Package 'prebs'

November 12, 2026

Title Probe region expression estimation for RNA-seq data for improved microarray comparability

Description The prebs package aims at making RNA-sequencing (RNA-seq) data more comparable to microarray data. The comparability is achieved by summarizing sequencing-based expressions of probe regions using a modified version of RMA algorithm. The pipeline takes mapped reads in BAM format as an input and produces either gene expressions or original microarray probe set expressions as an output.

Version 1.52.0

Author Karolis Uziela and Antti Honkela

Maintainer Karolis Uziela <karolis.uziela@scilifelab.se>

Depends R (>= 2.14.0), GenomicAlignments, affy, RPA

Imports parallel, methods, stats, GenomicRanges (>= 1.13.3), IRanges, Biobase, Seqinfo, S4Vectors

Suggests prebsdata, hgu133plus2cdf, hgu133plus2probe

License Artistic-2.0

Collate 'PREBS.R'

biocViews ImmunoOncology, Microarray, RNASeq, Sequencing, GeneExpression, Preprocessing

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

git_branch RELEASE_3_23

git_last_commit 4f6f3e6

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

calc_prebs 2
prebs 4

Index **5**

`calc_prebs` *Calculate PREBS values*

Description

`calc_prebs` calculates PREBS values for given set of BAM files.

Usage

```
calc_prebs(bam_files, probe_mapping_file, cdf_name = NULL, cluster = NULL,
  output_eset = TRUE, paired_ended_reads = FALSE, ignore_strand = TRUE,
  sum.method = "rpa")
```

Arguments

`bam_files`	A vector containing .bam files.
`probe_mapping_file`	A file containing probe mappings in the genome.
`cdf_name`	A name of CDF package to use in RMA algorithm. If cdf_name=NULL, the package name is inferred from the name of probe_mapping_file ("HGU133Plus2_Hs_ENSG_mapping.txt -> "hgu133plus2hsensgcdf")
`cluster`	A cluster object created using "makeCluster" function from "parellel" package. If cluster=NULL, no parallelization is used.
`output_eset`	If set to TRUE, the output of `calc_prebs` will be ExpressionSet object. Otherwise, the output will be a data frame.
`paired_ended_reads`	Set it to TRUE if your data contains paired-ended reads. Otherwise, the two read mates will be treated as independent units.
`ignore_strand`	If set to TRUE, then the strand is ignored while counting read overlaps with probe regions. If you use strand-specific RNA-seq protocol, set to FALSE, otherwise set it to TRUE.
`sum.method`	Microarray summarization method to be used. Can be either `rpa` or `rma`. The default mode is `rpa`.

Details

`calc_prebs` is the main function of `prebs` package that implements the whole pipeline. The function takes mapped reads in BAM format and probe sequence mappings as an input.

`calc_prebs` can run in two modes: `rpa` and `rma`. RMA is the classical microarray summarization algorithm developed by R. A. Irizarry et al. (2003), while RPA is a newer algorithm that was developed by L. Lahti et al. (2011). The default mode is `rpa`. NOTE: before `prebs` version 1.7.1 only RMA mode was available.

The output format depends on `output_eset` option. If `output_eset=TRUE` then `calc_prebs` returns ExpressionSet object (ExpressionSet object is defined in `affy` package). Otherwise, it returns a data frame containing PREBS values.

For running `calc_prebs` with custom CDF, the custom CDF package has to be downloaded and installed from Custom CDF website: http://brainarray.mbni.med.umich.edu/CustomCDF

For running `calc_prebs` with manufacturer's CDF, the manufacturer's CDF package can be installed from Bioconductor, for example: BiocManager::install("GenomicRanges"); BiocManager::install("hgu133plus2cdf")

For a detailed input specification, please refer to the `prebs` vignette.

Value

ExpressionSet object or a data frame containing PREBS values

Examples

```
if (require(prebsdata)) {
  # Get full paths to data files in \code{prebsdata} package
 bam_file1 <- system.file(file.path("sample_bam_files", "input1.bam"), package="prebsdata")
 bam_file2 <- system.file(file.path("sample_bam_files", "input2.bam"), package="prebsdata")
  bam_files <- c(bam_file1, bam_file2)
 custom_cdf_mapping1 <- system.file(file.path("custom-cdf", "HGU133Plus2_Hs_ENSG_mapping.txt"),
                                    package="prebsdata")
 custom_cdf_mapping2 <- system.file(file.path("custom-cdf", "HGU133A2_Hs_ENSG_mapping.txt"),
                                    package="prebsdata")
 manufacturer_cdf_mapping <- system.file(file.path("manufacturer-cdf", "HGU133Plus2_mapping.txt"),
                                         package="prebsdata")
  if (interactive()) {
    # Run PREBS using custom CDF without parallelization ("rpa" mode)
    prebs_values <- calc_prebs(bam_files, custom_cdf_mapping1)
    head(exprs(prebs_values))

    # Run PREBS using custom CDF without parallelization ("rma" mode)
    prebs_values <- calc_prebs(bam_files, custom_cdf_mapping1, sum.method="rma")
    head(exprs(prebs_values))

    # Run PREBS using custom CDF with parallelization
    library(parallel)
    N_CORES = 2
    CLUSTER <- makeCluster(N_CORES)
    prebs_values <- calc_prebs(bam_files, custom_cdf_mapping1, cluster=CLUSTER)
    stopCluster(CLUSTER)

    # Run PREBS using another custom CDF
    prebs_values <- calc_prebs(bam_files, custom_cdf_mapping2)

    # Run PREBS and return data frame instead of ExpressionSet object
    prebs_values <- calc_prebs(bam_files, custom_cdf_mapping1, output_eset=FALSE)
    head(prebs_values)
  }

  # Run PREBS using Manufacturer's CDF (outputs probe set expressions)
```

```
  prebs_values <- calc_prebs(bam_files, manufacturer_cdf_mapping)
  head(exprs(prebs_values))

  # Same as above, but state CDF package name explicitly
 prebs_values <- calc_prebs(bam_files, manufacturer_cdf_mapping, cdf_name="hgu133plus2cdf")
}
```

---

prebs *PREBS package*

Description

The prebs package aims at making RNA-sequencing (RNA-seq) data more comparable to microarray data. The comparability is achieved by summarizing sequencing-based expressions of probe regions using standard microarray summarization algorithms (RPA or RMA). The pipeline takes mapped reads in BAM format as an input and produces either gene expressions or original microarray probe set expressions as an output.

Details

The package has only one public function: `calc_prebs`. Type help(calc_prebs) for more information on the usage.

Index

calc_prebs, 2

prebs, 4
prebs-package (prebs), 4