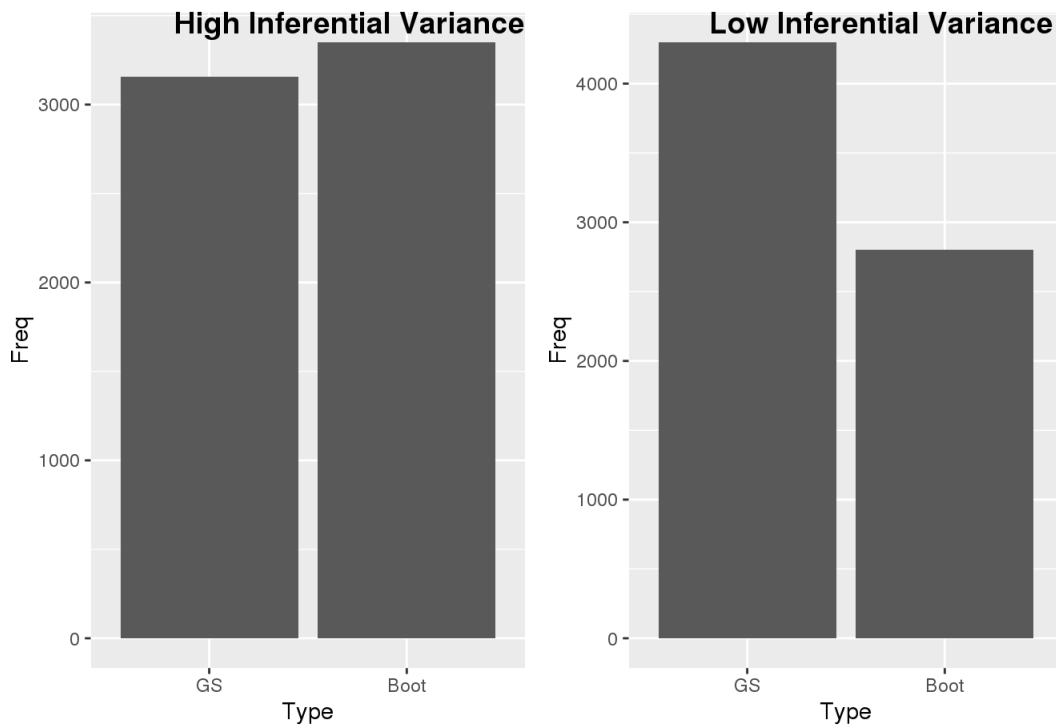
```
## data.table 1.13.0 using 1 threads (see ?getDTthreads). Latest news: r-datatable.com
```

```
pLow <- plotBarplot(list("GS"=dfCov[,1], "Boot"=dfCov[,2]), list("GS"=indsExp[["Low"]], "Boot" = indsExp[["Low
"]]))
pExp <- ggarrange(pHigh, pLow, labels = c("High Expression", "Low Expression"))
annotate_figure(pExp, top = text_grob(paste("Total transcripts with coverage 1 by expression out of", length
(indsExp[["High"]]), "Transcripts")))
```



```
pHighInf <- plotBarplot(list("GS"=dfCov[,1], "Boot"=dfCov[,2]), list("GS"=indsHighRV[["GS"]], "Boot" = indsHig
hRV[["Boot"]]))
pLowInf <- plotBarplot(list("GS"=dfCov[,1], "Boot"=dfCov[,2]), list("GS"=indsLowRV[["GS"]], "Boot" = indsLowRV
[["Boot"]]))
pInf <- ggarrange(pHighInf, pLowInf, labels = c("High Inferential Variance", "Low Inferential Variance"))
annotate_figure(pInf, top = text_grob(paste("Total transcripts with coverage 1 by Inf Variance out of", leng
th(indsLowRV[["Boot"]]), "Transcripts")))
```

Total transcripts with coverage 1 by Inf Variance out of 4719 Transcripts

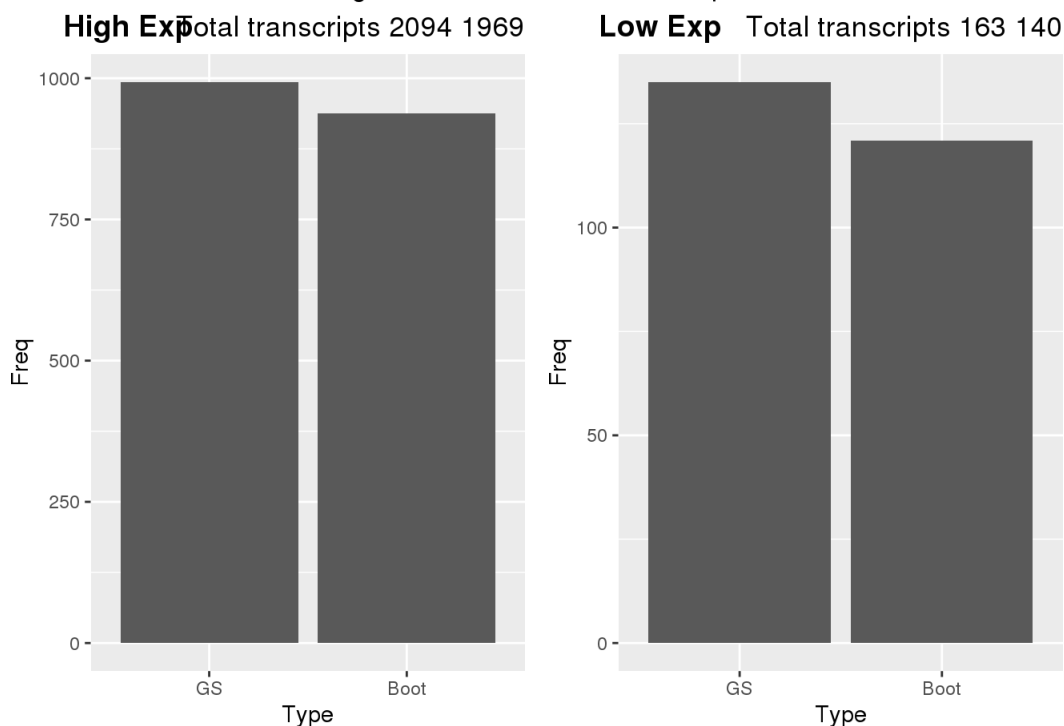


High Inf by expression

```
highInfGS <- list("HighExp" = intersect(indsHighRV[["GS"]], indsExp[["High"]]),
  "LowExp" = intersect(indsHighRV[["GS"]], indsExp[["Low"]]))
highInfBoot <- list("HighExp" = intersect(indsHighRV[["Boot"]], indsExp[["High"]]),
  "LowExp" = intersect(indsHighRV[["Boot"]], indsExp[["Low"]]))

pHighInfH <- plotBarplot(list("GS"=dfCov[,1], "Boot"=dfCov[,2]), list("GS" = highInfGS[["HighExp"]], "Boot" =
highInfBoot[["HighExp"]]), title = T)
pHighInfL <- plotBarplot(list("GS"=dfCov[,1], "Boot"=dfCov[,2]), list("GS" = highInfGS[["LowExp"]], "Boot" =
highInfBoot[["LowExp"]]), title = T)
pHighInf <- ggarrange(pHighInfH, pHighInfL, labels = c("High Exp", "Low Exp"))
annotate_figure(pHighInf, top = text_grob(paste("High Inferential Variance Transcripts")))
```

High Inferential Variance Transcripts



Low Inf by expression

```

lowInfGS <- list("HighExp" = intersect(indsLowRV[["GS"]], indsExp[["High"]]),
               "LowExp" = intersect(indsLowRV[["GS"]], indsExp[["Low"]]))
lowInfBoot <- list("HighExp" = intersect(indsLowRV[["Boot"]], indsExp[["High"]]),
                  "LowExp" = intersect(indsLowRV[["Boot"]], indsExp[["Low"]]))

pLowInfL <- plotBarplot(list("GS"=dfCov[,1], "Boot"=dfCov[,2]), list("GS" = lowInfGS[["LowExp"]], "Boot" = lowInfBoot[["LowExp"]]), title = T)
print(pLowInfL)

```

