Package 'FilterFFPE'

July 7, 2025

Type Package

Title FFPE Artificial Chimeric Read Filter for NGS data

Version 1.19.0

Description This package finds and filters artificial chimeric reads specifically generated in next-generation sequencing (NGS) process of formalin-fixed paraffin-embedded (FFPE) tissues. These artificial chimeric reads can lead to a large number of false positive structural variation (SV) calls. The required input is an indexed BAM file of a FFPE sample.

License LGPL-3

Encoding UTF-8

Imports foreach, doParallel, GenomicRanges, IRanges, Rsamtools, parallel, S4Vectors

Suggests BiocStyle

biocViews StructuralVariation, Sequencing, Alignment, QualityControl, Preprocessing

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

git_branch devel

git_last_commit dd5e1cd

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-07

Author Lanying Wei [aut, cre] (ORCID: <https://orcid.org/0000-0002-4281-8017>)

Maintainer Lanying Wei <lanying.wei@uni-muenster.de>

Contents

FilterFFPE-package *FFPE Artificial Chimeric Read Filter for NGS data*

Description

This package finds and filters artificial chimeric reads specifically generated in next-generation sequencing (NGS) process of formalin-fixed paraffin-embedded (FFPE) tissues. These artificial chimeric reads can lead to a large number of false positive structural variation (SV) calls. The required input is an indexed BAM file of a FFPE sample.

Details

This package was not yet installed at build time.
The next-generation sequencing (NGS) reads from formalin-fixed paraffin-embedded (FFPE) samples contain numerous artifact chimeric reads, which can lead to a large number of false positive structural variation (SV) calls. This package finds and filters these artifact chimeric reads from BAM files of FFPE samples to improve SV calling performance.

Index: This package was not yet installed at build time.
There are three available functions to find and/or filter artifact chimeric reads of FFPE samples:

1. `findArtifactChimericReads`: Find artifact chimeric reads in BAM file of FFPE sample.

2. `filterBamByReadNames`: Filter reads from BAM file by read names.

3. `FFPEReadFilter`: Find and filter artifact chimeric reads in BAM file of FFPE sample.

Author(s)

Lanying Wei <lanying.wei@uni-muenster.de>

See Also

`FilterFFPE`, `filterBamByReadNames`, `FFPEReadFilter`

Examples

```
file <- system.file("extdata", "example.bam", package = "FilterFFPE")
outFolder <- tempdir()
FFPEReadsFile <- paste0(outFolder, "/example.FFPEReads.txt")
dupChimFile <- paste0(outFolder, "/example.dupChim.txt")
destination <- paste0(outFolder, "/example.FilterFFPE.bam")
FFPEReadFilter(file = file, threads = 2, destination = destination,
               overwrite = TRUE, FFPEReadsFile = FFPEReadsFile,
               dupChimFile = dupChimFile)
```

FFPEReadFilter *Find and filter artifact chimeric reads in BAM file of FFPE sample*

Description

Artifact chimeric reads are enriched in NGS data of FFPE samples, these reads can lead to a large number of false positive SV calls. This function finds and filters these artifact chimeric reads.

Usage

```
FFPEReadFilter(file, maxReadsOfSameBreak=2, minMapBase=1, threads=1,
index=file, destination=sub("\\.bam(\\.gz)?", ".FilterFFPE.bam", file),
overwrite=FALSE, FFPEReadsFile=sub("\\.bam(\\.gz)?", ".FFPEReads.txt", file),
dupChimFile=sub("\\.bam(\\.gz)?", ".dupChim.txt", file), filterdupChim=TRUE)
```

Arguments

file	Path to the BAM file.
maxReadsOfSameBreak	The maximum allowed number of artifact chimeric reads sharing a false positive breakpoint. If the number of reads sharing the same breakpoint exceeds this number, these reads are not recognized as artifact chimeric reads. Reads marked as PCR or optical duplicates are excluded from the calculation. For paired-end sequencing, a read pair of artifact chimeric fragments may both contain the artifact breakpoints; thereby, the defalut is set to 2.
minMapBase	The minimum required length (bp) of a short complementary mapping for an artifact chimeric read. Artifact chimeric reads are derived from the combination of two single-stranded DNA fragments linked by short reverse complementary regions (SRCR). Reads with SRCR shorter than this length are not recognized as artifact chimeric reads. Note: sequence errors and mutations might influence the detection of the existence and length of SRCR. Suggested range: 0-3. When it is set to 0 or any value below 1, this step will be skipped.
threads	Number of threads to use. Multi-threading can speed up the process.
index	Path of the index file of the input BAM file.
destination	Path of the output filtered BAM file.
overwrite	Boolean value indicating whether the destination can be over-written if it already exists.
FFPEReadsFile	Path of the output txt file with artifact chimeric read names.
dupChimFile	Path of the output txt file with supplementary reads that are marked as PCR or optical duplicates.
filterdupChim	Filter PCR or optical duplicates of all chimeric reads when set to true. These reads may contain duplicates of artifact chimeric reads; therefore, it is recommended to also remove these reads.

Details

The next-generation sequencing (NGS) reads from formalin-fixed paraffin-embedded (FFPE) samples contain numerous artifact chimeric reads, which can lead to a large number of false positive structural variation (SV) calls. This function finds and filters these artifact chimeric reads. An index file is also generated for the created filtered BAM file.

Value

The file name of the created destination file.

Author(s)

Lanying Wei <lanying.wei@uni-muenster.de>

See Also

FilterFFPE, findArtifactChimericReads, filterBamByReadNames

Examples

```
file <- system.file("extdata", "example.bam", package = "FilterFFPE")
outFolder <- tempdir()
FFPEReadsFile <- paste0(outFolder, "/example.FFPEReads.txt")
dupChimFile <- paste0(outFolder, "/example.dupChim.txt")
destination <- paste0(outFolder, "/example.FilterFFPE.bam")
FFPEReadFilter(file = file, threads = 2, destination = destination,
               overwrite = TRUE, FFPEReadsFile = FFPEReadsFile,
               dupChimFile = dupChimFile)
```

---

filterBamByReadNames *Filter reads from BAM file by read names*

Description

Generate filtered BAM file that does not contain reads with the input read names.

Usage

```
filterBamByReadNames(file, readsToFilter, index=file,
destination=sub("\\.bam(\\.gz)?", ".FilterFFPE.bam", file), overwrite=FALSE)
```

Arguments

file	Path to the input BAM file.
readsToFilter	A character vector of read names to filter.
index	Path of the index file of the input BAM file.
destination	Path of the output filtered BAM file.
overwrite	Boolean value indicating whether the destination can be over-written if it already exists.

Details

Generate filtered BAM file that does not contain reads with the input read names, index file is also created.

Value

The file name of the created destination file.

Author(s)

Lanying Wei <lanying.wei@uni-muenster.de>

See Also

FilterFFPE, findArtifactChimericReads, FFPEReadFilter

Examples

```
file <- system.file("extdata", "example.bam", package = "FilterFFPE")
outFolder <- tempdir()
FFPEReadsFile <- paste0(outFolder, "/example.FFPEReads.txt")
dupChimFile <- paste0(outFolder, "/example.dupChim.txt")
destination <- paste0(outFolder, "/example.FilterFFPE.bam")
artifactReads <- findArtifactChimericReads(file = file, threads = 2,
                                           FFPEReadsFile = FFPEReadsFile,
                                           dupChimFile = dupChimFile)
dupChim <- readLines(dupChimFile)
readsToFilter <- c(artifactReads, dupChim)
filterBamByReadNames(file = file, readsToFilter = readsToFilter,
                     destination = destination, overwrite=TRUE)
```

---

findArtifactChimericReads

Find artifact chimeric reads in BAM file of FFPE sample

Description

Artifact chimeric reads are enriched in NGS data of FFPE samples, these reads can lead to a large number of false positive SV calls. This function finds these artifact chimeric reads.

Usage

```
findArtifactChimericReads(file, maxReadsOfSameBreak=2, minMapBase=1,
threads=1, FFPEReadsFile=sub("\\.bam(\\.gz)?", ".FFPEReads.txt", file),
dupChimFile=sub("\\.bam(\\.gz)?", ".dupChim.txt", file))
```

Arguments

file	Path to the BAM file.
maxReadsOfSameBreak	The maximum allowed number of artifact chimeric reads sharing a false positive breakpoint. If the number of reads sharing the same breakpoint exceeds this number, these reads are not recognized as artifact chimeric reads. Reads marked as PCR or optical duplicates are excluded from the calculation. For paired-end sequencing, a read pair of artifact chimeric fragments may both contain the artifact breakpoints; thereby, the defalut is set to 2.
minMapBase	The minimum required length (bp) of a short complementary mapping for an artifact chimeric read. Artifact chimeric reads are derived from the combination of two single-stranded DNA fragments linked by short reverse complementary regions (SRCR). Reads with SRCR shorter than this length are not recognized as artifact chimeric reads. Note: sequence errors and mutations might influence

	the detection of the existence and length of SRCR. Suggested range: 0-3. When it is set to 0 or any value below 1, this step will be skipped.
`threads`	Number of threads to use. Multi-threading can speed up the process.
`FFPEReadsFile`	Path of the output txt file with artifact chimeric read names.
`dupChimFile`	Path of the output txt file with read names of PCR or optical duplicates of all chimeric reads.

Details

The next-generation sequencing (NGS) reads from formalin-fixed paraffin-embedded (FFPE) samples contain numerous artifact chimeric reads, which can lead to a large number of false positive structural variation (SV) calls. This function finds the read names of these artifact chimeric reads. To further filter these reads, `filterBamByReadNames` can be applied.

Value

A character vector of artifact chimeric read names.

Author(s)

Lanying Wei <lanying.wei@uni-muenster.de>

See Also

`FilterFFPE`, `filterBamByReadNames`, `FFPEReadFilter`

Examples

```
file <- system.file("extdata", "example.bam", package = "FilterFFPE")
outFolder <- tempdir()
FFPEReadsFile <- paste0(outFolder, "/example.FFPEReads.txt")
dupChimFile <- paste0(outFolder, "/example.dupChim.txt")
artifactReads <- findArtifactChimericReads(file = file, threads = 2,
                                           FFPEReadsFile = FFPEReadsFile,
                                           dupChimFile = dupChimFile)
head(artifactReads)
```

Index