gUtils Documentation

Release 1.0.0

Marcin Imielinski and Jeremiah Wala

Dec 10, 2018

Contents

Contents:

CHAPTER 1

Introduction

GRanges is like a data frame whose “rows” are attached to genomic coordinates.

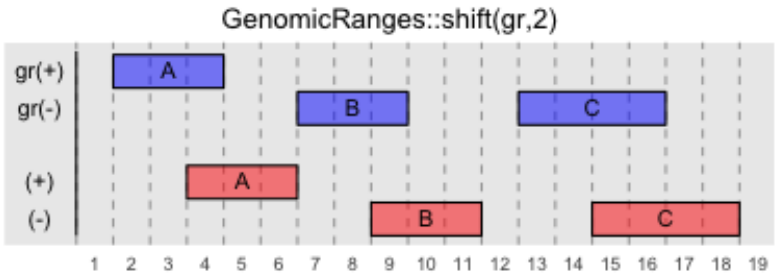
These coordinates live inside some “genome” that has a fixed set of sequences each with a length.

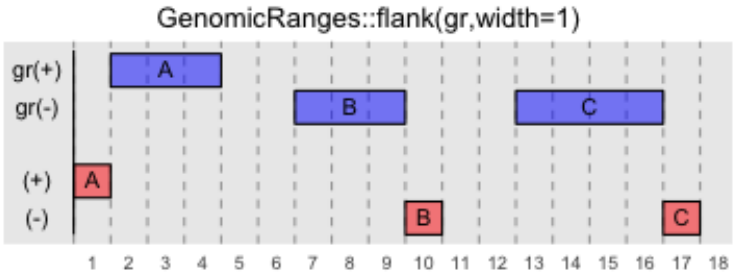
CHAPTER 2

Range Operations

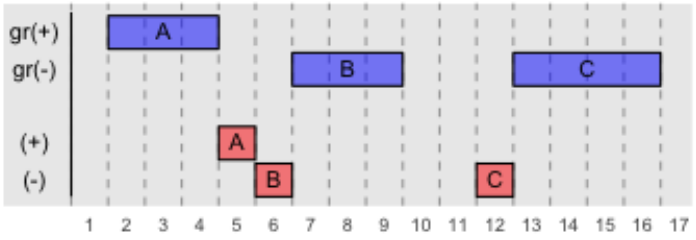
This section will describe additional GRanges operations provided by gUtils.

```
## make some example data sets
ref19 <- readRDS(system.file("extdata","refGene.hg19.gr.rds", package="gUtils"))
gr  <- GRanges(1, IRanges(c(2,5,10), c(4,9,16)), seqinfo=Seqinfo("1", 20))
gr2 <- c(gr, GRanges(1, IRanges(c(1,9), c(6,14)), seqinfo=Seqinfo("1", 20)))
dt <- data.table(seqnames=1, start=c(2,5,10), end=c(3,8,15))
```

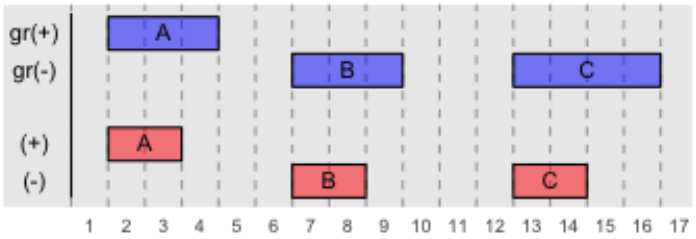




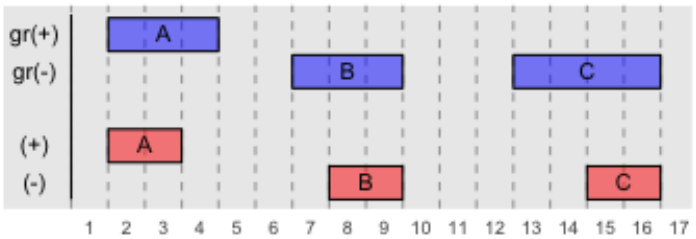
GenomicRanges::flank(gr,start=FALSE, width=1)



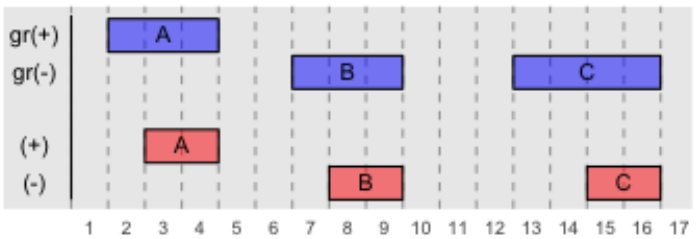
gr.start(gr, width=2)



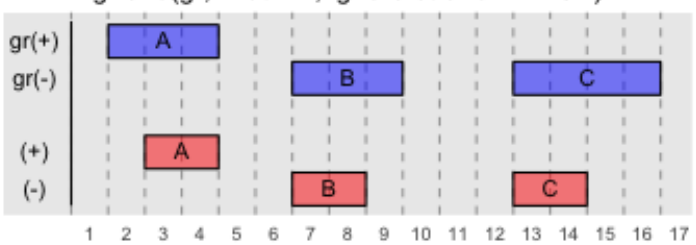
gr.start(gr, width=2, ignore.strand=FALSE)



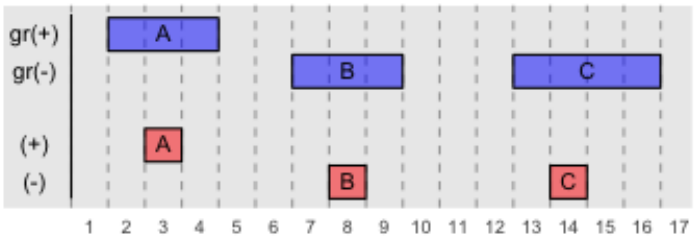
gr.end(gr, width=2)



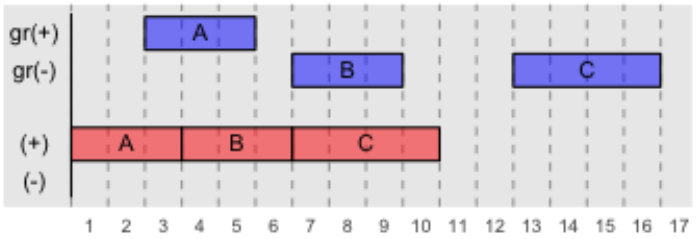
gr.end(gr, width=2, ignore.strand=FALSE)



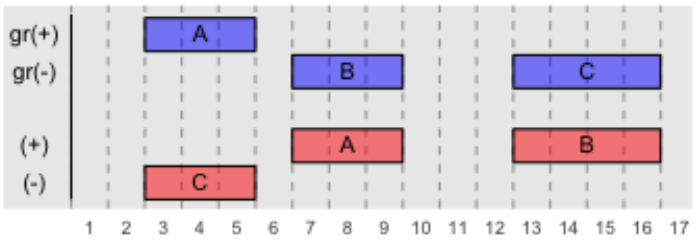
gr.mid(gr)



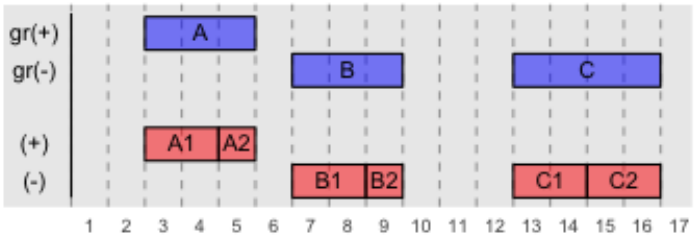
gr.flatten(gr)



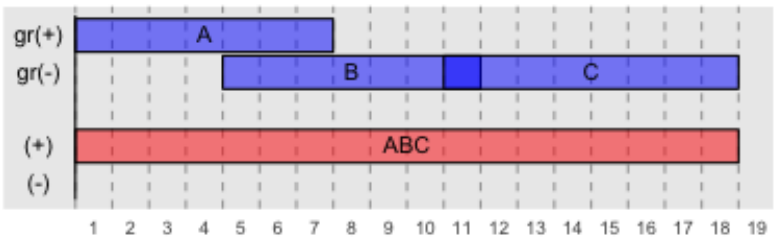
gr.flipstrand(gr)

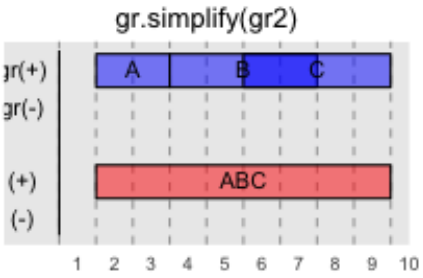


gr.tile(gr,w=2)



streduce(gr+2)




gr.simplify(gr2)
gr(+)
gr(-)
(+)
(-)
A
B
C
ABC
1
2
3
4
5
6
7
8
9
10

CHAPTER 3

Data manipulation

R provides a number of data structures for storing genomic data, each with its advantages and drawbacks.

The most useful structures for this purpose are:

GRanges Store ranges along with metadata, sequences and the coordaintes of the reference genome.

GRangesList Store groups of ranges, with additional metadata belonging to the group.

data.table Fast and efficient general-purpose container similar to data.frame, but with significant performance improvements.

In gUtils functions, we often manipulate the data to move between these data structures where one is more useful than another. A key example is in `gr.findoverlaps`, which converts the input `GRanges` into `data.table` objects to take advantage of the blazing fast `foverlaps` util. For the most part, these conversions should be invisible to the user.

However, often there are data structures conversions that may be useful to the end user. This includes unlisting GRangesList objects into GRanges, making data.table objects from GRanges, and binding together multiple GRanges or GRangesList objects, among others. This section will describe and demonstrate the functionality `gUtils` provides for manipulating these data structures.

```
ref19 <- readRDS(system.file("extdata","refGene.hg19.gr.rds", package="gUtils"))
gr <- GRanges(1, IRanges(c(2,5,10), c(4,9,16)), seqinfo=Seqinfo("1", 20))
dt <- data.table(seqnames=1, start=c(2,5,10), end=c(3,8,15))
```

`grbind`

```
## add metadata to one field
mcols(gr)$score = 3
## try to concatenate
c(gr,gr2)  ## ERROR
## with grbind
grbind(gr, gr2) ## SUCCESS. Adds NA for missing fields
## GenomicRanges::c does this already for GRangesList
```

CHAPTER 4

Additional utilities

gr.string

grl.string

gr.chr

gr.fix

gr.findoverlaps

gr.dist

grl.in

CHAPTER 5

Tutorial

gr1 and gr2 are GRanges objects

`gr1 %^% gr2`

gives a length(gr1) logical vector with values of TRUE or FALSE. TRUE if gr1 intersects some interval in gr2, FALSE otherwise.

`gr1 %*% gr2`

gives a length n (where n <= length(gr1)*length(gr2)) of all pairwise overlaps of gr1 and gr2 with merged meta.data

`gr1 %$% gr2`

returns length(gr1) GRanges aggregating all metadata values of gr2 within gr1, taking mean if the meta data item is numeric and concatenating the value if it is a string

`gr1 %Q% (expression)`

subsets gr1 using indices or logical values resulting from expression

`gr1 %Q% (gene== "EGFR")`

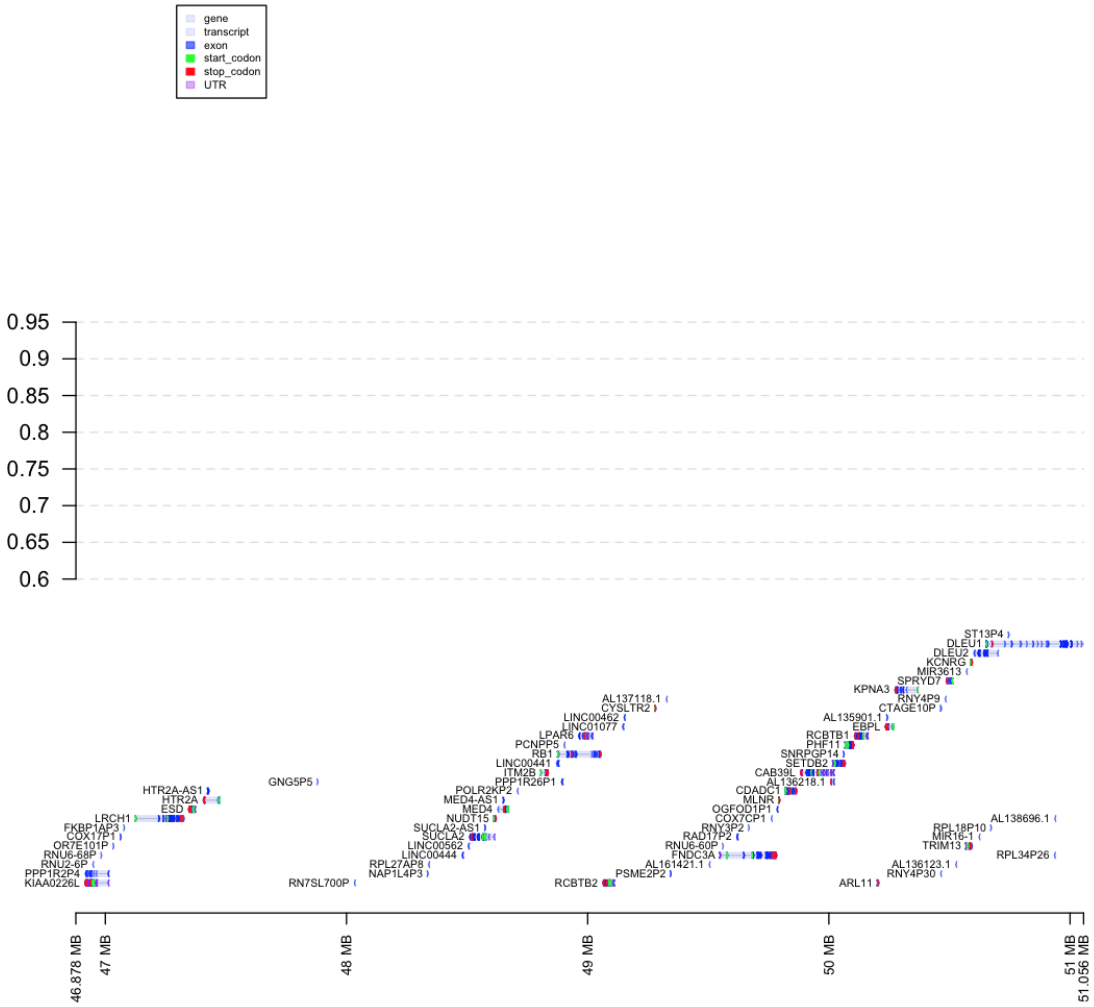
will return the subset of gr1 entries for whose metadata column $gene has "EGFR"

```
library(gTrack)
library(gUtils)

## tutorial on using the above operations

## set current working directory to the folder where the OSMIC TSV was saved in.
setwd("~/Downloads")

## load the load. Need to download cosmic data set - http://cancer.sanger.ac.uk/census
COSMICgenes <- read.delim("cosmicdata.tsv")

## save the gene symbols into a factor
geneSymbols <- COSMICgenes[,1]
```

(continues on next page)

(continued from previous page)

```
## loading gene definitions which we can use to plot windows (GRanges)
genes = readRDS('files/genes.rds')

## this loads coverage data from cancer cell line (GRanges object, have to make it
↪into a gTrack)
cov = readRDS('files/coverage.rds')

## loading the GENCODE gene model gTrack
gt.ge = track.gencode()

# subset genes to just the genes in the COSMIC dataset
genez <- genes %Q%(gene_name==geneSymbols[1])

# subset the rest of the genes
for (i in 2:length(geneSymbols)) {genez <- c( genes %Q% (gene_name==geneSymbols[i]) ,
↪genez)}

# add the average coverage metadata column for each gene
genez <- genez %$% cov

gt.cov = gTrack(genezz , y.field = 'mean' , circles = TRUE , col = 'blue' , name =
↪'Cov')

# set window for plot -RB1 gene and genes around it in a 2e6 range
window = genez[genez$gene_name == "RB1"] + 2e6

plot(c(gt.ge , gt.cov) , window)
```



chr 13:
46.878-
51.056 MB
(4.18 MB)

CHAPTER 6

Indices and tables

- genindex
- modindex
- search