Package 'ompBAM'

March 26, 2024

Title C++ Library for OpenMP-based multi-threaded sequential profiling of Binary Alignment Map (BAM) files

Version 1.6.0

Date 2022-4-21

Description This packages provides C++ header files for developers wishing to create R packages that processes BAM files. ompBAM automates file access, memory management, and handling of multiple threads 'behind the scenes', so developers can focus on creating domain-specific functionality. The included vignette contains detailed documentation of this API, including quick-start instructions to create a new ompBAM-based package, and step-by-step explanation of the functionality behind the example packaged included within ompBAM.

License MIT + file LICENSE

Imports utils, Rcpp, zlibbioc

Suggests RcppProgress, knitr, rmarkdown, roxygen2, devtools, usethis, desc, testthat (>= 3.0.0)

VignetteBuilder knitr

biocViews Alignment, DataImport, RNASeq, Software, Sequencing, Transcriptomics, SingleCell

Encoding UTF-8

RoxygenNote 7.1.2

URL https://github.com/alexchwong/ompBAM

BugReports https://support.bioconductor.org/

Config/testthat/edition 3

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

git_branch RELEASE_3_18

git_last_commit e27fbbb

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-03-25

Author Alex Chit Hei Wong [aut, cre, cph]

Maintainer Alex Chit Hei Wong <a.wong@centenary.org.au>

R topics documented:

`ompBAM-package` *ompBAM: C++ header library for parallel sequential reading of BAM files*

Description

Although various tools for multi-threaded BAM processing (samtools, sambamba) are available, currently there are none available to the R/Bioconductor environment. Parallelism in R is achieved using BiocParallel or related packages. Typical alternatives include processing multiple BAM files, each using a single core. Although easy to set up, memory-intensive applications limit such approaches. ompBAM is a header-only C++ library based on OpenMP, allowing developers to implement OpenMP-based parallelism for the sequential reading of BAM files. ompBAM makes it easy by handling all the parallel file-access and bgzf decompression, allowing developers to focus on multi-threaded handling of individual reads.

Details

For a quick start guide and a full list of functionality, refer to the vignette `vignette("ompBAM-API-Docs", package = "ompBAM")`

Author(s)

Alex Chit Hei Wong

`example_BAM` *Returns the path of a test BAM file*

Description

Two types of BAM files are available. "Unsorted" is an unsorted BAM file of paired RNA-seq of THP1 cell culture from Green et al (GSE130011, GSM3729250), truncated to the first 10k reads. "scRNAseq" is a single cell 10X chromium V2 BAM file (sorted). It is sample E3.5_Rep1 from Nowotschin et al (GSE123046 / GSM3494334), truncated to the first 50k reads.

Usage

```
example_BAM(dataset = c("Unsorted", "scRNAseq"))
```

Arguments

dataset	Returns the path to either the "Unsorted" or "scRNAseq" BAM.

Value

A file path to the specified BAM file.

Examples

```
example_BAM("Unsorted")
```

`install_ompBAM_example` *ompBAM Example*

Description

Installs the ompBAMExample package included with the ompBAM package

Usage

```
install_ompBAM_example()
```

Details

This function installs the ompBAMExample package located inside the 'examples' subfolder of ompBAM. It uses devtools::load_all() to simulate package creation. After running this function to compile ompBAMExample(), users can run the example functions contained therein.

Value

None.

Examples

```
# The directory containing the source code is given by the path here

print(system.file(file.path('examples', "ompBAMExample"),
    package = 'ompBAM'))

# Install the ompBAMExample package

install_ompBAM_example()
```

use_ompBAM *Sets up the package in the given directory to use ompBAM*

Description

This function creates a new package in the given directory (unless one already exists). It then sets up the requisite dependencies and 'Make' and 'configure' files so that the source code will successfully compile and link to the appropriate external libraries (OpenMP, zlib) required for ompBAM. All 3 major platforms (Linux, MacOS and Windows) are supported, although MacOS users must first install the required OpenMP libraries first. See details below.

Usage

```
use_ompBAM(path = ".")
```

Arguments

path	The path to the desired directory in which to set up a new R package. The directory name is assumed to be the package name.

Details

OpenMP is natively supported on Linux and Windows but not on MacOS. To compile ompBAM_based packages with OpenMP on MacOS, users should install the requisite OpenMP libraries. An easy way to do this would be to run: `brew install libomp`. For details, refer to this guide

Value

None

Examples

```
path <- file.path(tempdir(), "myPkgName")
use_ompBAM(path)
```

Index