mod2_group

2024-11-15

RNA-Seq Data Analysis Workflow

This markdown file documents the steps and commands for processing RNA-Seq data. Each step is explained and the command is provided for reproducibility.

1) Cut Adapters

To remove low-quality sequences and adapter contamination, we use a trimming tool. This step ensures that only high-quality reads are retained, which is crucial for accurate alignment in downstream analysis. Here, `cutadapt` is utilized for adapter trimming, which effectively removes unwanted sequences based on specific adapter patterns.

```
# Cut adapters command
sbatch trim_reads2.sh INPUT_PATH OUTPUT_PATH
```

2) Align reads command

Once the data has been cleaned of adapters, the next step is to align the high-quality reads to a reference genome. We use hisat2 for this alignment process, as it is specifically optimized for RNA-Seq data. This alignment step maps each read to its corresponding position in the genome, which is essential for quantifying gene expression accurately.

```
sbatch hisat2.sh INPUT_PATH REFERENCE_GENOME_PATH OUTPUT_PATH
```

3) Merge and sort BAM files command

Following alignment, the generated BAM files need to be merged and sorted. This step consolidates all aligned reads, organizing them by genomic position. Sorting is critical for efficient access. We use a merging and sorting script to achieve this.

```
sbatch sam_merge_sort_bam.sh INPUT_PATH OUTPUT_PATH GTF_FILE
```

4) Quantify gene counts command

Next we quantify gene expression by counting reads that align to each gene. featureCounts is used here to count the aligned reads for each annotated gene in the reference genome.

```
sbatch featureCounts.sh INPUT_PATH OUTPUT_PATH GTF_FILE
```

5) Plotting the gene expression

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(openxlsx)
library(ggplot2)

# Load the data
data <- read.xlsx("/home/flyingnimbus2/Desktop/Module_2/out/genes.xlsx")
data = data[, -1] # remove geneID
head(data)
```

```
##   C1 C2 C3 T1 T2 T3 V1 V2 V3
## 1 64 38 47 23 45 39 60 30 33
## 2  0  0  0  0  0  0  0  0  0
## 3 35 42 29 26 29 29 44 39 27
## 4  8  5 14 11 10  8  2 14 11
## 5 14 24 13  8  2 11 14 14 16
## 6  0  2  0  0  0  0  0  0  0
```

```
# pivot table
data <- pivot_longer(
  data,
  names(data)
)
colnames(data) <- c("Treat", "Expression") # name columns

# combine replicates
data <- mutate(data, Treat = gsub("\\d", "", Treat))

# sort
data <- arrange(data, Treat)
head(data)
```

```
## # A tibble: 6 x 2
##   Treat Expression
##   <chr>      <dbl>
## 1 C             64
## 2 C             38
## 3 C             47
```

```
## 4 C              0
## 5 C              0
## 6 C              0
```

```
# Create the box plot
ggplot(
  data,
  aes(x = Treat, y = Expression, fill = Treat)
) +
geom_boxplot() +
labs(
    title = "Gene Expression Box Plot",
    x = "Condition",
    y = "Expression"
) +
scale_y_log10()
```

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## Warning: Removed 432834 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```