

# Package ‘SeqKat’

May 7, 2026

**Type** Package

**Title** Detection of Kataegis

**Version** 0.0.9

**URL** <https://github.com/TheBoutrosLab/package-SeqKat>

**BugReports** <https://github.com/TheBoutrosLab/package-SeqKat/issues>

**Description** Kataegis is a localized hypermutation occurring when a region is enriched in somatic SNVs. Kataegis can result from multiple cytosine deaminations catalyzed by the AID/APOBEC family of proteins. This package contains functions to detect kataegis from SNVs in BED format. This package reports two scores per kataegic event, a hypermutation score and an APOBEC mediated kataegic score. Yousif, F. et al.; The Origins and Consequences of Localized and Global Somatic Hypermutation; Biorxiv 2018 <[doi:10.1101/287839](https://doi.org/10.1101/287839)>.

**Depends** R (>= 2.15.1), foreach, doParallel

**Imports** Rcpp (>= 0.11.0)

**LinkingTo** Rcpp

**Suggests** testthat, doMC, rmarkdown, knitr

**License** GPL (>= 2)

**RoxygenNote** 6.0.1

**VignetteBuilder** rmarkdown, knitr

**NeedsCompilation** yes

**Author** Fouad Yousif [aut],  
Xihui Lin [aut],  
Fan Fan [aut],  
Christopher Lalansingh [aut],  
John Macdonald [aut],  
Paul C. Boutros [cre]

**Maintainer** Paul C. Boutros <[pboutros@sbgdiscover.org](mailto:pboutros@sbgdiscover.org)>

**Repository** CRAN

**Date/Publication** 2026-01-24 00:20:01 UTC

## Contents

combine.table . . . . .	2
final.score . . . . .	3
get.context . . . . .	4
get.exprobntcx . . . . .	5
get.nucleotide.chunk.counts . . . . .	6
get.pair . . . . .	7
get.tn . . . . .	7
get.toptn . . . . .	8
get.trinucleotide.counts . . . . .	9
seqkat . . . . .	10
test.kataegis . . . . .	11
<b>Index</b>	<b>13</b>

---

combine.table	<i>Combine Table</i>
---------------	----------------------

---

## Description

Merges overlapped windows to identify genomic boundaries of kataegic events. This function also assigns hypermutation and kataegic score for combined windows

## Usage

```
combine.table(test.table, somatic, mutdistance, segnum, output.name)
```

## Arguments

test.table	Data frame of kataegis test scores
somatic	Data frame of somatic variants
mutdistance	The maximum intermutational distance allowed for SNVs to be grouped in the same kataegic event. Recommended value: 3.2
segnum	Minimum mutation count. The minimum number of mutations required within a cluster to be identified as kataegic. Recommended value: 4
output.name	Name of the generated output directory.

## Author(s)

Fouad Yousif

Fan Fan

**Examples**

```
load(
  paste0(
    path.package("SeqKat"),
    "/extdata/test/somatic.rda"
  )
);

load(
  paste0(
    path.package("SeqKat"),
    "/extdata/test/final.score.rda"
  )
);

combine.table(
  final.score,
  somatic,
  3.2,
  4,
  tempdir()
);
```

---

final.score	<i>Final Score</i>
-------------	--------------------

---

**Description**

Assigns hypermutation score (hm.score) and kataegic score (k.score)

**Usage**

```
final.score(test.table, cutoff, somatic, output.name)
```

**Arguments**

test.table	Data frame of kataegis test scores
cutoff	The minimum hypermutation score used to classify the windows in the sliding binomial test as significant windows. The score is calculated per window as follows: $-\log_{10}(\text{binomial test p-value})$ . Recommended value: 5
somatic	Data frame of somatic variants
output.name	Name of the generated output directory.

**Author(s)**

Fan Fan  
Fouad Yousif

**Examples**

```
load(
  paste0(
    path.package("SeqKat"),
    "/extdata/test/somatic.rda"
  )
);

load(
  paste0(
    path.package("SeqKat"),
    "/extdata/test/test.table.rda"
  )
);

final.score(
  test.table,
  5,
  somatic,
  tempdir()
);
```

---

`get.context`*Get Context*

---

**Description**

Gets the 5' and 3' neighboring bases to the mutated base

**Usage**

```
get.context(file, start)
```

**Arguments**

file	Reference files directory
start	The position of the mutation gene

**Value**

The trinucleotide context.

**Author(s)**

Fouad Yousif  
Fan Fan

## Examples

```
example.ref.dir <- paste0(
  path.package("SeqKat"),
  "/extdata/test/ref/"
);
get.context(file.path(example.ref.dir, 'chr4.fa'), c(1582933, 1611781))
```

---

get.exprobntcx	<i>get.exprobntcx</i>
----------------	-----------------------

---

## Description

Gets the expected probability for each trinucleotide and total number of tcx

## Usage

```
get.exprobntcx(somatic, ref.dir, trinucleotide.count.file)
```

## Arguments

somatic	Data frame of somatic variants
ref.dir	Path to a directory containing the reference genome.
trinucleotide.count.file	A tab separated file containing a count of all trinucleotides present in the reference genome. This can be generated with the get.trinucleotide.counts() function in this package.

## Author(s)

Fan Fan  
Fouad Yousif

## Examples

```
load(
  paste0(
    path.package("SeqKat"),
    "/extdata/test/somatic.rda"
  )
);

trinucleotide.count.file <- paste0(
  path.package("SeqKat"),
  "/extdata/tn_count.txt"
);

example.ref.dir <- paste0(
  path.package("SeqKat"),
```

```
"/extdata/test/ref/"
);

get.exprobntcx(somatic, example.ref.dir, trinucleotide.count.file)
```

---

get.nucleotide.chunk.counts

*Get Nucleotide Chunk Counts*

---

## Description

Obtain counts for all possible trinucleotides within a specified genomic region

## Usage

```
get.nucleotide.chunk.counts(key, chr, upstream = 1, downstream = 1,
  start = 1, end = -1)
```

## Arguments

key	List of specify trinucleotides to count
chr	Chromosome
upstream	Length upstream to read
downstream	Length downstream to read
start	Starting position
end	Ending position

## Author(s)

Fouad Yousif

## Examples

```
example.ref.dir <- paste0(
  path.package("SeqKat"),
  "/extdata/test/ref/"
);

bases.raw <- c('A','C','G','T','N');
tri.types.raw <- c(
  outer(
    c(outer(bases.raw, bases.raw, function(x, y) paste0(x,y))),
    bases.raw, function(x, y) paste0(x,y))
  );
tri.types.raw <- sort(tri.types.raw);
get.nucleotide.chunk.counts(
  tri.types.raw,
  file.path(example.ref.dir, 'chr4.fa'),
```

```
upstream = 1,  
downstream = 1,  
start = 1,  
end = -1  
);
```

---

*get.pair*

*Get Pair*

---

**Description**

Generates the reverse compliment of a nucleotide sequence

**Usage**

```
get.pair(x)
```

**Arguments**

x                    asdf

**Details**

Reverses and compliments the bases of the input string. Bases must be (A, C, G, T, or N).

**Author(s)**

Fouad Yousif

**Examples**

```
get.pair("GATTACA")
```

---

*get.tn*

*Get Trinucleotides*

---

**Description**

Count the frequencies of 32 trinucleotide in a region respectively

**Usage**

```
get.tn(chr, start.bp, end.bp, ref.dir)
```

**Arguments**

chr	Chromosome
start.bp	Starting position
end.bp	Ending position
ref.dir	Path to a directory containing the reference genome.

**Author(s)**

Fan Fan

**Examples**

```
example.ref.dir <- paste0(
  path.package("SeqKat"),
  "/extdata/test/ref/"
);
get.tn(chr=4, start.bp=1, end.bp=-1, example.ref.dir)
```

---

get.toptn

*Get Top Trinucleotides*

---

**Description**

Generate a tri-nucleotide summary for each sliding window

**Usage**

```
get.toptn(somatic.subset, chr, start.bp, end.bp, ref.dir)
```

**Arguments**

somatic.subset	Data frame of somatic variants subset for a specific chromosome
chr	Chromosome
start.bp	Starting position
end.bp	Ending position
ref.dir	Path to a directory containing the reference genome.

**Author(s)**

Fan Fan

Fouad Yousif

### Examples

```
## Not run:  
get.toptn(somatic.subset, chr, start.bp, end.bp, ref.dir)  
  
## End(Not run)
```

---

`get.trinucleotide.counts`

*Get Trinucleotide Counts*

---

### Description

Aggregates the total counts of each possible trinucleotide.

### Usage

```
get.trinucleotide.counts(ref.dir, ref.name, output.dir)
```

### Arguments

<code>ref.dir</code>	Path to a directory containing the reference genome.
<code>ref.name</code>	Name of the reference genome being used (i.e. hg19, GRCh38, etc)
<code>output.dir</code>	Path to a directory where output will be created.

### Author(s)

Fan Fan  
Fouad Yousif

### Examples

```
## Not run:  
get.trinucleotide.counts(ref.dir, "hg19", tempdir());  
  
## End(Not run)
```

seqkat

*SeqKat***Description**

Kataegis detection from SNV BED files

**Usage**

```
seqkat(sigcutoff = 5, mutdistance = 3.2, segnum = 4, ref.dir = NULL,
      bed.file = "./", output.dir = "./", chromosome = "all",
      chromosome.length.file = NULL, trinucleotide.count.file = NULL)
```

**Arguments**

sigcutoff	The minimum hypermutation score used to classify the windows in the sliding binomial test as significant windows. The score is calculated per window as follows: $-\log_{10}(\text{binomial test p-value})$ . Recommended value: 5
mutdistance	The maximum intermutational distance allowed for SNVs to be grouped in the same kataegic event. Recommended value: 3.2
segnum	Minimum mutation count. The minimum number of mutations required within a cluster to be identified as kataegic. Recommended value: 4
ref.dir	Path to a directory containing the reference genome. Each chromosome should have its own .fa file and chromosomes X and Y are named as chr23 and chr24. The fasta files should contain no header
bed.file	Path to the SNV BED file. The BED file should contain the following information: Chromosome, Position, Reference allele, Alternate allele
output.dir	Path to a directory where output will be created.
chromosome	The chromosome to be analysed. This can be (1, 2, ..., 23, 24) or "all" to run sequentially on all chromosomes.
chromosome.length.file	A tab separated file containing the lengths of all chromosomes in the reference genome.
trinucleotide.count.file	A tab separated file containing a count of all trinucleotides present in the reference genome. This can be generated with the <code>get.trinucleotide.counts()</code> function in this package.

**Details**

The default paramters in SeqKat have been optimized using Alexanrov's "Signatures of mutational processes in human cancer" dataset. SeqKat accepts a BED file and outputs the results in TXT format. A file per chromosome is generated if a kataegic event is detected, otherwise no file is generated. SeqKat reports two scores per kataegic event, a hypermutation score and an APOBEC mediated kataegic score.

**Author(s)**

Fouad Yousif  
Fan Fan  
Christopher Lalansingh

**Examples**

```
example.bed.file <- paste0(
  path.package("SeqKat"),
  "/extdata/test/PD4120a-chr4-1-2000000_test_snvs.bed"
);
example.ref.dir <- paste0(
  path.package("SeqKat"),
  "/extdata/test/ref/"
);
example.chromosome.length.file <- paste0(
  path.package("SeqKat"),
  "/extdata/test/length_hg19_chr_test.txt"
);
seqkat(
  5,
  3.2,
  2,
  bed.file = example.bed.file,
  output.dir = tempdir(),
  chromosome = "4",
  ref.dir = example.ref.dir,
  chromosome.length.file = example.chromosome.length.file
);
```

---

test.kataegis

*Test Kataegis*

---

**Description**

Performs exact binomial test to test the deviation of the 32 tri-nucleotides counts from expected

**Usage**

```
test.kataegis(chromosome.num, somatic, units, exprobntcx, output.name, ref.dir,
  chromosome.length.file)
```

**Arguments**

chromosome.num Chromosome  
somatic Data frame of somatic variants  
units Base window size

exprobntcx      Expected probability for each trinucleotide and total number of tcx  
output.name     Name of the generated output directory.  
ref.dir         Path to a directory containing the reference genome.  
chromosome.length.file  
                 A tab separated file containing the lengths of all chromosomes in the reference  
                 genome.

**Author(s)**

Fouad Yousif

**Examples**

```
load(  
  paste0(  
    path.package("SeqKat"),  
    "/extdata/test/somatic.rda"  
  )  
);  
  
load(  
  paste0(  
    path.package("SeqKat"),  
    "/extdata/test/exprobntcx.rda"  
  )  
);  
  
example.chromosome.length.file <- paste0(  
  path.package("SeqKat"),  
  "/extdata/test/length_hg19_chr_test.txt"  
);  
  
example.ref.dir <- paste0(  
  path.package("SeqKat"),  
  "/extdata/test/ref/"  
);  
  
test.kataegis(  
  4,  
  somatic,  
  2,  
  exprobntcx,  
  tempdir(),  
  example.ref.dir,  
  example.chromosome.length.file  
);
```

# Index

`combine.table`, 2

`final.score`, 3

`get.context`, 4

`get.exprobntcx`, 5

`get.nucleotide.chunk.counts`, 6

`get.pair`, 7

`get.tn`, 7

`get.toptn`, 8

`get.trinucleotide.counts`, 9

`seqkat`, 10

`test.kataegis`, 11