

# ALKfusiondiscover

<https://github.com/CTrierMaansson/ALKfusiondiscover>

Christoffer Trier Maansson

Emma Roger Andersen

19 jul 2022

## Abstract

Circulating tumor DNA (ctDNA) containing somatic mutations can be found in blood plasma. This includes DNA fusions, such as the EML4-ALK, which can be an oncogenic driver in non-small cell lung cancer. This is an introduction to the **ALKfusiondiscover** package for R, which can be used to evaluate whether EML4-ALK is present in blood plasma.

## Contents

|          |                        |          |
|----------|------------------------|----------|
| <b>1</b> | <b>Introduction</b>    | <b>2</b> |
| <b>2</b> | <b>Installation</b>    | <b>2</b> |
| <b>3</b> | <b>Package data</b>    | <b>2</b> |
| <b>4</b> | <b>Functions</b>       | <b>3</b> |
| 4.1      | EML4_ALK_detection()   | 3        |
| 4.2      | EML4_sequence()        | 4        |
| 4.3      | ALK_sequence()         | 4        |
| 4.4      | break_position()       | 5        |
| 4.5      | break_position_depth() | 6        |
| 4.6      | EML4_ALK_analysis()    | 6        |

# 1 Introduction

This package was created in order to increase the sensitivity of EML4-ALK detection from commercially available NGS products such as the AVENIO (Roche) pipeline.

Paired-end sequencing of cfDNA generated BAM files can be used as input to discover EML4-ALK variants. This package was developed using position deduplicated BAM files generated with the AVENIO Oncology Analysis Software. These files are made using the AVENIO ctDNA surveillance kit and Illumina Nextseq 500 sequencing. This is a targeted hybridization NGS approach and includes ALK-specific but not EML4-specific probes.

The package includes six functions.

The output of the first function, `EML4_ALK_detection()`, is used to determine whether EML4-ALK is detected and serves as input for the next four exploratory functions characterizing the EML4-ALK variant. The last function `EML4_ALK_analysis()` combines the output of the exploratory functions.

To serve as examples, this package includes BAM files representing the EML4-ALK positive cell line H3122 and the EML4-ALK negative cell line, HCC827.

# 2 Installation

Use `devtools` to install `ALKfusiondiscover`.

```
if (!require(devtools)) install.packages('devtools')
library(devtools)

install_github("CTrierMaansson/ALKfusiondiscover")
#>
#> * checking for file 'C:\Users\Christoffer\AppData\Local\Temp\RtmpUDPcVc\remotes59b86f879f6\CTrierMaansson\ALKfusiondiscover'
#> * preparing 'ALKfusiondiscover':
#> * checking DESCRIPTION meta-information ... OK
#> * checking for LF line-endings in source and make files and shell scripts
#> * checking for empty or unneeded directories
#> * building 'ALKfusiondiscover_0.0.1.1.tar.gz'
#>
library(ALKfusiondiscover)
```

# 3 Package data

BAM files from the cell lines, H3122 and HCC827, are included in the package and can be used as examples to explore the functions.

```
H3122_bam <- system.file("extdata", "H3122_EML4.bam", package = "ALKfusiondiscover")
HCC827_bam <- system.file("extdata", "HCC827_EML4.bam", package = "ALKfusiondiscover")
```

#### 4.1 EML4\_ALK\_detection()

**Input:**

The name of the file which the data are to be read from.

**character** representing the reference genome. Can either be “hg38” or “hg19”. Default = “hg38”.

**integer**, the minimum number EML4-ALK mate pairs needed to be detected in order to call a variant. Default = 2.

If EML4-ALK is detected, a `data.frame` with soft-clipped reads representing EML4-ALK is returned. Otherwise “No EML4-ALK was detected” is returned.

```
head(EML4_ALK_detection(file = H3122_bam, genome = "hg38", mates = 2))

#>
#> 1 TTGCTTCTTTCACTTAGTTTTTTTGTTTTGTTTGTTTGTTTGTTTTTGAGATGGGGTTTCACTCTTGTTGCCCAGGCTGGAGTGCAGTGGTCT
#> 2 TTGCTTCTTTCACTTAGTTTTTTTGTTTTGTTTGTTTGTTTGTTTTTGAGATGGGGTTTCACTCTTGTTGCCCAGGCTGGAGTGCAGTGGTCT
#> 3 TTGCTTCTTTCACTTAGTTTTTTTGTTTTGTTTGTTTGTTTGTTTTTGAGATGGGGTTTCACTCTTGTTGCCCAGGCTGGAGTGCAGTGGTCT
#> 4 TTGCTTCTTTCACTTAGTTTTTTTGTTTTGTTTGTTTGTTTGTTTTTGAGATGGGGTTTCACTCTTGTTGCCCAGGCTGGAGTGCAGTGGTCT
#> 5 TTGCTTCTTTCACTTAGTTTTTTTGTTTTGTTTGTTTGTTTGTTTTTGAGATGGGGTTTCACTCTTGTTGCCCAGGCTGGAGTGCAGTGGTCT
#> 6 TTGCTTCTTTCACTTAGTTTTTTTGTTTTGTTTGTTTGTTTGTTTTTGAGATGGGGTTTCACTCTTGTTGCCCAGGCTGGAGTGCAGTGGTCT

#>      mate position cigar
#> 1 29223691 42299657 94M2S
#> 2 29223375 42299657 94M2S
#> 3 29223479 42299657 94M2S
#> 4 29223686 42299657 94M2S
#> 5 29223636 42299657 94M2S
#> 6 29223687 42299657 94M2S

EML4_ALK_detection(file = HCC827_bam, genome = "hg38", mates = 2)
#> [1] "No EML4-ALK was detected"
```

## 4.2 EML4\_sequence()

This function identifies the basepairs leading up to the EML4 breakpoint.

### Input:

**reads**

data.frame returned by EML4\_ALK\_detection().

**basepairs**

integer, number of basepairs identified from the EML4-ALK fusion. Default = 20.

### Output:

If EML4-ALK is detected, a **table** of identified EML4 basepairs is returned, with the number of corresponding reads for each sequence. Otherwise “No EML4-ALK was detected” is returned.

### Examples:

```
EML4_sequence(EML4_ALK_detection(file = H3122_bam, genome = "hg38", mates = 2), basepairs = 20)
#> EML4_seq
#> CCAGGCTGGAGTGCAGTGGT GGAGTGCAGTGGTGTGATT TCAGGCTGGAGTGCAGTGGT
#>                201                1                1
EML4_sequence(EML4_ALK_detection(file = HCC827_bam, genome = "hg38", mates = 2), basepairs = 20)
#> [1] "No EML4-ALK was detected"
```

## 4.3 ALK\_sequence()

This function identifies the basepairs following the ALK breakpoint.

### Input:

**reads**

data.frame returned by EML4\_ALK\_detection().

**basepairs**

integer, number of basepairs identified from the EML4-ALK fusion. Default = 20.

### Output:

If EML4-ALK is detected, a **table** of identified ALK basepairs is returned, with the number of corresponding reads for each sequence. Otherwise “No EML4-ALK was detected” is returned.

### Examples:

```

ALK_sequence(EML4_ALK_detection(file = H3122_bam, genome = "hg38", mates = 2), basepairs = 20)
#> ALK_seq
#> CAGAATTTTAGCTTTGCAAT      CGGATTTTCTAGCTTT CGGATTTTCTAGCTTTTCATT
#>           1           1           2
#> CGGAATTTTAGCTTTGCATT      CT           CTG
#>           1           8           3
#>           CTGA           CTGAT           CTGATTTT
#>           11          16           5
#>           CTGATTTT      CTGATTTTCTA      CTGATTTTCTAG
#>           6           3           3
#> CTGATTTTCTAGATTTGCATT      CTGATTTTCTAGC      CTGATTTTCTAGCT
#>           1           14          10
#>           CTGATTTTCTAGCTT      CTGATTTTCTAGCTTT      CTGATTTTCTAGCTTTG
#>           10           3           4
#>           CTGATTTTCTAGCTTTGC      CTGATTTTCTAGCTTTGCA      CTGATTTTCTAGCTTTGCAT
#>           7           8           1
#> CTGATTTTCTAGCTTTGCATT CTGATTTTCTAGCTTTGCAAT      CTGATTTTCTAGCTTTT
#>           71           1           1
#> CTGATTTTCTAGCTTTTCATA      CTGATTTTCTAT      CTGATTTTCTATCTTTG
#>           1           1           2
#> CTGATTTTCTATCTTTGCATT CTGATTTTCTATCTTTTGATT CTGTGTTTCTAGATTTGCATT
#>           2           1           1
#> CTGTGTTTCTATCTTTGCAAT      CTGAA CTTATTTTCTATCTTTGCATT
#>           1           1           1
#>           TTAGCTTTG
#>           1
ALK_sequence(EML4_ALK_detection(file = HCC827_bam, genome = "hg38", mates = 2), basepairs = 20)
#> [1] "No EML4-ALK was detected"

```

#### 4.4 break\_position()

This function identifies the genomic position in EML4 where the breakpoint has happened.

**Input:**

**reads**

data.frame returned by EML4\_ALK\_detection().

**Output:**

If EML4-ALK is detected, a table of genomic positions is returned with the number of corresponding reads for each sequence. Otherwise “No EML4-ALK was detected” is returned.

**Examples:**

```

break_position(EML4_ALK_detection(file = H3122_bam, genome = "hg38", mates = 2))
#> break_pos
#> 42299750 42299757
#>      202      1
break_position(EML4_ALK_detection(file = HCC827_bam, genome = "hg38", mates = 2))
#> [1] "No EML4-ALK was detected"

```

## 4.5 break\_position\_depth()

This function identifies the read depth at the basepair before the breakpoint in EML4.

### Input:

#### file

The name of the file which the data are to be read from.

#### reads

data.frame returned by EML4\_ALK\_detection().

### Output:

If EML4-ALK is detected a single integer corresponding to the read depth at the breakpoint is returned. Otherwise “No EML4-ALK was detected” is returned

### Examples:

```
break_position_depth(H3122_bam, EML4_ALK_detection(file = H3122_bam, genome = "hg38", mates = 2))
#> [1] 251
break_position_depth(HCC827_bam, EML4_ALK_detection(file = HCC827_bam, genome = "hg38", mates = 2))
#> [1] "No EML4-ALK was detected"
```

## 4.6 EML4\_ALK\_analysis()

This functions collects the results from the other functions of the package.

### Input:

#### file

The name of the file which the data are to be read from.

#### genome

character representing the reference genome. Can be either “hg38” or “hg19”. Default = “hg38”.

#### mates

integer, the minimum number EML4-ALK mate pairs needed to be detected in order to call a variant. Default = 2.

**integer**, number of basepairs identified from the EML4-ALK fusion. Default = 20.

A `list` object with `clipped_reads` corresponding to `EML4_ALK_detection()`, `last_EML4` corresponding to `EML4_sequence()`, `first_ALK` corresponding to `ALK_sequence()`, `breakpoint` corresponding to `break_position()`, and `read_depth` corresponding to `break_position_depth()`.

```
H3122_results <- EML4_ALK_analysis(file = H3122_bam, genome = "hg38", mates = 2, basepairs = 20)
HCC827_results <- EML4_ALK_analysis(file = HCC827_bam, genome = "hg38", mates = 2, basepairs = 20)
```

```
H3122_results$last_EML4
#> EML4_seq
#> CCAGGCTGGAGTGCAGTGGT GGAGTGCAGTGGTGTGATTT TCAGGCTGGAGTGCAGTGGT
#>                201                      1                      1
```

```
H3122_results$first_ALK
#> ALK_seq
#> CAGAATTTTAGCTTTGCAAT          CGGATTTTtagctTTT  CGGATTTTtagctTTTTCATT
#>                                1                      1                      2
#> CGGAATTTTAGCTTTGCATT          CT                      CTG
#>                                1                      8                      3
#>                CTGA                CTGAT                CTGATTTT
#>                11                16                5
#>                CTGATTTTT          CTGATTTTTTA          CTGATTTTTTAG
#>                6                3                3
#> CTGATTTTTTAGATTTGCATT          CTGATTTTTTAGC          CTGATTTTTTAGCT
#>                1                14                10
#>                CTGATTTTTTAGCTT          CTGATTTTTTAGCTTT          CTGATTTTTTAGCTTTG
#>                10                3                4
#>                CTGATTTTTTAGCTTTGC          CTGATTTTTTAGCTTTGCA          CTGATTTTTTAGCTTTGCAT
#>                7                8                1
#> CTGATTTTTTAGCTTTGCATT          CTGATTTTTTAGCTTTGCAAT          CTGATTTTTTAGCTTTT
#>                71                1                1
```

```

#> CTGATTTTCTAGCTTTTCATA          CTGATTTTCTAT          CTGATTTTCTATCTTTG
#>                                1          1          2
#> CTGATTTTCTATCTTTGCATT CTGATTTTCTATCTTTGATT CTGTGTTTCTAGATTGCATT
#>                                2          1          1
#> CTGTCTTTTCTATCTTTGCAAT          CTGAA CTTATTTTCTATCTTTGCATT
#>                                1          1          1
#>          TTAGCTTTG
#>          1

```

```

H3122_results$breakpoint

```

```

#> break_pos
#> 42299750 42299757
#>      202      1

```

```

H3122_results$read_depth

```

```

#> [1] 251

```

```

HCC827_results

```

```

#> [1] "No EML4-ALK was detected"

```