GENCODEAnalysis

NishaMadavaprasad

2025-09-03

Introduction

This analysis explores transcript diversity in human genes using GENCODE v, `gencode_version`, annotations. We investigate how many genes produce multiple transcript isoforms through alternative splicing and other mechanisms.

Objectives

- Quantify the number of genes and transcripts in the human genome
- Analyze the distribution of transcript counts per gene
- Visualize patterns of transcript diversity
- Identify genes with high transcript complexity

Package Installation and Loading

```
#\' Install Required Packages
installPackages <- function() {
  # Install BiocManager if not available
  if (!requireNamespace("BiocManager", quietly = TRUE)) {
    install.packages("BiocManager")
  }

  # Bioconductor packages
  bioc_packages <- c("GenomicFeatures", "txdbmaker", "rtracklayer")
  for (pkg in bioc_packages) {
    if (!requireNamespace(pkg, quietly = TRUE)) {
      BiocManager::install(pkg, ask = FALSE)
```

```
    }
  }

  # CRAN packages
  cran_packages <- c("R.utils", "dplyr", "ggplot2", "knitr", "DT", "scales")
  installed_packages <- cran_packages %in% rownames(installed.packages())
  if (any(!installed_packages)) {
    install.packages(cran_packages[!installed_packages])
  }

  # Load all libraries
  suppressPackageStartupMessages({
    library(rtracklayer)
    library(R.utils)
    library(GenomicFeatures)
    library(dplyr)
    library(ggplot2)
    library(txdbmaker)
    library(knitr)
    library(DT)
    library(scales)
  })

  cat("All packages loaded successfully\\n")
}

# Install and load packages
installPackages()
```

```
All packages loaded successfully\n
```

Data Download and Preprocessing

```
#\' Download GENCODE GTF File
downloadGencodeData <- function(gencode_version = ', gencode_version, ', data_dir = "data/")
  # Create data directory if it doesn\'t exist
  if (!dir.exists(data_dir)) {
    dir.create(data_dir, recursive = TRUE)
  }
```

```
  # Construct download URL
  base_url <- "http://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_"
  download_url <- paste0(base_url, gencode_version, "/gencode.v", gencode_version, ".annotati

  # File paths
  compressed_file <- paste0(data_dir, "gencode.v", gencode_version, ".annotation.gtf.gz")
  uncompressed_file <- paste0(data_dir, "gencode.v", gencode_version, ".annotation.gtf")

  # Download compressed file if it doesn\'t exist
  if (!file.exists(compressed_file)) {
    cat("Downloading GENCODE v", gencode_version, " GTF file...\\n")
    download.file(url = download_url, destfile = compressed_file)
  }

  # Uncompress file if needed
  if (file.exists(compressed_file) && !file.exists(uncompressed_file)) {
    cat("Uncompressing GTF file...\\n")
    R.utils::gunzip(compressed_file, remove = FALSE)
  }

  return(uncompressed_file)
}

# Download GENCODE data
gtf_file <- downloadGencodeData(gencode_version=46)
cat("GTF file location:", gtf_file, "\n")
```

```
GTF file location: data/gencode.v46.annotation.gtf
```

Database Creation

```
#\' Create TxDb Object from GTF
createTxDb <- function(gtf_file) {
  txdb <- makeTxDbFromGFF(
    file = gtf_file,
    format = "gtf",
    dataSource = "gencode",
    organism = "Homo sapiens"
  )
```

```
  return(txdb)
}

# Create transcript database
txdb <- createTxDb(gtf_file)
```

```
Import genomic features from the file as a GRanges object ... OK
Prepare the 'metadata' data frame ... OK
Make the TxDb object ...
```

```
Warning in .get_cds_IDX(mcols0$type, mcols0$phase): The "phase" metadata column contains non-
  stop_codon. This information was ignored.
```

```
Warning in .makeTxDb_normarg_chrominfo(chrominfo): genome version information
is not available for this TxDb object
```

```
OK
```

```
# Display TxDb information
txdb
```

```
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: gencode
# Organism: Homo sapiens
# Taxonomy ID: 9606
# miRBase build ID: NA
# Genome: NA
# Nb of transcripts: 254070
# Db created by: txdbmaker package from Bioconductor
# Creation time: 2025-09-03 19:13:08 -0400 (Wed, 03 Sep 2025)
# txdbmaker version at creation time: 1.4.2
# RSQLite version at creation time: 2.4.3
# DBSCHEMAVERSION: 1.2
```

Basic Gene and Transcript Counts

```r
# Extract basic statistics
gene_count <- length(genes(txdb))
transcript_count <- length(transcripts(txdb))

cat("Total number of genes:", format(gene_count, big.mark = ","), "\n")
```

```
Total number of genes: 63,086
```

```r
cat("Total number of transcripts:", format(transcript_count, big.mark = ","), "\n")
```

```
Total number of transcripts: 254,070
```

Transcript Diversity Analysis

```r
# Get transcripts grouped by gene
gencode_version = 26
tx_by_gene <- transcriptsBy(txdb, by = "gene")
tx_counts <- lengths(tx_by_gene)

# Calculate key statistics
genes_with_multiple_tx <- sum(tx_counts > 1)
percent_multiple_tx <- round(100 * genes_with_multiple_tx / length(tx_counts), 1)
max_transcripts <- max(tx_counts)
genes_with_max_tx <- sum(tx_counts == max_transcripts)

# Summary table
summary_stats <- data.frame(
  Metric = c("Total Genes", "Total Transcripts", "Genes with Multiple Transcripts", "Percenta
  Value = c(format(gene_count, big.mark = ","),
            format(transcript_count, big.mark = ","),
            format(genes_with_multiple_tx, big.mark = ","),
            paste0(percent_multiple_tx, "%"),
            max_transcripts)
)

kable(
  summary_stats,
  caption = paste("GENCODE v", gencode_version, "Transcript Diversity Summary")
)
```

Table 1: GENCODE v 26 Transcript Diversity Summary

Metric	Value
Total Genes	63,086
Total Transcripts	254,070
Genes with Multiple Transcripts	23,769
Percentage with Multiple Transcripts	37.7%
Maximum Transcripts per Gene	296

Detailed Gene Analysis

```
# Create detailed data frame
gene_transcript_df <- data.frame(
  gene_id = names(tx_by_gene),
  transcript_count = as.numeric(tx_counts),
  stringsAsFactors = FALSE
) %>%
  mutate(
    transcript_category = case_when(
      transcript_count == 1 ~ "1 transcript",
      transcript_count >= 2 & transcript_count <= 5 ~ "2-5 transcripts",
      transcript_count > 5 ~ ">5 transcripts"
    )
  ) %>%
  mutate(
    transcript_category = factor(
      transcript_category,
      levels = c("1 transcript", "2-5 transcripts", ">5 transcripts")
    )
  )
head(gene_transcript_df)
```

```
             gene_id transcript_count transcript_category
1 ENSG00000000003.16                4     2-5 transcripts
2  ENSG00000000005.6                2     2-5 transcripts
3 ENSG00000000419.14               16      >5 transcripts
4 ENSG00000000457.14                5     2-5 transcripts
5 ENSG00000000460.17                9      >5 transcripts
6 ENSG00000000938.13                7      >5 transcripts
```

```r
write.csv(
  gene_transcript_df,
  file = "gene_transcript_summary.csv",
  row.names = FALSE
)
```

Main Distribution Plot

```r
# Prepare plot data
plot_data <- gene_transcript_df %>%
  count(transcript_category) %>%
  mutate(
    percentage = round(100 * n / sum(n), 1),
    category = "All Genes"
  )
# Create the main plot
main_plot <- ggplot(plot_data, aes(x = category, y = n, fill = transcript_category)) +
  geom_col(width = 0.6, color = "white", linewidth = 0.5) +
  geom_text(
    aes(label = paste0(format(n, big.mark = ","), "\\n(", percentage, "%)")),
    position = position_stack(vjust = 0.5),
    color = "white",
    fontface = "bold",
    size = 4
  ) +
  scale_fill_manual(
    values = c("1 transcript" = "#2E86AB",
               "2-5 transcripts" = "#A23B72",
               ">5 transcripts" = "#F18F01")
  ) +
  scale_y_continuous(labels = comma_format()) +
  labs(
    title = "Distribution of Human Genes by Transcript Count",
    subtitle = paste("GENCODE v", gencode_version, "Analysis | Total genes:", format(gene_cou
    x = "",
    y = "Number of Genes",
    fill = "Transcript Category",
    caption = "Data source: GENCODE v", gencode_version, paste0("GENCODE v", gencode_version
  ) +
  theme_minimal() +
```

```
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, color = "gray50"),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    legend.position = "bottom",
    legend.title = element_text(face = "bold")
  )

print(main_plot)
```

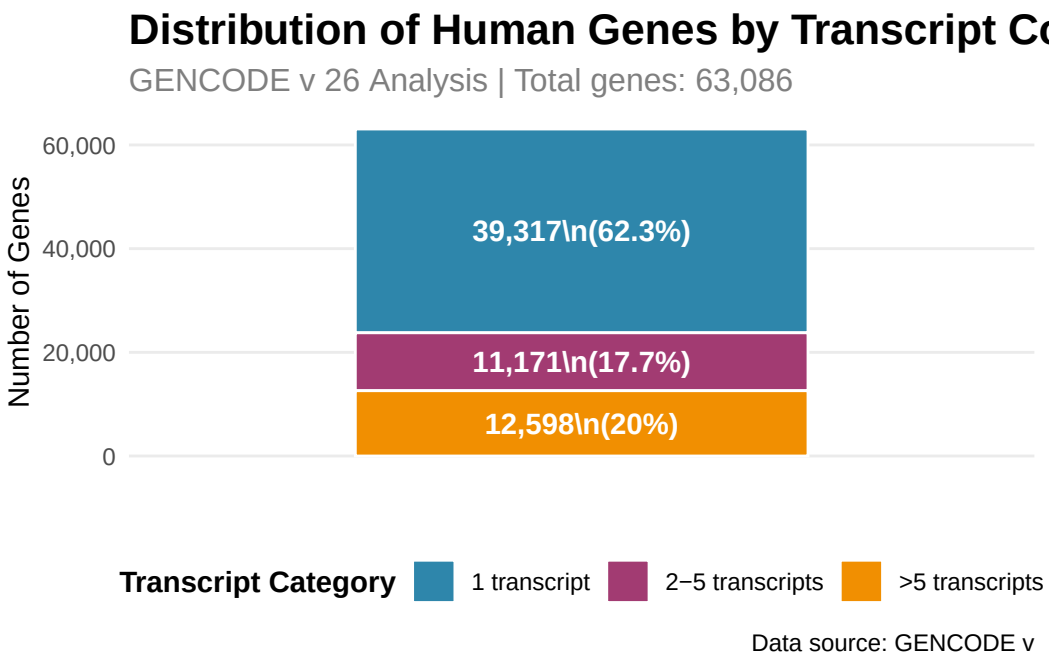

Figure 1: Distribution of genes by transcript count categories