```
> # 2. Preparations
> source("http://bioconductor.org/biocLite.R")
> source("groHMM")
> # 3 groHMM Workflow
> # 3.1 Read GROseq data files
> library(groHMM)
> loadData <- function(file) {
+     df <- read.table(file, header=TRUE)
+     return( GRanges(seqnames = Rle(df$seqnames), ranges = IRanges(df$start, c
+             strand = Rle(strand(df$strand))))
+ }
> S0mR1 <- loadData(system.file("extdata", "S0mR1.txt", package="groHMM"))
> S0mR2 <- loadData(system.file("extdata", "S0mR2.txt", package="groHMM"))
> S10mR1 <- loadData(system.file("extdata", "S10mR1.txt", package="groHMM"))
> S10mR2 <- loadData(system.file("extdata", "S10mR2.txt", package="groHMM"))
> S40mR1 <- loadData(system.file("extdata", "S40mR1.txt", package="groHMM"))
> S40mR2 <- loadData(system.file("extdata", "S40mR2.txt", package="groHMM"))
> S160mR1 <- loadData(system.file("extdata", "S160mR1.txt", package="groHMM"))
> S160mR2 <- loadData(system.file("extdata", "S160mR2.txt", package="groHMM"))
> # Combine replicates
> S0m <- c(S0mR1, S0mR2)
> S10m <- c(S10mR1, S10mR2)
> S40m <- c(S40mR1, S40mR2)
> S160m <- c(S160mR1, S160mR2)
> # Write wig files
> writeWiggle(pgr=S0m, file="S0m_Plus", strand="+", size=25, reverse=FALSE)
> writeWiggle(pgr=S0m, file="S0m_Minus", strand="-", size=25, reverse=TRUE)
> expCounts <- mean(c(NROW(S0m), NROW(S10m), NROW(S40m), NROW(S160m)))
> # normalization
> writeWiggle(pgr=S0m, file="S0m_Plus_Norm", strand="+", size=25, normCounts=ex
> writeWiggle(pgr=S0m, file="S0m_Minus_Norm", strand="-", size=25, normCounts=e
> # wig to bigWig
> # library("rtracklayer")
> # seqlengths(S0m) <- 159138663
> # complete chrom info by using GenomicFeatures
> # wigToBigWig(x="S0m_Plus_Norm.wig", seqinfo(S0m))
> # https://cgwb.nci.nih.gov/goldenPath/help/bigWig.html
> # fetchChromSizes
>
> #3. De novo Transcript Calling
> Sall <- c(S0mR1, S0mR2, S10mR1, S10mR2, S40mR1, S40mR2, S160mR1, S160mR2)
> hmmResult <- detectTranscriptsEM(Sall, LtProbB=-200, UTS=5, thresh=1)
> txHMM <- hmmResult$transcripts
> # Evaluation of Transcript Calling
> library(GenomicFeatures)
> rgdb <- makeTranscriptDbFromUCSC(genome="hg19", tablename="refGene")
> saveDb(rgdb, file="hg19RefGene.sqlite")
> rgdb <- loadDb("hg19RefGene.sqlite")
> rgChr7 <- transcripts(rgdb, vals <- list(tx_chrom = "chr7"), columns=c("gene_
> seqlevels(rgChr7) <- "chr7"
```

```
> rgReducedSimple <- reduce(rgChr7)  # simple approach
> rgReduced <- makeConsensusAnnotations(rgChr7, keytype="gene_id")
> library(org.Hs.eg.db)
> map <- select(org.Hs.eg.db, keys=as.integer(unlist(elementMetadata(rgReduced)
> elementMetadata(rgReduced)$symbol <- map$SYMBOL
> # save this for future reference
> #a <- getChromInfoFromUCSC("hg19")
> #library(org.Hs.eg.db)
> #c = c("ENTREZID","SYMBOL", "CHR")
> #kt = c("SYMBOL")
> #k = head(Rkeys(org.Hs.egSYMBOL))
> #select(org.Hs.eg.db, keys=as.integer(unlist(elementMetadata(ref)$gene_id)),
> #ss <- select(org.Hs.eg.db, keys=as.character(unlist(elementMetadata(ref)$tx_
> #select(org.Hs.eg.db, keys=c(1132,1133), cols=c, keytype=c("REFSEQ"))
> # select(org.Hs.eg.db, keys=as.integer(unlist(elementMetadata(rgReduced)$gene
> e <- evaluateHMM(txHMM, rgReduced)  # Run genes together: 80, Broken up a sin
> # Break and Combine Transcripts
> bTRPlus <- breakTranscriptsOnGenes(txHMM, rgReduced, strand="+", debug=FALSE)
> bTRMinus <- breakTranscriptsOnGenes(txHMM, rgReduced, strand="-", debug=FALSE
> bTR <- c(bTRPlus, bTRMinus)
> TR <- combineTranscripts(bTR, rgReduced, debug=FALSE)
> # Get nonoverlapping genes with expression
> getExpressedAnnotation <- function(fgr, pgr) {
+     grLimit <- limitToXkb(fgr)
+     grCount <- countReadsInInterval(fgr=grLimit, pgr=pgr)
+
+     fgr <- fgr[grCount!=0,]
+     return(fgr[(quantile(width(fgr), .05)< width(fgr)) & (width(fgr) < quanti
+ }
> rgExpressed <- getExpressedAnnotation(fgr=rgReduced, pgr=Sall)
> print(paste("length(rg):", length(rgChr7)))
> print(paste("length(rgReduced):", length(rgReduced)))
> print(paste("length(rgExpressed):", length(rgExpressed)))
> plotTHistogram(TR, rgExpressed,  scale=TRUE, runGenes="best", brokenAnnotatio
> #-------------------------------------------------------------------------
> # Differential Analysis
> #-------------------------------------------------------------------------
> #1. transcripts
> library(edgeR)
> TRlimit <- limitToXkb(TR)
> countS0mR1 <- countReadsInInterval(fgr=TRlimit, pgr=S0mR1)
> countS0mR2 <- countReadsInInterval(fgr=TRlimit, pgr=S0mR2)
> countS40mR1 <- countReadsInInterval(fgr=TRlimit, pgr=S40mR1)
> countS40mR2 <- countReadsInInterval(fgr=TRlimit, pgr=S40mR2)
> counts <- as.matrix(data.frame(countS0mR1, countS0mR2, countS40mR1, countS40m
> group <- factor(c("S0m", "S0m", "S40m", "S40m"))
> lib.size <- c(NROW(S0mR1), NROW(S0mR2), NROW(S40mR1), NROW(S40mR2))
> d <- DGEList(counts=counts, lib.size=lib.size, group=group)
> #d <- calcNormFactors(d)
> d <- estimateCommonDisp(d)
```

```
> #d <- estimateTagwiseDisp(d)
> et <- exactTest(d)
> #topTags(et)
>
> de <- decideTestsDGE(et, p=0.001, adjust="fdr")
> detags <- (1:NROW(d))[as.logical(de)]
> #summary(de)
>
> plotSmear(et, de.tags=detags)
> abline(h = c(-2,2), col="blue")
> print("Number of Transcripts regulated at S40")
> print(paste("up:",sum(de==1)))
> print(paste("down:",sum(de==-1)))
> #2. genes
> rgChr7 <- transcripts(rgdb, vals <- list(tx_chrom = "chr7"), columns=c("gene_
> map <- select(org.Hs.eg.db, keys=as.integer(unique(unlist(elementMetadata(rgC
> inx <- match(unlist(elementMetadata(rgChr7)$gene_id), map$ENTREZID)
> elementMetadata(rgChr7)$symbol <- map[inx,"SYMBOL"]
> rgLimit <- limitToXkb(rgChr7)
> countS0mR1 <- countReadsInInterval(fgr=rgLimit, pgr=S0mR1)
> countS0mR2 <- countReadsInInterval(fgr=rgLimit, pgr=S0mR2)
> countS40mR1 <- countReadsInInterval(fgr=rgLimit, pgr=S40mR1)
> countS40mR2 <- countReadsInInterval(fgr=rgLimit, pgr=S40mR2)
> counts <- as.matrix(data.frame(countS0mR1, countS0mR2, countS40mR1, countS40m
> group <- factor(c("S0m", "S0m", "S40m", "S40m"))
> lib.size <- c(NROW(S0mR1), NROW(S0mR2), NROW(S40mR1), NROW(S40mR2))
> d <- DGEList(counts=counts, lib.size=lib.size, group=group)
> #d <- calcNormFactors(d)
> d <- estimateCommonDisp(d)
> #d <- estimateTagwiseDisp(d)
> et <- exactTest(d)
> #topTags(et)
>
> de <- decideTestsDGE(et, p=0.001, adjust="fdr")
> detags <- (1:NROW(d))[as.logical(de)]
> #summary(de)
>
> plotSmear(et, de.tags=detags)
> abline(h = c(-2,2), col="blue")
> print("Number of genes regulated at S40")
> print(paste("up:",sum(de==1)))
> print(paste("down:",sum(de==-1)))
> symbols <- elementMetadata(rgChr7)$symbol
> print("Number of Unique Genes at S40")
> print(paste("up:", NROW(unique(unique(symbols[de==1])))))
> print(paste("down:", NROW(unique(unique(symbols[de==-1])))))
> #-----------------------------------------------------------------------------
> # Metagene Analysis
> #-----------------------------------------------------------------------------
> upGenes <- rgChr7[de==1,]
```

```
> TSS <- resize(upGenes, width=1)
> expReads <- mean(c(NROW(S0m), NROW(S10m), NROW(S40m), NROW(S160m)))
> count0m <- runMetaGene(fgr=TSS, pgr=S0m, size=100, normCounts=expReads, sampl
> count40m <- runMetaGene(fgr=TSS, pgr=S40m, size=100, normCounts=expReads, sam
> plotMetaGene <- function(POS=c(-10000:+10000), counts, MIN, MAX){
+     plot(POS, counts$sense, col="red", type="h", xlim=c(-5000, 5000),
+         ylim=c(floor(MIN),ceiling(MAX)), ylab="Read Density",
+         xlab="Position (relative to TSS)")
+     points(POS, (-1*rev(counts$antisense)), col="blue", type="h")
+     abline(mean(counts$sense[5000:8000]), 0, lty="dotted")
+ }
> MAX <- max(c(count0m$sense, count40m$sense))
> MIN <- -1*max(c(count0m$antisense, count40m$antisense))
> #par(mfrow=c(1, 2), mar=c(2, 2, 2, 0) + 0.1)
> plotMetaGene(counts=count0m, MIN=MIN, MAX=MAX)
> plotMetaGene(counts=count40m, MIN=MIN, MAX=MAX)
> save.image(file="progress.RData")
> load("progress.RData")
> sessionInfo()
> #R version 2.15.2 (2012-10-26)
> #Platform: x86_64-unknown-linux-gnu (64-bit)
> #
> #locale:
> # [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
> # [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
> # [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
> # [7] LC_PAPER=C                 LC_NAME=C
> # [9] LC_ADDRESS=C               LC_TELEPHONE=C
> #[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
> #
> #attached base packages:
> #[1] stats     graphics  grDevices utils     datasets  methods   base
>
> #other attached packages:
> # [1] edgeR_3.0.8            limma_3.14.4           org.Hs.eg.db_2.8.0
> # [4] RSQLite_0.11.2         DBI_0.2-5              GenomicFeatures_1.10.1
> # [7] AnnotationDbi_1.20.3   Biobase_2.18.0         groHMM_0.99.0
> #[10] GenomicRanges_1.10.6   IRanges_1.16.4         BiocGenerics_0.4.0
> #[13] MASS_7.3-23
>
> #loaded via a namespace (and not attached):
> # [1] biomaRt_2.14.0     Biostrings_2.26.3  bitops_1.0-5       BSgenome_1.26.
> # [5] parallel_2.15.2    RCurl_1.95-3       Rsamtools_1.10.2   rtracklayer_1.
> # [9] stats4_2.15.2      tools_2.15.2       XML_3.95-0.1       zlibbioc_1.4.0
```