# Package 'miRmine'

October 16, 2019

**Type** Package

**Title** Data package with miRNA-seq datasets from miRmine database as RangedSummarizedExperiment

**Version** 1.6.0

**Date** 2017-08-29

**Description** miRmine database is a collection of expression profiles from different publicly available miRNA-seq datasets, Panwar et al (2017) miRmine: A Database of Human miRNA Expression, prepared with this data package as RangedSummarizedExperiment.

**Depends** R (>= 3.4), SummarizedExperiment

**Suggests** BiocStyle, knitr, rmarkdown, DESeq2

**License** GPL (>= 3)

**LazyData** true

**biocViews** Homo_sapiens_Data, RNASeqData, SequencingData, ExpressionData

**VignetteBuilder** knitr

**NeedsCompilation** no

**RoxygenNote** 6.0.1

**git_url** https://git.bioconductor.org/packages/miRmine

**git_branch** RELEASE_3_9

**git_last_commit** 970ce64

**git_last_commit_date** 2019-05-02

**Date/Publication** 2019-10-15

**Author** Dusan Randjelovic [aut, cre]

**Maintainer** Dusan Randjelovic <dusan.randjelovic@sbgenomics.com>

## R topics documented:

`miRmine` *miRmine dataset*

## Description

miRmine dataset, from Panwar et al (2017) miRmine: A Database of Human miRNA Expression

## Usage

```
data("miRmine")
```

## Format

`miRmine` A RangedSummarizedExperiment object.

For all the details on how this dataset was produced, see examples.

## Source

Panwar et al (2017) miRmine: A Database of Human miRNA Expression

## Examples

```
## Not run:
library(GenomicRanges)
library(rtracklayer)
library(SummarizedExperiment)
library(Biostrings)

ext.data <- system.file("extdata", package = "miRmine")
hsa.gff3.file = file.path(ext.data, "hsa.gff3")
mature.fa.file = file.path(ext.data, "mature.fa")
miRmine.info.file = file.path(ext.data, "miRmine-info.txt")
miRmine.tissues.file = file.path(ext.data, "miRmine-tissues.csv")
miRmine.cell.lines.file = file.path(ext.data, "miRmine-cell-lines.csv")

gffRangedData.all <- import.gff3(hsa.gff3.file, genome="GRCh38")
gffRangedData.all$source = "miRBase v21"
gffRangedData.all$UniqueName = gffRangedData.all$Name
for (id in seq_along(as.character(gffRangedData.all$ID))){
    name = gffRangedData.all[id, ]$Name
    derives_from = gffRangedData.all[id, ]$Derives_from
    if (!is.na(derives_from)){
        precursor =
            gffRangedData.all[gffRangedData.all$ID == derives_from, ]$Name
        gffRangedData.all[id, ]$UniqueName = paste(name, precursor, sep=".")
    }
}
gff = gffRangedData.all[gffRangedData.all$type == "miRNA"]
gff = sort(gff, by=~UniqueName)

tiss = read.csv(miRmine.tissues.file)
tiss$UniqueName =
    paste(tiss$Mature.miRNA.ID, tiss$Precursor.miRNA.ID, sep=".")
tiss = tiss[base::order(tiss$UniqueName), ]
```

```
diff.names = setdiff(tiss$UniqueName, gff$UniqueName) # 7 rows differ

cellines = read.csv(miRmine.cell.lines.file)
cellines$UniqueName =
    paste(cellines$Mature.miRNA.ID, cellines$Precursor.miRNA.ID, sep=".")
cellines = cellines[base::order(cellines$UniqueName), ]

setdiff(cellines$UniqueName, gff$UniqueName) # same 7 rows differ

tissue.mirnas.freq = base::sort(table(tiss$UniqueName))
gff.mirnas.freq = base::sort(table(gff$UniqueName))
setdiff(tissue.mirnas.freq, gff.mirnas.freq) # additional 2 rows duplicated
tissue.mirnas.freq[tissue.mirnas.freq > 1] # shows which rows are different

base::rownames(
    tiss[(tiss$UniqueName %in%
        c('hsa-miR-3142.hsa-mir-3142','hsa-miR-4487.hsa-mir-4487')),])

tiss = tiss[-c(624, 1213),]
tiss = tiss[!(tiss$UniqueName %in% diff.names), ]
cellines = cellines[-c(624, 1213),]
cellines = cellines[!(cellines$UniqueName %in% diff.names), ]

mirnas.unique.names = tiss$UniqueName
tiss.counts =
    tiss[, -which(names(tiss) %in%
        c("UniqueName", "Mature.miRNA.ID", "Precursor.miRNA.ID"))]
cellines.counts =
    cellines[, -which(names(cellines) %in%
        c("UniqueName", "Mature.miRNA.ID", "Precursor.miRNA.ID"))]
expression = as.matrix(cbind(tiss.counts, cellines.counts))
rownames(expression) = mirnas.unique.names

# add mirna sequences
library(Rsamtools)
fasta = FaFile(mature.fa.file)
mirna.string.set = scanFa(fasta)
newnames = strsplit(names(mirna.string.set), " ")
newnames = unlist(newnames)[ c(TRUE, rep(FALSE, 4)) ]
names(mirna.string.set) = newnames

dna.strings = list()
for (id in seq_along(gff)){
    name = gff[id, ]$Name
    unique_name = gff[id, ]$UniqueName
    dna.strings[[unique_name]] = mirna.string.set[[name]]
}
gff$mirna_seq = dna.strings

# construct RSE
meta = read.csv(miRmine.info.file, sep="\t")

miRmine =
    SummarizedExperiment(
        assays=SimpleList(counts=expression),
        rowData=NULL,
```

```
        rowRanges=gff,
        colData=meta
    )

## End(Not run)
```

# Index