

# Package ‘metabinR’

May 2, 2024

**Type** Package

**Title** Abundance and Compositional Based Binning of Metagenomes

**Version** 1.7.0

**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.

**License** GPL-3

**Encoding** UTF-8

**Language** en-US

**LazyData** false

**Depends** R (>= 4.3)

**Imports** methods, rJava

**SystemRequirements** Java (>= 8)

**RoxygenNote** 7.3.1

**URL** <https://github.com/gkanogiannis/metabinR>

**BugReports** <https://github.com/gkanogiannis/metabinR/issues>

**Suggests** BiocStyle, 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** devel

**git\_last\_commit** 611ad69

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.20

**Date/Publication** 2024-05-01

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

**Maintainer** Anestis Gkanogiannis <anestis@gkanogiannis.com>

## Contents

metabinR-package	2
abundance_based_binning	3
composition_based_binning	4
hierarchical_binning	5
<b>Index</b>	<b>7</b>

---

metabinR-package	<i>metabinR: Abundance and Compositional Based Binning of Metagenomes</i>
------------------	---

---

## 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.

## Author(s)

**Maintainer:** Anestis Gkanogiannis <anestis@gkanogiannis.com> ([ORCID](https://orcid.org/0000-0002-6441-0688))

## 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](https://doi.org/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 = 1
)
```

## Arguments

...	Input fasta/fastq files locations (uncompressed or gzip compressed).
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.

## Value

A [data.frame](#) of the binning assignments. Return value contains 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

**Author(s)**

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

**References**

<https://github.com/gkanogiannis/metabinR>

**Examples**

```
abundance_based_binning(
  system.file("extdata", "reads.metagenome.fasta.gz", package = "metabinR"),
  dryRun = TRUE, kMerSizeAB = 8
)
```

---

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](https://doi.org/10.1186/s1285901611863) for more details.

**Usage**

```
composition_based_binning(
  ...,
  kMerSizeCB = 4,
  numOfClustersCB = 5,
  outputCB = "CB.cluster",
  keepQuality = FALSE,
  dryRun = FALSE,
  gzip = FALSE,
  numOfThreads = 1
)
```

**Arguments**

...	Input fasta/fastq files locations (uncompressed or gzip compressed).
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.

**Value**

A `data.frame` of the binning assignments. Return value contains `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

**Author(s)**

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

**References**

<https://github.com/gkanogiannis/metabinR>

**Examples**

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

---

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](https://doi.org/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 = 1  
)
```

**Arguments**

...	Input fasta/fastq files locations (uncompressed or gzip compressed).
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.

**Value**

A [data.frame](#) of the binning assignments. Return value contains numOfClustersAB + 2 columns.

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

**Author(s)**

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

**References**

<https://github.com/gkanogiannis/metabinR>

**Examples**

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

# Index

\* **internal**

metabinR-package, [2](#)

abundance\_based\_binning, [3](#), [5](#)

composition\_based\_binning, [4](#), [5](#)

data.frame, [3](#), [5](#), [6](#)

hierarchical\_binning, [5](#)

metabinR (metabinR-package), [2](#)

metabinR-package, [2](#)