

An Introduction to *FilterFFPE*

Lanying Wei

Modified: 20 August, 2020. Compiled: August 29, 2024

Contents

1	Introduction	1
2	Input	1
3	Artifact Chimeric Read Filtration	1
A	Session info	3

1 Introduction

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. The *FilterFFPE* package finds and filters these artifact chimeric reads from BAM files to improve SV calling performance in FFPE samples.

2 Input

The required input is an indexed BAM file of the FFPE sample, the PCR or optical duplicates should be marked or removed from the BAM file. If you plan to filter reads with mapping quality, or to only keep reads in targeted region, please do that after using *FilterFFPE*, as these steps may remove some of the alignments that is useful to the *FilterFFPE*'s filtering algorithm. Example of such a BAM file is stored in the 'extdata' directory of *FilterFFPE* package).

3 Artifact Chimeric Read Filtration

The filtration includes two steps: 1) Find artifact chimeric reads from BAM file . 2) Remove these artifact chimeric reads in the filtered BAM file. `findArtifactChimericReads` can be used to find artifact chimeric reads; read names of PCR or optical duplicates of all chimeric reads are also found and written in a txt file for filtration. `filterBamByReadNames` can be used for further filtration, it generates a filtered and indexed BAM file. `FFPereadFilter` combines these two functions.

```
> library(FilterFFPE)
> # Find artifact chimeric reads
```

An Introduction to *FilterFFPE*

```
> 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)

[1] "SRR1523265.24545253" "SRR1523265.31420529"
[3] "SRR1523265.38291385" "SRR1523265.49620943"
[5] "SRR1523265.5056364"  "SRR1523265.52887917"

>
```

```
> # Filter artifact chimeric reads and PCR or optical duplicates of chimeric reads
> dupChim <- readLines(dupChimFile)
> readsToFilter <- c(artifactReads, dupChim)
> destination <- paste0(outFolder, "/example.FilterFFPE.bam")
> filterBamByReadNames(file = file, readsToFilter = readsToFilter,
+                      destination = destination, overwrite=TRUE)

[1] "/tmp/RtmpbUSKBP/example.FilterFFPE.bam"

>
```

```
> # Perform finding and filtering with one function
> 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)

[1] "/tmp/RtmpbUSKBP/example.FilterFFPE.bam"

>
```

The generated BAM file can be loaded with `scanBam` function from *Rsamtools* package for further interrogation.

```
> # load Bam file with scanBAM
> newBam <- Rsamtools::scanBam(destination)
> head(newBam[[1]]$seq)

DNAStrngSet object of length 6:
  width seq
[1] 90 CAGCTGCTCAACCACCTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
[2] 90 CAGCTGCTCAACCACCTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
[3] 90 CAGCTGCTCAACCACCTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
[4] 90 CAGCTGCTCAACCACCTCCTCTCT...CCCTGGCCCTCCCAGCCCACGAT
[5] 90 ACCCCACTCCCTGGCCCTCCCAGC...CCTGAACCCCAGCCTGTGGTTC
[6] 90 CCCCCACTCCCTGGCCCTCCCAGC...CCTGAACCCCAGCCTGTGGTTC
```

A Session info

```
> packageDescription("FilterFFPE")
Package: FilterFFPE
Type: Package
Title: FFPE Artificial Chimeric Read Filter for NGS
      data
Version: 1.15.0
Authors@R: person("Lanying", "Wei",
  email="lanying.wei@uni-muenster.de", role =
  c("aut", "cre"), comment = c(ORCID =
  "0000-0002-4281-8017"))
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
Repository: https://bioc.r-universe.dev
RemoteUrl: https://github.com/bioc/FilterFFPE
RemoteRef: HEAD
RemoteSha: c749f1be346ee277b03594ed1ea72533823ccee6
Author: Lanying Wei [aut, cre]
  (<https://orcid.org/0000-0002-4281-8017>)
Maintainer: Lanying Wei <lanying.wei@uni-muenster.de>
Built: R 4.4.1; ; 2024-08-29 03:20:34 UTC; unix

-- File: /tmp/Rtmp8mozhp/Rinstea07f1ba9a5/FilterFFPE/Meta/package.rds
> sessionInfo()

R version 4.4.1 (2024-06-14)
Platform: x86_64-pc-linux-gnu
Running under: Ubuntu 24.04 LTS

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-p-r0.3.26.so; LAPACK version 3.12.0

locale:
  [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
```

```
[3] LC_TIME=en_US.UTF-8      LC_COLLATE=C
[5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8    LC_NAME=C
[9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

```
time zone: Etc/UTC
tzcode source: system (glibc)
```

```
attached base packages:
[1] stats      graphics  grDevices  utils      datasets
[6] methods    base
```

```
other attached packages:
[1] FilterFFPE_1.15.0
```

```
loaded via a namespace (and not attached):
[1] doParallel_1.0.17      crayon_1.5.3
[3] httr_1.4.7             cli_3.6.3
[5] knitr_1.48             rlang_1.1.4
[7] xfun_0.47             UCSC.utils_1.1.0
[9] jsonlite_1.8.8        S4Vectors_0.43.2
[11] buildtools_1.0.0      Biostrings_2.73.1
[13] BiocStyle_2.33.1     htmltools_0.5.8.1
[15] maketools_1.3.0       sys_3.4.2
[17] stats4_4.4.1          rmarkdown_2.28
[19] evaluate_0.24.0       bitops_1.0-8
[21] fastmap_1.2.0         foreach_1.5.2
[23] yaml_2.3.10           IRanges_2.39.2
[25] GenomeInfoDb_1.41.1   BiocManager_1.30.25
[27] compiler_4.4.1        codetools_0.2-20
[29] XVector_0.45.0        BiocParallel_1.39.0
[31] digest_0.6.37         R6_2.5.1
[33] GenomeInfoDbData_1.2.12 parallel_4.4.1
[35] GenomicRanges_1.57.1  tools_4.4.1
[37] iterators_1.0.14      Rsamtools_2.21.1
[39] zlibbioc_1.51.1       BiocGenerics_0.51.0
```