

## coverage.Rmd

### Loading the data

```
suppressPackageStartupMessages(library(tximeta))
suppressPackageStartupMessages(library(SummarizedExperiment))
dir <- "../real_data/out"
poleeDir <- "../real_data/polee"
load("../real_data/out_1/sim_counts_matrix.rda")
source("helper_func.R")

# samp <- "sample_01"
# nInf <- c(10, 100, 1000, 2000)
# infTypes <- c("B", "GS", "polee")
counts_matrix <- read.table("../real_data/out_1/sample_01_counts.csv", header=T, row.names=1)
# seList <- vector(mode = "list", length(nInf))
# names(seList) <- as.character(nInf)
#
# for(n in nInf)
# {
#   files <- file.path(dir, paste(samp, infTypes[1:2], n, sep = "_"), "quant.sf")
#   coldata <- data.frame(files = files, names = paste(samp, infTypes[1:2], n, sep = "_"), infType = "quant")
#   se <- tximeta(coldata)
#   mInds <- match(rownames(counts_matrix), rownames(se))
#   seList[[as.character(n)]] <- se[mInds,]
#
#   ### Integrating polee
#   poleeFiles <- file.path(poleeDir, samp, paste("boot", n, sep = "_"), "abundance.h5")
#   coldata <- data.frame(files = poleeFiles, names = paste(samp, "polee", n, sep = "_"), infType = "polee")
#   se <- tximeta(coldata, type = "kallisto", txOut = T)
#   mInds <- match(rownames(counts_matrix), rownames(se))
#   se <- se[mInds,]
#   mInds <- match(rownames(se), rownames(seList[[as.character(n)]]))
#   se <- se[mInds,]
#   rowData(se) <- rowData(seList[[as.character(n)]]))
#   rowRanges(se) <- rowRanges(seList[[as.character(n)]]))
#   seList[[as.character(n)]] <- cbind(seList[[as.character(n)]], se)
# }
#
# r <- length(nInf)*length(infTypes)
# scaleTypes <- c("DESeq2", "medScale", "depthScale")
#
# dfCoverage <- data.frame(NoScale = rep(0, r), DeScale = rep(0, r), MedScale = rep(0, r), depthScale = rep(0, r))
# rownames(dfCoverage) <- as.vector(t(outer(nInf, infTypes, paste, sep = "_")))
#
# dfLogCoverage <- data.frame(NoScale = rep(0, r), DeScale = rep(0, r), MedScale = rep(0, r), depthScale = rep(0, r))
# rownames(dfLogCoverage) <- as.vector(t(outer(nInf, infTypes, paste, sep = "_")))
```

Below we compute coverage using different conditions ### Without scaling factors

```
# start = 1
# for(i in seq_along(seList))
# {
#   end = (start + length(infTypes)-1)
#   seList[[i]] <- computeSizeFactors(seList[[i]], counts_matrix[,1])
#   seList[[i]] <- computeConfInt(seList[[i]], sf = F)
#   dfCoverage[start:end,1] <- t(computeCoverage(counts_matrix[,1], seList[[i]], list(seq(nrow(counts_matrix[,1]), 1))))
#   for(j in seq_along(scaleTypes))
#   {
#     seList[[i]] <- computeConfInt(seList[[i]], sf = T, type = scaleTypes[j])
#     dfCoverage[start:end,j+1] <- t(computeCoverage(counts_matrix[,1], seList[[i]], list(seq(nrow(counts_matrix[,1]), 1))))
#   }
#   print(c(start, end))
#   start = end + 1
# }
# save(seList, file = "environment/seList.RData")
# save(dfCoverage, file = "environment/dfCoverage.RData")
load("environment/seList.RData")
load("environment/dfCoverage.RData")
print(dfCoverage)
```

##	NoScale	DeScale	MedScale	depthScale
## 10_B	0.8426070	0.8426921	0.8356278	0.8425006
## 10_GS	0.8277124	0.8332447	0.8333085	0.8293083
## 10_polee	0.7409409	0.7448348	0.7430474	0.7442603
## 100_B	0.9278677	0.9277613	0.9248888	0.9278465
## 100_GS	0.9455072	0.9470392	0.9474009	0.9461668
## 100_polee	0.8470115	0.8507564	0.8509905	0.8511607
## 1000_B	0.9347405	0.9347192	0.9321233	0.9347618
## 1000_GS	0.9586357	0.9598698	0.9604230	0.9592953
## 1000_polee	0.8586506	0.8618848	0.8622678	0.8625231
## 2000_B	0.9351661	0.9350384	0.9327617	0.9352937
## 2000_GS	0.9595293	0.9611465	0.9612316	0.9604443
## 2000_polee	0.8591399	0.8628210	0.8625019	0.8627359

## Using logs

```
# start = 1
# for(i in seq_along(seList))
# {
#   end = (start + length(infTypes)-1)
#   seList[[i]] <- computeConfInt(seList[[i]], sf = F, log = T)
#   dfLogCoverage[start:end,1] <- t(computeCoverage(counts_matrix[,1], seList[[i]], list(seq(nrow(counts_matrix[,1]), 1))))
#   for(j in seq_along(scaleTypes))
#   {
#     seList[[i]] <- computeConfInt(seList[[i]], sf = T, log = T, type = scaleTypes[j])
#     dfLogCoverage[start:end,j+1] <- t(computeCoverage(counts_matrix[,1], seList[[i]], list(seq(nrow(counts_matrix[,1]), 1))))
#   }
#   print(c(start, end))
#   start = end + 1
# }
# save(dfLogCoverage, file = "environment/dfLogCoverage.RData")
load("environment/dfLogCoverage.RData")
```

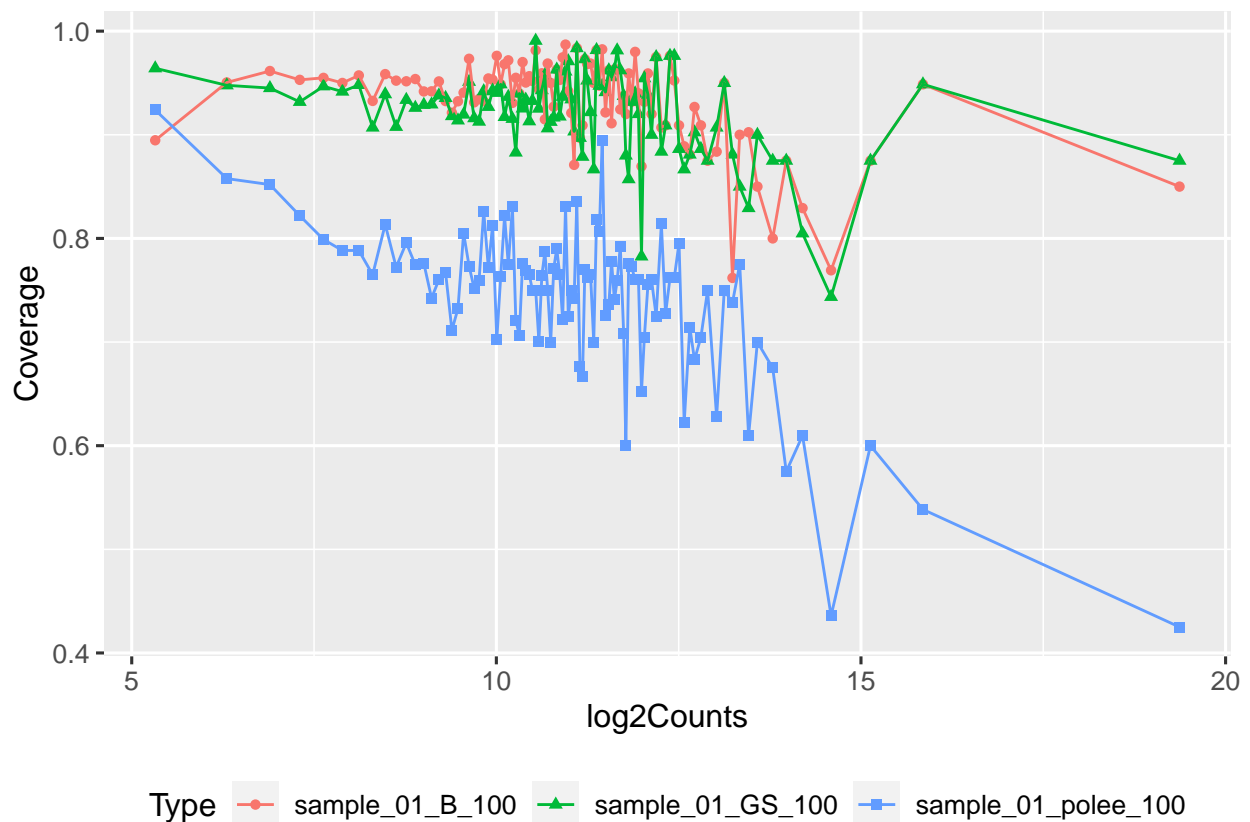
```
print(dfLogCoverage)
```

```
##           NoScale  DeScale  MedScale  depthScale
## 10_B      0.8393727 0.8393727 0.8324787 0.8392451
## 10_GS     0.8281805 0.8338617 0.8337128 0.8297338
## 10_polee  0.7409835 0.7448561 0.7433240 0.7442815
## 100_B     0.9273783 0.9272720 0.9243994 0.9273358
## 100_GS    0.9455710 0.9471243 0.9474222 0.9462519
## 100_polee 0.8468838 0.8508415 0.8510118 0.8510756
## 1000_B    0.9347405 0.9347192 0.9321233 0.9347618
## 1000_GS   0.9586357 0.9598698 0.9604230 0.9592953
## 1000_polee 0.8586506 0.8618848 0.8622678 0.8625231
## 2000_B    0.9351661 0.9350384 0.9327617 0.9352937
## 2000_GS   0.9595293 0.9611465 0.9612316 0.9604443
## 2000_polee 0.8591399 0.8628210 0.8625019 0.8627359
```

```
seList[["100"]] <- computeConfInt(seList[["100"]], sf = T)
cInds100 <- extractBinInds(counts_matrix[,1], breaks = 100)
covDf <- createCovDf(seList[["100"]], counts_matrix[,1], cInds100)
```

```
##
## Attaching package: 'reshape'
## The following objects are masked from 'package:S4Vectors':
##
##   expand, rename
```

```
p <- plotCovDf(covDf, line=T)
print(p)
```



```
ggsave(filename = "../presentations/Oct7/coverage.png", p)
```

```
## Saving 6.5 x 4.5 in image
```