ATAC-seq Analysis Pipeline for GSE85330 Dataset

Dr. Nazanin Bagherlou

2025-07-27

Introduction

In this report, I demonstrate my ability to perform a comprehensive ATAC-seq data analysis using the publicly available GSE85330 dataset. This dataset includes ATAC-seq profiles of human induced pluripotent stem cells (hiPSCs) at two critical timepoints:

Day 0 (undifferentiated state)

Day 30 (differentiated cardiomyocytes)

The main objective of this analysis is to identify and interpret changes in chromatin accessibility between these two conditions, which reflect the epigenomic dynamics associated with cardiac lineage differentiation.

To achieve this, I implemented a complete and reproducible pipeline using R and Bioconductor, including:

- Preprocessing and reading raw BED files from GEO
- Converting peaks to GRanges objects
- Peak annotation with known gene features
- Functional enrichment analysis (GO)
- Consensus peak matrix generation
- Differential accessibility analysis using DESeq2
- Visualization of genomic distributions and statistical results

This project serves as an example of my proficiency in epigenomic data analysis, reproducible coding, and interpretation of chromatin accessibility landscapes using real-world biological datasets. All steps are fully automated and documented to facilitate reproducibility and reuse.

---

1. Install and Load Required Packages

```
cran_packages <- c("ggplot2", "enrichplot")
bioc_packages <- c(
  "GenomicRanges", "IRanges", "ChIPseeker", "TxDb.Hsapiens.UCSC.hg38.knownGene",
  "org.Hs.eg.db", "clusterProfiler", "DESeq2"
```

```
)

install_if_missing_cran <- function(pkg) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg, dependencies = TRUE)
  }
}

install_if_missing_bioc <- function(pkg) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    if (!requireNamespace("BiocManager", quietly = TRUE))
      install.packages("BiocManager")
    BiocManager::install(pkg, ask = FALSE, update = FALSE)
  }
}

invisible(lapply(cran_packages, install_if_missing_cran))
```

```
##
```

```
invisible(lapply(bioc_packages, install_if_missing_bioc))
```

```
##
```

```
suppressPackageStartupMessages({
  lapply(c(cran_packages, bioc_packages), library, character.only = TRUE)
})
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'matrixStats' was built under R version 4.4.3
```

```
## [[1]]
## [1] "ggplot2"   "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[2]]
## [1] "enrichplot" "ggplot2"    "stats"      "graphics"   "grDevices"
## [6] "utils"      "datasets"   "methods"    "base"
##
## [[3]]
##  [1] "GenomicRanges" "GenomeInfoDb"  "IRanges"       "S4Vectors"
##  [5] "BiocGenerics"  "stats4"        "enrichplot"    "ggplot2"
##  [9] "stats"         "graphics"      "grDevices"     "utils"
## [13] "datasets"      "methods"       "base"
##
## [[4]]
##  [1] "GenomicRanges" "GenomeInfoDb"  "IRanges"       "S4Vectors"
##  [5] "BiocGenerics"  "stats4"        "enrichplot"    "ggplot2"
##  [9] "stats"         "graphics"      "grDevices"     "utils"
## [13] "datasets"      "methods"       "base"
```

```
##
## [[5]]
##  [1] "ChIPseeker"    "GenomicRanges" "GenomeInfoDb"  "IRanges"
##  [5] "S4Vectors"     "BiocGenerics"  "stats4"        "enrichplot"
##  [9] "ggplot2"       "stats"         "graphics"      "grDevices"
## [13] "utils"         "datasets"      "methods"       "base"
##
## [[6]]
##  [1] "TxDb.Hsapiens.UCSC.hg38.knownGene" "GenomicFeatures"
##  [3] "AnnotationDbi"                     "Biobase"
##  [5] "ChIPseeker"                        "GenomicRanges"
##  [7] "GenomeInfoDb"                      "IRanges"
##  [9] "S4Vectors"                         "BiocGenerics"
## [11] "stats4"                            "enrichplot"
## [13] "ggplot2"                           "stats"
## [15] "graphics"                          "grDevices"
## [17] "utils"                             "datasets"
## [19] "methods"                           "base"
##
## [[7]]
##  [1] "org.Hs.eg.db"                      "TxDb.Hsapiens.UCSC.hg38.knownGene"
##  [3] "GenomicFeatures"                   "AnnotationDbi"
##  [5] "Biobase"                           "ChIPseeker"
##  [7] "GenomicRanges"                     "GenomeInfoDb"
##  [9] "IRanges"                           "S4Vectors"
## [11] "BiocGenerics"                      "stats4"
## [13] "enrichplot"                        "ggplot2"
## [15] "stats"                             "graphics"
## [17] "grDevices"                         "utils"
## [19] "datasets"                          "methods"
## [21] "base"
##
## [[8]]
##  [1] "clusterProfiler"                   "org.Hs.eg.db"
##  [3] "TxDb.Hsapiens.UCSC.hg38.knownGene" "GenomicFeatures"
##  [5] "AnnotationDbi"                     "Biobase"
##  [7] "ChIPseeker"                        "GenomicRanges"
##  [9] "GenomeInfoDb"                      "IRanges"
## [11] "S4Vectors"                         "BiocGenerics"
## [13] "stats4"                            "enrichplot"
## [15] "ggplot2"                           "stats"
## [17] "graphics"                          "grDevices"
## [19] "utils"                             "datasets"
## [21] "methods"                           "base"
##
## [[9]]
##  [1] "DESeq2"                            "SummarizedExperiment"
##  [3] "MatrixGenerics"                    "matrixStats"
##  [5] "clusterProfiler"                   "org.Hs.eg.db"
##  [7] "TxDb.Hsapiens.UCSC.hg38.knownGene" "GenomicFeatures"
##  [9] "AnnotationDbi"                     "Biobase"
## [11] "ChIPseeker"                        "GenomicRanges"
## [13] "GenomeInfoDb"                      "IRanges"
## [15] "S4Vectors"                         "BiocGenerics"
```

```
## [17] "stats4"                          "enrichplot"
## [19] "ggplot2"                         "stats"
## [21] "graphics"                        "grDevices"
## [23] "utils"                           "datasets"
## [25] "methods"                         "base"
```

2. Extract Raw Data (.tar)

We first extract the compressed .tar file containing the raw ATAC-seq BED files.

```
raw_tar_file <- "C:/users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW.tar"
extract_dir  <- "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW"

if (!dir.exists(extract_dir)) dir.create(extract_dir, recursive = TRUE)
untar(tarfile = raw_tar_file, exdir = extract_dir)

bed_files <- list.files(extract_dir, pattern = "\\.bed\\.gz$", full.names = TRUE)
bed_files
```

```
##  [1] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264802_C15_0_1.filterBL.bed.gz"
##  [2] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264803_C15_0_2.filterBL.bed.gz"
##  [3] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264804_C15_2_1.filterBL.bed.gz"
##  [4] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264805_C15_2_2.filterBL.bed.gz"
##  [5] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264806_C15_4_1.filterBL.bed.gz"
##  [6] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264807_C15_4_2.filterBL.bed.gz"
##  [7] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264808_C15_30_1.filterBL.bed.gz"
##  [8] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264809_C15_30_2.filterBL.bed.gz"
##  [9] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264810_C20_0_1.filterBL.bed.gz"
## [10] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264811_C20_0_2.filterBL.bed.gz"
## [11] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264812_C20_2_1.filterBL.bed.gz"
## [12] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264813_C20_2_2.filterBL.bed.gz"
## [13] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264814_C20_4_1.filterBL.bed.gz"
## [14] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264815_C20_4_2.filterBL.bed.gz"
## [15] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264816_C20_30_1.filterBL.bed.gz"
## [16] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264817_C20_30_2.filterBL.bed.gz"
## [17] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264818_H1_0_1.filterBL.bed.gz"
## [18] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264819_H1_0_2.filterBL.bed.gz"
## [19] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264820_H1_2_1.filterBL.bed.gz"
## [20] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264821_H1_2_2.filterBL.bed.gz"
## [21] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264822_H1_4_1.filterBL.bed.gz"
## [22] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264823_H1_4_2.filterBL.bed.gz"
## [23] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264824_H1_30_1.filterBL.bed.gz"
## [24] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264825_H1_30_2.filterBL.bed.gz"
## [25] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264826_H9_0_1.filterBL.bed.gz"
## [26] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264827_H9_0_2.filterBL.bed.gz"
## [27] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264828_H9_2_1.filterBL.bed.gz"
## [28] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264829_H9_2_2.filterBL.bed.gz"
## [29] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264830_H9_4_1.filterBL.bed.gz"
## [30] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264831_H9_4_2.filterBL.bed.gz"
## [31] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264832_H9_30_1.filterBL.bed.gz"
## [32] "C:/Users/OEM/Desktop/ATACseq-DiffCardio/data/GSE85330_RAW/GSM2264833_H9_30_2.filterBL.bed.gz"
```

3. Convert First Sample to GRanges

We now load the first BED file as an example and convert it into a GRanges object.

```
bed_raw <- read.table(bed_files[1], header = FALSE)
gr <- GRanges(
  seqnames = bed_raw$V1,
  ranges   = IRanges(start = bed_raw$V2 + 1, end = bed_raw$V3),
  strand   = "*",
  score    = bed_raw$V5,
  name     = bed_raw$V4
)

output_dir <- "C:/Users/OEM/Desktop/ATACseq-DiffCardio/output"
if (!dir.exists(output_dir)) dir.create(output_dir, recursive = TRUE)
saveRDS(gr, file = file.path(output_dir, "ATAC_peaks_GRanges_sample1.rds"))
```

4. Peak Annotation

We annotate peaks using ChIPseeker and the human genome reference (hg38).

```
peak_gr <- readRDS(file.path(output_dir, "ATAC_peaks_GRanges_sample1.rds"))
txdb <- TxDb.Hsapiens.UCSC.hg38.knownGene

peak_anno <- annotatePeak(peak_gr, TxDb = txdb, annoDb = "org.Hs.eg.db")
```

```
## >> preparing features information...      2025-07-31 8:02:16 AM
## >> identifying nearest features...        2025-07-31 8:02:18 AM
## >> calculating distance from peak to TSS...   2025-07-31 8:02:19 AM
## >> assigning genomic annotation...        2025-07-31 8:02:19 AM
## >> adding gene annotation...          2025-07-31 8:02:47 AM

## 'select()' returned 1:many mapping between keys and columns

## >> assigning chromosome lengths           2025-07-31 8:02:48 AM
## >> done...                     2025-07-31 8:02:48 AM
```

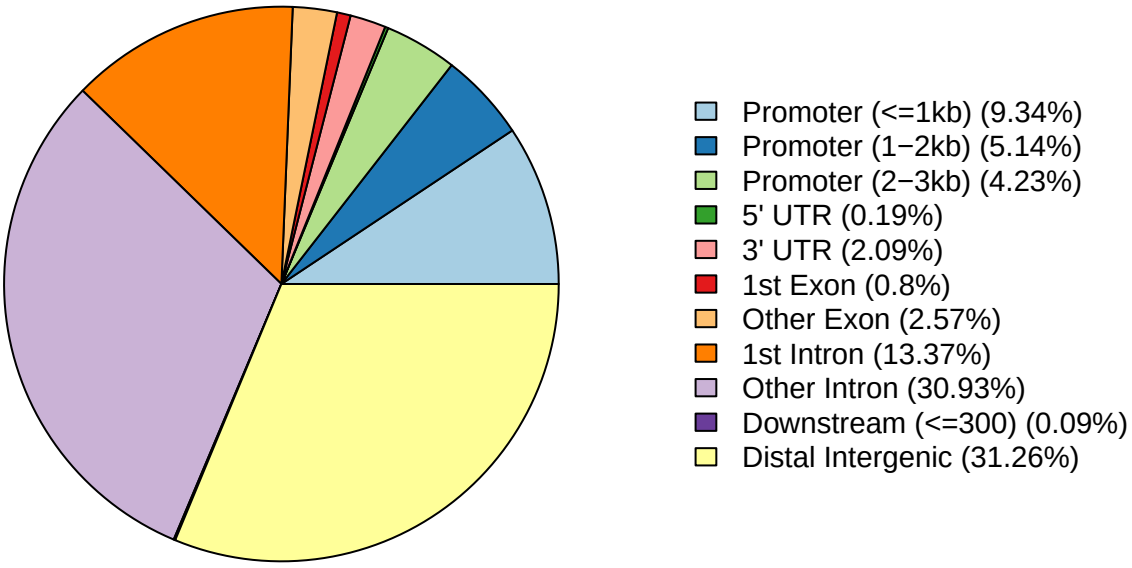
```
head(as.data.frame(peak_anno))
```

```
##   seqnames  start    end width strand score
## 1     chr1 713921 714463   543      *   362
## 2     chr1 762659 762976   318      *   161
## 3     chr1 781078 781374   297      *   114
## 4     chr1 794131 794334   204      *    79
## 5     chr1 839928 840222   295      *    80
## 6     chr1 894671 894937   267      *    41
##                                                                                 name
## 1 /srv/gsfs0/projects/snyder/qingliu/ESandIPS/ATACSEQ/test/C15_peakcall/Replicates/C15_0_1_peak_4
## 2 /srv/gsfs0/projects/snyder/qingliu/ESandIPS/ATACSEQ/test/C15_peakcall/Replicates/C15_0_1_peak_5
## 3 /srv/gsfs0/projects/snyder/qingliu/ESandIPS/ATACSEQ/test/C15_peakcall/Replicates/C15_0_1_peak_6
```

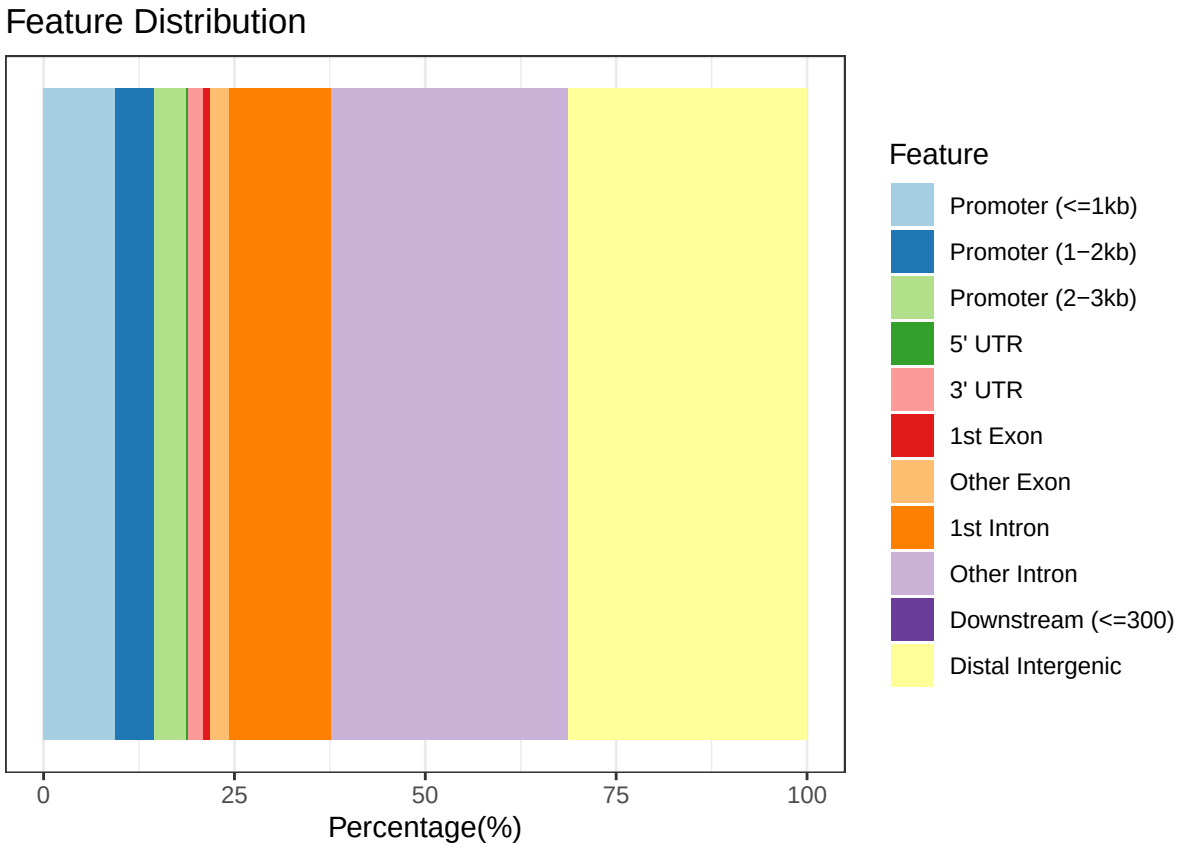
```
## 4 /srv/gsfs0/projects/snyder/qingliu/ESandIPS/ATACSEQ/test/C15_peakcall/Replicates/C15_0_1_peak_7
## 5 /srv/gsfs0/projects/snyder/qingliu/ESandIPS/ATACSEQ/test/C15_peakcall/Replicates/C15_0_1_peak_8
## 6 /srv/gsfs0/projects/snyder/qingliu/ESandIPS/ATACSEQ/test/C15_peakcall/Replicates/C15_0_1_peak_9
##                                           annotation geneChr geneStart geneEnd
## 1     Intron (ENST00000419394.2/81399, intron 1 of 3)       1    701936  720150
## 2 Intron (ENST00000635509.2/105378947, intron 1 of 3)       1    764723  774280
## 3                                  Promoter (2-3kb)       1    778972  808378
## 4 Intron (ENST00000655765.1/105378580, intron 1 of 2)       1    803836  806580
## 5    Intron (ENST00000624927.3/643837, intron 1 of 2)       1    831617  854096
## 6                                 Distal Intergenic       1    904834  914971
##   geneLength geneStrand    geneId      transcriptId distanceToTSS
## 1      18215          2     81399 ENST00000441245.5         5687
## 2       9558          2 100288069 ENST00000428504.2        11304
## 3      29407          1 105378580 ENST00000443772.2         2106
## 4       2745          1 105378580 ENST00000655384.1        -9502
## 5      22480          1    643837 ENST00000688420.1         8311
## 6      10138          1    284600 ENST00000715285.1        -9897
##           ENSEMBL       SYMBOL
## 1 ENSG00000284662       OR4F16
## 2            <NA> LOC100288069
## 3 ENSG00000237491    LINC01409
## 4 ENSG00000237491    LINC01409
## 5 ENSG00000228794    LINC01128
## 6            <NA>    LOC284600
##                                         GENENAME
## 1 olfactory receptor family 4 subfamily F member 16
## 2                     uncharacterized LOC100288069
## 3       long intergenic non-protein coding RNA 1409
## 4       long intergenic non-protein coding RNA 1409
## 5       long intergenic non-protein coding RNA 1128
## 6                         uncharacterized LOC284600
```

Annotation Plots

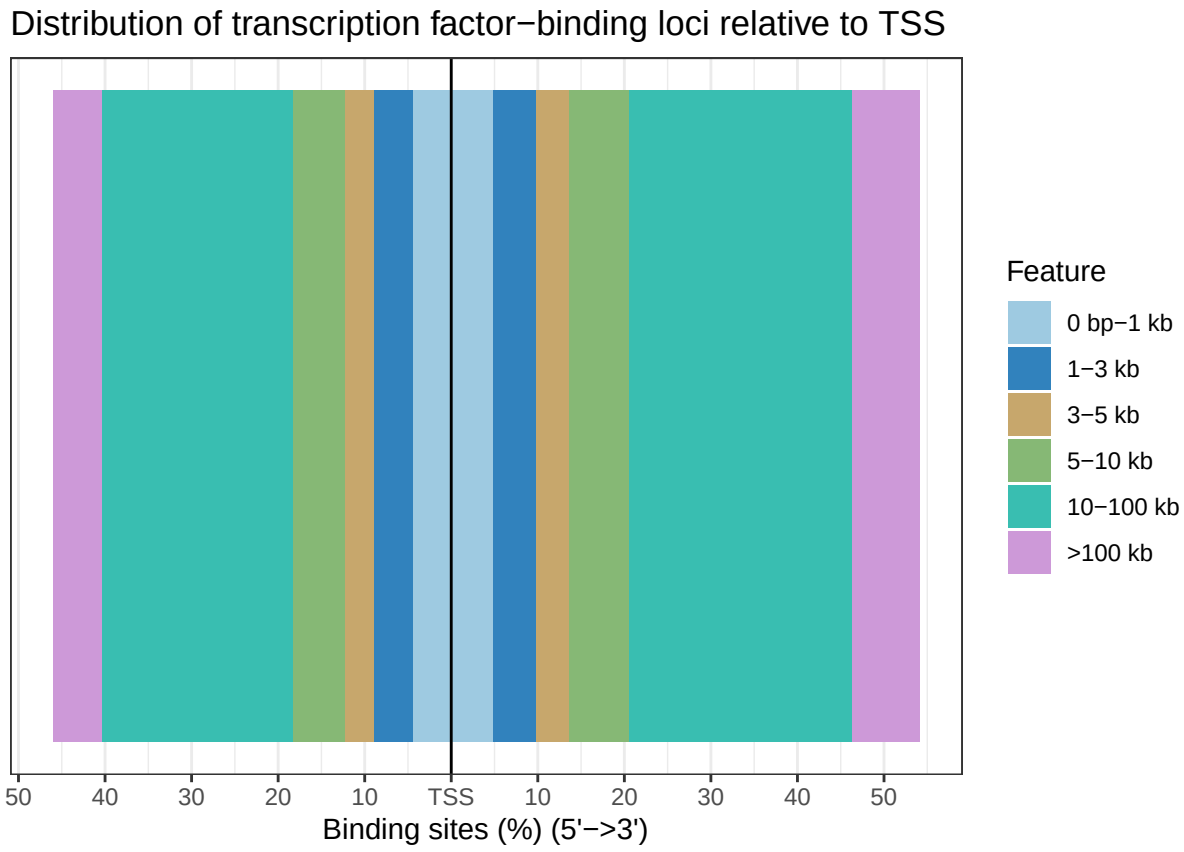
```
plotAnnoPie(peak_anno)
```




```
plotAnnoBar(peak_anno)
```




```
plotDistToTSS(peak_anno)
```



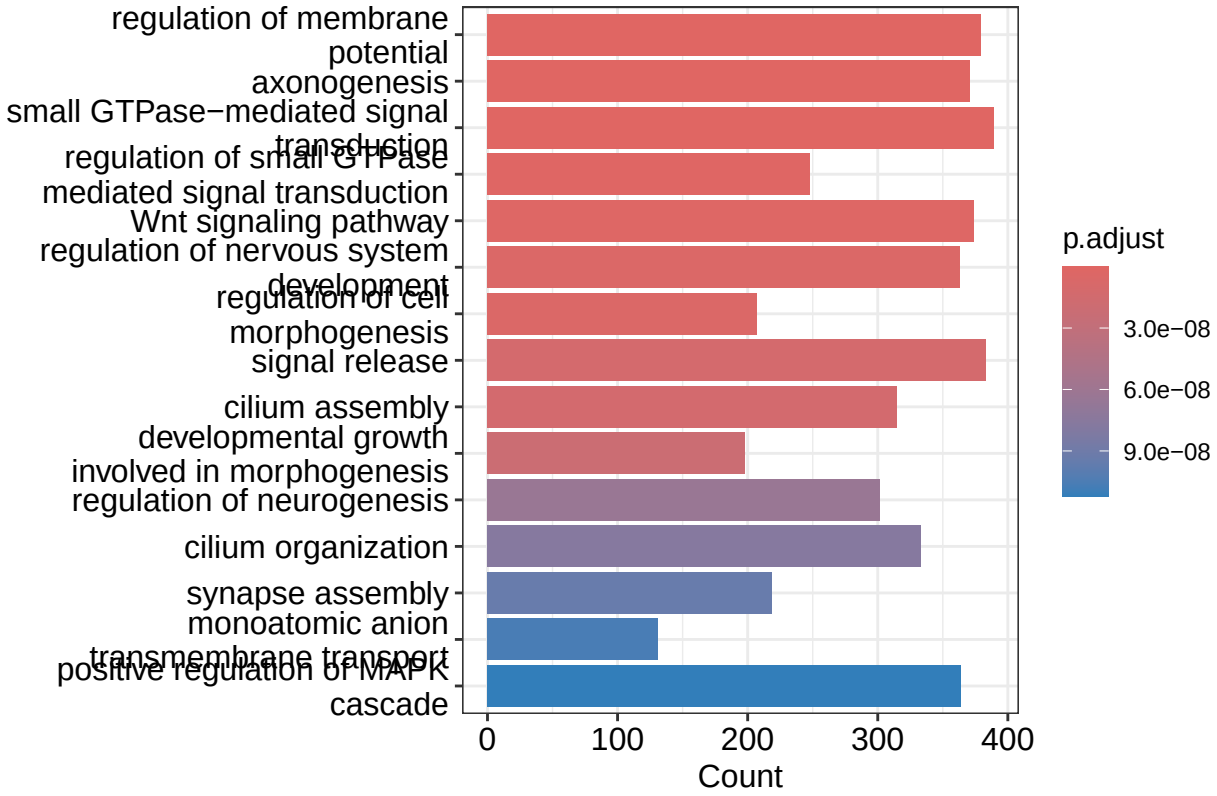
5. GO Enrichment Analysis

We test whether annotated genes are enriched in specific biological processes.

```
genes <- unique(na.omit(as.data.frame(peak_anno)$SYMBOL))

ego <- enrichGO(
  gene          = genes,
  OrgDb         = org.Hs.eg.db,
  keyType       = "SYMBOL",
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.01,
  qvalueCutoff  = 0.05
)

barplot(ego, showCategory = 15)
```



6. Build Consensus Peak Matrix

```
bed_paths <- list(
  D0_1   = file.path(extract_dir, "GSM2264802_C15_0_1.filterBL.bed.gz"),
  D0_2   = file.path(extract_dir, "GSM2264803_C15_0_2.filterBL.bed.gz"),
  D30_1  = file.path(extract_dir, "GSM2264808_C15_30_1.filterBL.bed.gz"),
  D30_2  = file.path(extract_dir, "GSM2264809_C15_30_2.filterBL.bed.gz")
)

read_bed <- function(path) {
  df <- read.table(path, header = FALSE)
  GRanges(seqnames = df$V1,
          ranges   = IRanges(start = df$V2 + 1, end = df$V3),
          strand   = "*")
}

peak_list <- lapply(bed_paths, read_bed)
all_peaks <- GenomicRanges::reduce(unlist(GRangesList(peak_list)))
consensus_peaks <- resize(all_peaks, width = 250, fix = "center")

count_matrix <- sapply(peak_list, function(peaks) {
  countOverlaps(consensus_peaks, peaks)
})

rownames(count_matrix) <- paste0("Peak_", seq_len(nrow(count_matrix)))
colnames(count_matrix) <- names(peak_list)
```

8. Differential Accessibility Analysis (Day 0 vs Day 30)

We identify peaks that change in accessibility between Day 0 and Day 30.

```
coldata <- data.frame(
  row.names = colnames(count_matrix),
  condition = c("D0", "D0", "D30", "D30")
)

dds <- DESeqDataSetFromMatrix(
  countData = count_matrix,
  colData   = coldata,
  design    = ~ condition
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
dds <- estimateSizeFactors(dds)
dds <- estimateDispersionsGeneEst(dds)
dispersions(dds) <- mcols(dds)$dispGeneEst
dds <- nbinomWaldTest(dds)

res <- results(dds)
head(res)
```

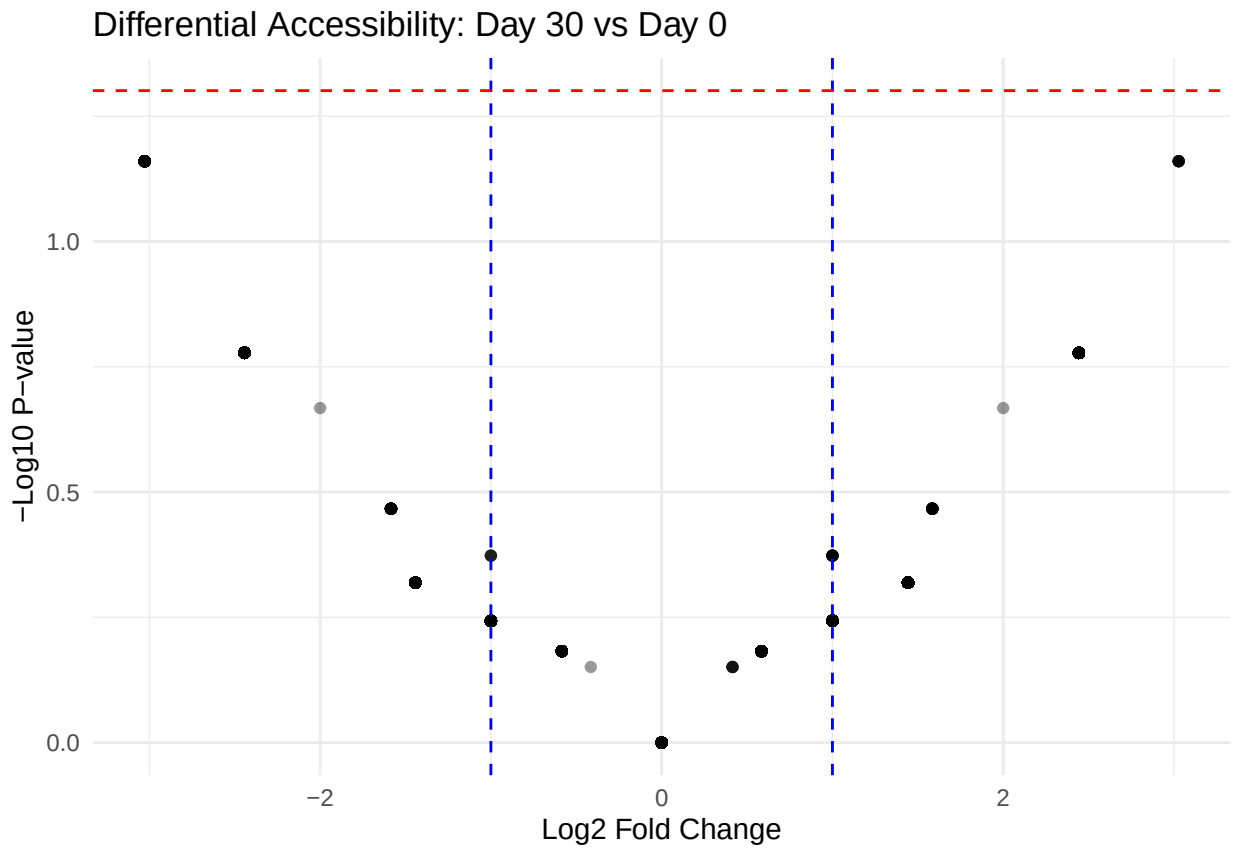
```
## log2 fold change (MLE): condition D30 vs D0
## Wald test p-value: condition D30 vs D0
## DataFrame with 6 rows and 6 columns
##          baseMean log2FoldChange     lfcSE         stat    pvalue      padj
##         <numeric>      <numeric> <numeric>    <numeric> <numeric> <numeric>
## Peak_1      1.00    4.65098e-16   1.44269  3.22382e-16  1.000000  1.000000
## Peak_2      0.50   -2.44269e+00   1.76693 -1.38245e+00  0.166834  0.509897
## Peak_3      0.25   -1.44269e+00   2.04027 -7.07109e-01  0.479499  0.617500
## Peak_4      0.25   -1.44269e+00   2.04027 -7.07109e-01  0.479499  0.617500
## Peak_5      0.25   -1.44269e+00   2.04027 -7.07109e-01  0.479499  0.617500
## Peak_6      0.25   -1.44269e+00   2.04027 -7.07109e-01  0.479499  0.617500
```

8. Volcano Plot

```
res_df <- as.data.frame(res)
res_df$PeakID <- rownames(res_df)

ggplot(res_df, aes(x = log2FoldChange, y = -log10(pvalue))) +
  geom_point(alpha = 0.4) +
  geom_vline(xintercept = c(-1, 1), linetype = "dashed", color = "blue") +
  geom_hline(yintercept = -log10(0.05), linetype = "dashed", color = "red") +
  theme_minimal() +
  labs(
    title = "Differential Accessibility: Day 30 vs Day 0",
```

```
    x     = "Log2 Fold Change",
    y     = "-Log10 P-value"
  )
```



Conclusion

This pipeline identifies genomic regions whose accessibility changes during cardiomyocyte differentiation. Further integration with gene expression (RNA-seq) data could enhance biological insights.