Package 'metabinR'

November 12, 2026

Type Package

Title Abundance and Compositional Based Binning of Metagenomes

Version 2.0.0

biocViews Classification, Clustering, Microbiome, Sequencing, Software

Description Provide functions for performing abundance and compositional based binning on metagenomic samples, directly from FASTA or FASTQ files. Functions are implemented in Java and called via rJava.
Parallel implementation that operates directly on input FASTA/FASTQ files for fast execution. Inputs may be file paths or Biostrings/ShortRead sequence objects; results are returned as a MetabinResult S4 object wrapping cluster assignments, algorithm parameters, and input metadata.

License GPL-3

Encoding UTF-8

Language en-US

LazyData false

Depends R (>= 4.4), methods

Imports BiocGenerics, BiocParallel, Biostrings, checkmate, cli, rJava, S4Vectors, ShortRead, utils

SystemRequirements Java (>= 11)

RoxygenNote 7.3.3

URL https://github.com/gkanogiannis/metabinR

BugReports https://github.com/gkanogiannis/metabinR/issues

Suggests BiocStyle, covr, cvms, data.table, dplyr, ggplot2, gridExtra, knitr, R.utils, rmarkdown, sabre, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

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

git_branch RELEASE_3_23

git_last_commit 0524acb

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Anestis Gkanogiannis [aut, cre] (ORCID:
<https://orcid.org/0000-0002-6441-0688>)

Maintainer Anestis Gkanogiannis <anestis@gkanogiannis.com>

Contents

metabinR-package 2
abundance_based_binning 3
as.data.frame,MetabinResult-method 4
composition_based_binning 5
hierarchical_binning 6
MetabinResult-accessors 8
MetabinResult-class 9
metabinR_jvm_options 10

Index **11**

`metabinR-package` *metabinR: Abundance and Compositional Based Binning of Metagenomes*

Description

Provide functions for performing abundance and compositional based binning on metagenomic samples, directly from FASTA or FASTQ files. Functions are implemented in Java and called via rJava. Parallel implementation that operates directly on input FASTA/FASTQ files for fast execution. Inputs may be file paths or Biostrings/ShortRead sequence objects; results are returned as a MetabinResult S4 object wrapping cluster assignments, algorithm parameters, and input metadata.

Author(s)

Maintainer: Anestis Gkanogiannis <anestis@gkanogiannis.com> (ORCID)

See Also

Useful links:

- https://github.com/gkanogiannis/metabinR
- Report bugs at https://github.com/gkanogiannis/metabinR/issues

abundance_based_binning *Abundance based binning on metagenomic samples*

Description

This function performs abundance based binning on metagenomic samples, directly from FASTA or FASTQ files, by long kmer analysis (k>8). See doi:10.1186/s1285901611863 for more details.

Usage

```
abundance_based_binning(
  ...,
  eMin = 1,
  eMax = 0,
  kMerSizeAB = 10,
  numOfClustersAB = 3,
  outputAB = "AB.cluster",
  keepQuality = FALSE,
  dryRun = FALSE,
  gzip = FALSE,
  numOfThreads = BiocParallel::bpworkers()
)
```

Arguments

...	Input sequences. Either character paths to FASTA/FASTQ files (uncompressed or gzip compressed), a DNAStringSet / QualityScaledDNAStringSet, or a ShortReadQ object. Non-file inputs are staged to a temporary FASTA/FASTQ file for the Java backend.
eMin	Exclude kmers of less or equal count.
eMax	Exclude kmers of more or equal count.
kMerSizeAB	kmer length for Abundance based Binning.
numOfClustersAB	Number of Clusters for Abundance based Binning.
outputAB	Output Abundance based Binning Clusters files location and prefix.
keepQuality	Keep fastq qualities on the output files. (will produce .fastq)
dryRun	Don't write any output files.
gzip	Gzip output files.
numOfThreads	Number of threads to use. Defaults to bpworkers().

Value

A MetabinResult object. Its `assignments` slot is a `DataFrame` with `numOfClustersAB + 2` columns:

- `read_id` : read identifier from fasta header
- `AB` : read was assigned to this AB cluster index
- `AB.n` : read to cluster AB.n distance

For backwards-compatible `data.frame` output use `as.data.frame(result)`.

Author(s)

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

References

https://github.com/gkanogiannis/metabinR

Examples

```
res <- abundance_based_binning(
    system.file("extdata", "reads.metagenome.fasta.gz", package = "metabinR"),
    dryRun = TRUE, kMerSizeAB = 8
)
res
head(as.data.frame(res))
```

---

as.data.frame,MetabinResult-method

Convert a MetabinResult to a `data.frame`

Description

Preserves backwards compatibility with the `data.frame` return of metabinR <= 1.x.

Usage

```
## S4 method for signature 'MetabinResult'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A MetabinResult.
`row.names, optional, ...`	Unused; retained for S3 signature compatibility.

Value

A base `data.frame` of the assignments.

`composition_based_binning` *Composition based binning on metagenomic samples*

Description

This function performs composition based binning on metagenomic samples, directly from FASTA or FASTQ files, by short kmer analysis (k<8). See doi:10.1186/s1285901611863 for more details.

Usage

```
composition_based_binning(
  ...,
  kMerSizeCB = 4,
  numOfClustersCB = 5,
  outputCB = "CB.cluster",
  keepQuality = FALSE,
  dryRun = FALSE,
  gzip = FALSE,
  numOfThreads = BiocParallel::bpworkers()
)
```

Arguments

`...`	Input sequences. Either character paths to FASTA/FASTQ files (uncompressed or gzip compressed), a `DNAStringSet` / `QualityScaledDNAStringSet`, or a `ShortReadQ` object. Non-file inputs are staged to a temporary FASTA/FASTQ file for the Java backend.
`kMerSizeCB`	kmer length for Composition based Binning.
`numOfClustersCB`	Number of Clusters for Composition based Binning.
`outputCB`	Output Composition based Binning Clusters files location and prefix.
`keepQuality`	Keep fastq qualities on the output files. (will produce .fastq)
`dryRun`	Don't write any output files.
`gzip`	Gzip output files.
`numOfThreads`	Number of threads to use. Defaults to `bpworkers()`.

Value

A MetabinResult object. Its `assignments` slot is a `DataFrame` with `numOfClustersCB + 2` columns:

- `read_id` : read identifier from fasta header
- `CB` : read was assigned to this CB cluster index
- `CB.n` : read to cluster CB.n distance

For backwards-compatible `data.frame` output use `as.data.frame(result)`.

Author(s)

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

References

https://github.com/gkanogiannis/metabinR

Examples

```
res <- composition_based_binning(
    system.file("extdata", "reads.metagenome.fasta.gz", package = "metabinR"),
    dryRun = TRUE, kMerSizeCB = 2
)
res
```

---

hierarchical_binning *Hierarchical (ABxCB) binning on metagenomic samples*

Description

This function performs hierarchical binning on metagenomic samples, directly from FASTA or FASTQ files. First it analyzes sequences by long kmer analysis (k>8), as in abundance_based_binning. Then for each AB bin, it guesses the number of composition bins in it and performs composition based binning by short kmer analysis (k<8), as in composition_based_binning. See doi:10.1186/s1285901611863 for more details.

Usage

```
hierarchical_binning(
  ...,
  eMin = 1,
  eMax = 0,
  kMerSizeAB = 10,
  kMerSizeCB = 4,
  genomeSize = 3e+06,
  numOfClustersAB = 3,
  outputC = "ABxCB.cluster",
  keepQuality = FALSE,
  dryRun = FALSE,
  gzip = FALSE,
  numOfThreads = BiocParallel::bpworkers()
)
```

Arguments

...	Input sequences. Either character paths to FASTA/FASTQ files (uncompressed or gzip compressed), a DNAStringSet / QualityScaledDNAStringSet, or a ShortReadQ object. Non-file inputs are staged to a temporary FASTA/FASTQ file for the Java backend.
eMin	Exclude kmers of less or equal count.
eMax	Exclude kmers of more or equal count.
kMerSizeAB	kmer length for Abundance based Binning.
kMerSizeCB	kmer length for Composition based Binning.
genomeSize	Average genome size of taxa in the metagenome data.
numOfClustersAB	Number of Clusters for Abundance based Binning.
outputC	Output Hierarchical Binning (ABxCB) Clusters files location and prefix.
keepQuality	Keep fastq qualities on the output files. (will produce .fastq)
dryRun	Don't write any output files.
gzip	Gzip output files.
numOfThreads	Number of threads to use. Defaults to bpworkers().

Value

A MetabinResult object. Its assignments slot is a DataFrame:

- read_id : read identifier from fasta header
- ABxCB : read was assigned to this ABxCB cluster index
- ABxCB.n : read to cluster ABxCB.n distance

For backwards-compatible data.frame output use as.data.frame(result).

Author(s)

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

References

https://github.com/gkanogiannis/metabinR

Examples

```
res <- hierarchical_binning(
    system.file("extdata", "reads.metagenome.fasta.gz", package = "metabinR"),
    dryRun = TRUE, kMerSizeAB = 4, kMerSizeCB = 2
)
res
```

`MetabinResult-accessors` *Accessors for MetabinResult*

Description

Accessors for MetabinResult

Usage

```
assignments(x)

nClusters(x)

parameters(x)

algorithm(x)

## S4 method for signature 'MetabinResult'
assignments(x)

## S4 method for signature 'MetabinResult'
parameters(x)

## S4 method for signature 'MetabinResult'
algorithm(x)

## S4 method for signature 'MetabinResult'
nClusters(x)
```

Arguments

x	A MetabinResult object.

Value

`assignments()` returns a `DataFrame` with the per-read cluster assignments and distances. `nClusters()` returns an integer scalar: the number of clusters inferred by the algorithm. `parameters()` returns the list of arguments passed to the binning function. `algorithm()` returns the algorithm tag (`"AB"`, `"CB"`, or `"ABxCB"`).

Examples

```
res <- abundance_based_binning(
    system.file("extdata", "reads.metagenome.fasta.gz", package = "metabinR"),
    dryRun = TRUE, kMerSizeAB = 4, numOfClustersAB = 2
)
assignments(res)
```

```
nClusters(res)
algorithm(res)
parameters(res)$kMerSizeAB
```

`MetabinResult-class` *MetabinResult: binning result container*

Description

An S4 class returned by [abundance_based_binning()], [composition_based_binning()] and [hierarchical_binning()].

Usage

```
## S4 method for signature 'MetabinResult'
show(object)
```

Arguments

`object`	A MetabinResult.

Value

Objects of this class are returned by the binning functions. Use `assignments`, `nClusters`, `parameters`, `algorithm`, or `as.data.frame()` to access the results.

Slots

`assignments` A `DataFrame` of cluster assignments. The first column is `read_id`; subsequent columns are algorithm-specific (see the corresponding binning function).

`parameters` Named list of the parameters passed to the algorithm.

`inputs` Character vector of input file paths that were processed.

`algorithm` Character scalar: one of `"AB"`, `"CB"`, `"ABxCB"`.

Examples

```
res <- abundance_based_binning(
    system.file("extdata", "reads.metagenome.fasta.gz", package = "metabinR"),
    dryRun = TRUE, kMerSizeAB = 4, numOfClustersAB = 2
)
res
is(res, "MetabinResult")
```

`metabinR_jvm_options` *JVM options used when metabinR loads*

Description

Returns the vector of JVM flags passed to `.jpackage` on package load. Set `options(metabinR.jvm.flags = c(...))` before loading the package to override; set `options(java.parameters = ...)` to prepend heap-size flags (e.g. `"-Xmx4g"`) in the usual rJava way.

Usage

```
metabinR_jvm_options()
```

Value

A character vector of JVM flags.

Examples

```
metabinR_jvm_options()
```

Index

∗ **internal**
    metabinR-package, 2
.jpackage, *10*

abundance_based_binning, 3, *6*
algorithm, *9*
algorithm *(*MetabinResult-accessors*)*, 8
algorithm,MetabinResult-method
        *(*MetabinResult-accessors*)*, 8
as.data.frame,MetabinResult-method, 4
assignments, *9*
assignments *(*MetabinResult-accessors*)*, 8
assignments,MetabinResult-method
        *(*MetabinResult-accessors*)*, 8

bpworkers, *3*, *5*, *7*

composition_based_binning, 5, *6*

DataFrame, *4, 5*, *7–9*
DNAStringSet, *3*, *5*, *7*

hierarchical_binning, 6

metabinR *(*metabinR-package*)*, 2
metabinR-package, 2
metabinR_jvm_options, 10
MetabinResult, *4, 5*, *7–9*
MetabinResult *(*MetabinResult-class*)*, 9
MetabinResult-accessors, 8
MetabinResult-class, 9

nClusters, *9*
nClusters *(*MetabinResult-accessors*)*, 8
nClusters,MetabinResult-method
        *(*MetabinResult-accessors*)*, 8

parameters, *9*
parameters *(*MetabinResult-accessors*)*, 8
parameters,MetabinResult-method
        *(*MetabinResult-accessors*)*, 8

QualityScaledDNAStringSet, *3*, *5*, *7*

ShortReadQ, *3*, *5*, *7*
show,MetabinResult-method
        *(*MetabinResult-class*)*, 9