

# Package ‘MicrobiomeBenchmarkData’

February 18, 2025

**Title** Datasets for benchmarking in microbiome research

**Version** 1.9.0

**Description** The MicrobiomeBenchmarkData package provides functionality to access microbiome datasets suitable for benchmarking. These datasets have some biological truth, which allows to have expected results for comparison. The datasets come from various published sources and are provided as TreeSummarizedExperiment objects. Currently, only datasets suitable for benchmarking differential abundance methods are available.

**License** Artistic-2.0

**LazyData** false

**Depends** R (>= 4.2), SummarizedExperiment, TreeSummarizedExperiment

**Imports** BiocFileCache, tools, S4Vectors, ape, utils

**Suggests** rmarkdown, knitr, BiocStyle, testthat (>= 3.0.0), mia, ggplot2, tidyr, dplyr, magrittr, tibble, purrr

**biocViews** ExperimentData, MicrobiomeData, ReproducibleResearch, SequencingData

**BugReports** <https://github.com/waldronlab/MicrobiomeBenchmarkData/issues>

**URL** <https://github.com/waldronlab/MicrobiomeBenchmarkData>,  
<http://waldronlab.io/MicrobiomeBenchmarkData/>

**BiocType** ExperimentData

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/MicrobiomeBenchmarkData>

**git\_branch** devel

**git\_last\_commit** d9e18a4

**git\_last\_commit\_date** 2024-10-29**Repository** Bioconductor 3.21**Date/Publication** 2025-02-18**Author** Samuel Gamboa [aut, cre] (ORCID:[<https://orcid.org/0000-0002-6863-7943>](https://orcid.org/0000-0002-6863-7943)),Levi Waldron [aut] (ORCID: [<https://orcid.org/0000-0003-2725-0694>](https://orcid.org/0000-0003-2725-0694)),

Marcel Ramos [ctb]

**Maintainer** Samuel Gamboa <Samuel.Gamboa.Tuz@gmail.com>

## Contents

<code>.assembleTreeSummarizedExperiment</code> . . . . .	2
<code>.getCache</code> . . . . .	3
<code>.getResourcePath</code> . . . . .	3
<code>.getSampleMetadata</code> . . . . .	4
<code>getBenchmarkData</code> . . . . .	4
<code>MicrobiomeBenchmarkData</code> . . . . .	5
<code>removeCache</code> . . . . .	6
<code>sampleMetadata</code> . . . . .	6
<code>scml</code> . . . . .	7
<b>Index</b>	<b>8</b>

---

`.assembleTreeSummarizedExperiment`*Assemble TreeSummarizedExperiment*

---

## Description

`.assembleTreeSummarizedExperiment` assembles a `TreeSummarizedDataset` taking as input the name of the dataset and the URL. This is a helper function for the `getBenchmarkData` function.

## Usage

```
.assembleTreeSummarizedExperiment(x)
```

## Arguments

<code>dat_name</code>	A character string with the name of the dataset.
<code>dat_url</code>	A character string with the URL from Zenodo.

## Value

A `TreeSummarizedExperiment`

---

.getCache                      *Get cache*

---

### **Description**

.getCache creates or loads a cache to store files downloaded through the MicrobiomeBenchmarkData package.

### **Usage**

```
.getCache()
```

### **Value**

A BiocFileCache object.

---

.getResourcePath              *Get resource path*

---

### **Description**

.getResource downloads the count matrix and store it in the cache.

### **Usage**

```
.getResourcePath(resource, suffix)
```

### **Arguments**

resource\_name    A character string with the name of the dataset.  
resource\_url     A character string with the URL from Zenodo.

### **Value**

A character string containing the path to the count matrix in the cache.

---

`.getSampleMetadata`      *Get sample metadata*

---

### Description

`.getSampleMetadata` returns `sampleMetadata`.

### Usage

```
.getSampleMetadata()
```

### Value

A data frame with sample metadata.

---

`getBenchmarkData`      *Get dataset*

---

### Description

`getBenchmarkData` imports datasets as `TreeSummarizedExperiment` objects.

### Usage

```
getBenchmarkData(x, dryrun = TRUE)
```

### Arguments

<code>x</code>	A character vector with the name(s) of the dataset(s). If empty and <code>dryrun = TRUE</code> , it returns a message with the names of the available datasets. If empty and <code>dryrun = FALSE</code> , it returns a list of <code>TreeSummarizedExperiments</code> with all of the datasets.
<code>dryrun</code>	If <code>TRUE</code> , only returns a message and invisibly returns the names of the datasets as a character vector. If <code>FALSE</code> , it returns the <code>TreeSummarizedExperiment</code> datasets indicated in the argument <code>'x'</code> .

### Value

A list of `TreeSummarizedExperiments` when `dryrun = FALSE`. A data frame with the datasets characteristics when `dryrun = TRUE`.

## Examples

```
## Example 1
datasets_names <- getBenchmarkData()
datasets_names

## Example 2
dataset <- getBenchmarkData(
  "HMP_2012_16S_gingival_V35_subset", dryrun = FALSE
)
dataset[[1]]
```

---

MicrobiomeBenchmarkData

*MicrobiomeBenchmarkData*

---

## Description

The MicrobiomeBenchmarkData provide functions for accessing various microbiome datasets with biological ground truth.

## Author(s)

**Maintainer:** Samuel Gamboa <Samuel.Gamboa.Tuz@gmail.com> ([ORCID](#))

Authors:

- Levi Waldron ([ORCID](#))

Other contributors:

- Marcel Ramos [contributor]

## See Also

Useful links:

- <https://github.com/waldronlab/MicrobiomeBenchmarkData>
- <http://waldronlab.io/MicrobiomeBenchmarkData/>
- Report bugs at <https://github.com/waldronlab/MicrobiomeBenchmarkData/issues>

removeCache                      *Remove cache*

---

**Description**

removeCache removes all files saved in the cache.

**Usage**

```
removeCache(ask = interactive())
```

**Arguments**

ask                      If TRUE, a prompt will appear asking the user to confirm removal of cache. Default value is given by the interactive function.

**Value**

NULL The cache and all of its contents are removed.

**Examples**

```
## Remove cache  
removeCache()
```

---

sampleMetadata                      *sampleMetadata*

---

**Description**

A data frame with the combined metadata of all of the samples in the datasets provided through the MicrobiomeBenchmarkData package.

**Usage**

```
data("sampleMetadata", package = "MicrobiomeBenchMarkData")
```

**Format**

A data.frame.

---

`scml`*SCML: spike-in-based calibration to total microbial load*

---

**Description**

The `scml` function applies the spike-in-based calibration to total microbial load (SCML) method to

**Usage**

```
scml(tse, bac = c("s", "r", "a"))
```

**Arguments**

`tse` A `treeSummarizedExperiment` from the `getBenchmarkData` function.  
`bac` A character. One of the following options: `s` = *Salinibacter ruber* (AF323500), `r` = *Rhizobium radiobacter* (AB247615), `a`, = *Alicyclobacillus acidiphilus* (AB076660)

**Value**

A `TreeSummarizedExperiment` with SCML data instead of counts.

**Examples**

```
tse <- getBenchmarkData("Stammler_2016_16S_spikein", dryrun = FALSE)[[1]]  
tseSCML <- scml(tse, bac = "s")
```

# Index

- \* **datasets**
  - sampleMetadata, [6](#)
- \* **internal**
  - .assembleTreeSummarizedExperiment,  
[2](#)
  - .getCache, [3](#)
  - .getResourcePath, [3](#)
  - .getSampleMetadata, [4](#)
  - .assembleTreeSummarizedExperiment, [2](#)
  - .getCache, [3](#)
  - .getResourcePath, [3](#)
  - .getSampleMetadata, [4](#)
- getBenchmarkData, [2, 4](#)
- MicrobiomeBenchmarkData, [5](#)
- MicrobiomeBenchmarkData-package  
(MicrobiomeBenchmarkData), [5](#)
- removeCache, [6](#)
- sampleMetadata, [6](#)
- scml, [7](#)