Package 'GeuvadisTranscriptExpr'

November 10, 2026

Type Package

Title Data package with transcript expression and bi-allelic genotypes from the GEUVADIS project

Version 1.40.0

Date 2025-06-27

Description Provides transcript expression and bi-allelic genotypes corresponding to the chromosome 19 for CEU individuals from the GEUVADIS project, Lappalainen et al.

Depends R (>= 3.3.0)

License GPL (>= 3)

LazyData true

biocViews Homo_sapiens_Data, SNPData, Genome, RNASeqData, SequencingData, ExpressionData

VignetteBuilder knitr

Suggests limma, rtracklayer, GenomicRanges, Rsamtools, VariantAnnotation, tools, BiocStyle, knitr, testthat

NeedsCompilation no

Author Malgorzata Nowicka [aut, cre]

Maintainer Malgorzata Nowicka <gosia.nowicka@uzh.ch>

git_url https://git.bioconductor.org/packages/GeuvadisTranscriptExpr

git_branch RELEASE_3_23

git_last_commit 302da61

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-10

Contents

counts 2

Index **4**

counts *Sample data for sQTL analysis*

Description

Subsets of raw data available in this package and saved as Rdata objects for faster loading.

Usage

```
counts

gene_ranges

genotypes

snp_ranges
```

Format

`counts` is a data frame with subset of counts from TrQuantCount_CEU_chr19.tsv

`gene_ranges` is a GRanges object containing subset of gene coordinates from genes_chr19.bed

`genotypes` is a data frame with subset of genotypes from genotypes_CEU_chr19.tsv

`snp_ranges` is a Granges object containing subset of SNP coordinates from genotypes_CEU_chr19.tsv

For all the details on how these data sets were produced, see examples.

Value

`counts`, `gene_ranges`, `genotypes`, `snp_ranges`

Source

Lappalainen T, Sammeth M, Friedlander MR, et al. Transcriptome and genome sequencing uncovers functional variation in humans. Nature. 2013;501(7468):506-11

Examples

```
library(rtracklayer)
data_dir <- system.file("extdata", package = "GeuvadisTranscriptExpr")

gene_id_subset <- readLines(file.path(data_dir, "gene_id_subset.txt"))
snp_id_subset <- readLines(file.path(data_dir, "snp_id_subset.txt"))

# Load gene ranges with names!
gene_ranges <- import(file.path(data_dir, "genes_chr19.bed"))
names(gene_ranges) <- mcols(gene_ranges)$name

gene_ranges <- gene_ranges[gene_id_subset, ]
```

```
# Load transcript counts
counts <- read.table(file.path(data_dir, "TrQuantCount_CEU_chr19.tsv"),
                     header = TRUE, sep = "\t", as.is = TRUE)

counts <- counts[counts$Gene_Symbol %in% gene_id_subset, ]

# Load genotypes
genotypes <- read.table(file.path(data_dir, "genotypes_CEU_chr19.tsv"),
                        header = TRUE, sep = "\t", as.is = TRUE)

genotypes <- genotypes[genotypes$snpId %in% snp_id_subset, ]

# Create SNP ranges with names!
snp_ranges <- GRanges(Rle(genotypes$chr), IRanges(genotypes$start,
                                                  genotypes$end))
names(snp_ranges) <- genotypes$snpId
```

Index

∗ **datasets**
counts, 2

counts, 2

gene_ranges *(*counts*)*, 2
genotypes *(*counts*)*, 2

snp_ranges *(*counts*)*, 2