

# Package ‘roverlaps’

May 15, 2018

**Type** Package

**Title** Fast, Memory-Efficient Genomic Range Overlaps

**Version** 0.1.1

**Author** Jeremiah Wala

**Date** 2018-05-13

**Maintainer** Jeremiah Wala <jwala@broadinstitute.org>

**Description** Range overlaps between different sets of genomic intervals can be memory intensive and slow for huge (1M+) interval queries. This package implements fast, memory-efficient interval tree overlaps in C++ and provides flexibility for users to choose only what they need, thereby improving memory performance.

**License** MIT + file LICENSE

**SystemRequirements** A C++11 compiler. Version 4.6.\* of g++ (as currently in Rtools) is insufficient; versions 4.8.\*, 4.9.\* or later will be fine.

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Imports** Rcpp (>= 0.12.16), GenomicRanges, data.table, S4Vectors, methods, utils

**LinkingTo** Rcpp

**Suggests** testthat

## R topics documented:

roverlaps . . . . .	<a href="#">2</a>
Index	<a href="#">4</a>

roverlaps

*roverlaps***Description**

Range overlaps between different sets of genomic intervals can be memory intensive and slow for huge (1M+) interval queries. This package implements fast, memory-efficient interval tree overlaps in C++ and provides flexibility for users to choose only what they need, thereby improving memory performance.

Performs interval overlaps between two genomic ranges, returning the intersecting set of ranges and the indices of the query and subject which created the interval.

**Usage**

```
roverlaps(query, subject, verbose = FALSE, index_only = FALSE)
```

**Arguments**

query	Query ranges as a data.table with mandatory fields seqnames and start
subject	Subject ranges as a data.table with mandatory fields seqnames and start
verbose	Increase the verbosity [FALSE]
index_only	Return only the indices ('query.id' and 'subject.id') [FALSE]

**Value**

data.table ('seqnames', 'start', 'end', 'query.id', 'subject.id') of overlaps

**Note**

Positions in ranges are inclusive. Example: chr1:2-5 (query) and chr1:5-7 (subject) will create an overlap with value chr1:5-5.

**Author(s)**

Jeremiah Wala <jwala@broadinstitute.org>

**Examples**

```
library(data.table)
set.seed(42)
C=10000
sn1 <- factor(c(1:22, "X")[sample(seq(23),C, replace=TRUE)]))
s1 <- sample(seq(100000), C, replace=TRUE)
o1 <- data.table(seqnames=sn1, start=s1)
o1[, end := start + 100]
sn2 <- factor(c(1:22, "X")[sample(seq(23),C, replace=TRUE)]))
s2 <- sample(seq(100000), C, replace=TRUE)
o2 <- data.table(seqnames=sn2, start=s2)
o2[, end := start + 100]
o <- roverlaps(o1,o2)
```

```
## output
# seqnames start   end query.id subject.id
#       7 83405 83451       3      3315
#       7 83405 83463       3      3148
#       7 83485 83505       3      1022
```

```
oi <- roverlaps(o1, o2, index_only=TRUE)
```

```
## output
#   query.id subject.id
#         3      3315
#         3      3148
#         3      1022
```

# Index

roverlaps, [2](#)

roverlaps-package (roverlaps), [2](#)