

Package: igvShiny (via r-universe)

September 30, 2026

Title igvShiny: a wrapper of Integrative Genomics Viewer (IGV - an interactive tool for visualization and exploration integrated genomic data)

Version 1.9.51

Date 2026-09-30

Description This package is a wrapper of Integrative Genomics Viewer (IGV). It comprises an htmlwidget version of IGV. It can be used as a module in Shiny apps.

URL <https://github.com/gladkia/igvShiny>,
<https://gladkia.github.io/igvShiny/>

BugReports <https://github.com/gladkia/igvShiny/issues>

Depends GenomicRanges, methods, shiny (>= 1.6.0)

Imports BiocGenerics, checkmate, futile.logger, GenomeInfoDb, GenomeInfoDbData, htmlwidgets, httr, jsonlite, randomcoloR, stats, utils

Suggests BiocStyle, bslib, covr, GenomicAlignments, knitr, Rhtslib, Rsamtools, rtracklayer, shinytest2, testthat, VariantAnnotation

biocViews Software, ShinyApps, Sequencing, Coverage

License MIT + file LICENSE

Encoding UTF-8

Collate 'constants.R' 'genomeSpec.R' 'referenceCheck.R' 'GWASTrack.R' 'rangeServer.R' 'igvShiny.R' 'packages.R' 'utils.R' 'zzz.R'

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libnode-dev zlib1g-dev

Repository <https://bioc.r-universe.dev>

Date/Publication 2026-09-29 23:50:39 UTC

RemoteUrl <https://github.com/bioc/igvShiny>

RemoteRef HEAD

RemoteSha bd7a6a83fd5df571c4bd06b2d94c2b6a97d2f91e

Contents

.packJunctionAttributes	3
.sanitizeAndMergeOptions	3
.sanitizeChromosomeColorMap	4
.sanitizeTrack	4
.sanitizeTracks	5
checkReferenceCompatibility	5
display,GWASTrack-method	6
get_basic_genomes	7
get_cas_genomes	8
get_css_genomes	8
get_tracks_dir	9
getUrl,GWASTrack-method	9
GWASTrack-class	10
igvShiny	12
igvShinyOutput	13
loadBamTrackFromLocalData	14
loadBamTrackFromLocalFile	15
loadBamTrackFromURL	16
loadBedGraphTrack	17
loadBedGraphTrackFromURL	19
loadBedTrack	20
loadCramTrackFromLocalData	21
loadCramTrackFromURL	23
loadGFF3TrackFromLocalData	24
loadGFF3TrackFromURL	25
loadGwasTrack	26
loadSegTrack	28
loadSpliceJunctionTrackFromLocalData	29
loadSpliceJunctionTrackFromURL	30
loadVcfTrack	31
parseAndValidateGenomeSpec	32
removeTracksByName	34
removeUserAddedTracks	34
renderIgvShiny	35
serveLocalFile	36
show,GWASTrack-method	37
showGenomicRegion	37

Index

39

`.packJunctionAttributes`

Pack junction attributes into a bed name column

Description

Pack junction attributes into a bed name column

Usage

```
.packJunctionAttributes(tbl)
```

Arguments

tbl A data.frame of junctions.

Value

A character vector, one packed name field per row.

`.sanitizeAndMergeOptions`

Sanitize and merge track configuration options

Description

Sanitize and merge track configuration options

Usage

```
.sanitizeAndMergeOptions(baseOptions, userOptions)
```

Arguments

baseOptions A list of default options set by the R function.
userOptions A list of options provided by the user via trackConfig.

Value

A merged and sanitized list of options ready to be sent to JavaScript.

`.sanitizeChromosomeColorMap`

Check a chromosome color map before it reaches igv.js

Description

Check a chromosome color map before it reaches igv.js

Usage

`.sanitizeChromosomeColorMap(map)`

Arguments

map	a named list or character vector of colors, possibly empty
-----	--

Value

the map as a named list, empty if nothing was supplied

`.sanitizeTrack`

Sanitize one startup track specification

Description

Sanitize one startup track specification

Usage

`.sanitizeTrack(track, depth)`

Arguments

track	A named list, one igv.js track configuration.
depth	An integer, how deep this track sits in a merged track.

Value

The sanitized track, or NULL if it cannot be used.

.sanitizeTracks	<i>Sanitize a list of startup track specifications</i>
-----------------	--

Description

Sanitize a list of startup track specifications

Usage

```
.sanitizeTracks(tracks, depth = 1L)
```

Arguments

tracks	A list of named lists, each an igv.js track configuration.
depth	An integer, how deep this list sits in a merged track; 1 for the startup list itself.

Value

A sanitized list of track configurations; invalid entries or keys are dropped with a warning.

checkReferenceCompatibility	<i>Check reference compatibility between track data and genome specification</i>
-----------------------------	--

Description

Compares contig naming conventions (e.g. UCSC chr1 vs Ensembl 1), contig lengths (e.g. hg19 vs hg38), and coordinate boundaries between a track (BAM, CRAM, VCF, BED, or Bioconductor object) and the active reference genome.

Usage

```
checkReferenceCompatibility(  
  target,  
  genomeName = NULL,  
  genomeSpec = NULL,  
  session = NULL,  
  id = NULL  
)
```

Arguments

target	A file path (.bam, .cram, .vcf, .vcf.gz, .bed) or data object (data.frame, GenomicAlignments, VCF, GRanges)
genomeName	Character string, e.g. "hg38", "hg19", "mm10"
genomeSpec	Named list of genome options, as produced by parseAndValidateGenomeSpec
session	Optional Shiny session object
id	Optional widget element ID

Value

A list with:

- checked: logical indicating whether both inputs could be inspected
- compatible: logical indicating if no critical mismatches were found, or NA when the inputs could not be checked
- mismatches: character vector of descriptive warning messages
- details: list of specific detected anomalies (naming, length, out-of-bounds)
- targetAssembly: detected assembly of track, if recognized
- referenceAssembly: reference genome name

Examples

```
bamFile <- system.file(package = "igvShiny", "extdata", "tumor.bam")
# tumor.bam is GRCh38: compatible with hg38, incompatible with hg19
res1 <- checkReferenceCompatibility(bamFile, genomeName = "hg38")
res1$compatible # TRUE

res2 <- checkReferenceCompatibility(bamFile, genomeName = "hg19")
res2$compatible # FALSE
```

display,GWASTrack-method

display the already constructed and configured track

Description

display the already constructed and configured track

Usage

```
## S4 method for signature 'GWASTrack'
display(obj, session, id, deleteTracksOfSameName = TRUE)
```

Arguments

obj	An object of class GWASTrack
session	a Shiny session object
id	character the identifier of the target igv object in the browser
deleteTracksOfSameName	logical to avoid duplications in track names

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

get_basic_genomes	<i>get_basic_genomes</i>
-------------------	--------------------------

Description

a helper function for basic genomes, obtains the genome codes (e.g. 'hg38')

Usage

```
get_basic_genomes()
```

Value

an list of short genome codes, e.g., "hg38", "dm6"

Examples

```
bs <- get_basic_genomes()
```

get_cas_genomes	<i>get_cas_genomes</i>
-----------------	------------------------

Description

a helper function for common always available stock genomes, obtains the genome codes (e.g. 'hg38')

Usage

```
get_cas_genomes()
```

Value

an list of short genome codes, e.g., "hg38", "dm6"

Examples

```
cas <- get_cas_genomes()
```

get_css_genomes	<i>get_css_genomes</i>
-----------------	------------------------

Description

a helper function for mostly internal use, obtains the genome codes (e.g. 'hg38') supported by igv.js

Usage

```
get_css_genomes(test = FALSE)
```

Arguments

test	logical(1) defaults to FALSE
------	------------------------------

Value

an list of short genome codes, e.g., "hg38", "dm6", "tair10"

Examples

```
css <- get_css_genomes(test = TRUE)
```

get_tracks_dir	<i>get_tracks_dir</i> Get the directory where tracks are stored. The directory can be defined with environmental variable. If not defined the default is a directory called "tracks" in the temp directory. We need a local directory to write files - for instance, a vcf file representing a genomic region of interest. We then tell shiny about that directory, so that shiny's built-in http server can serve up files we write there, ultimately consumed by igv.js
----------------	---

Description

get_tracks_dir Get the directory where tracks are stored. The directory can be defined with environmental variable. If not defined the default is a directory called "tracks" in the temp directory. We need a local directory to write files - for instance, a vcf file representing a genomic region of interest. We then tell shiny about that directory, so that shiny's built-in http server can serve up files we write there, ultimately consumed by igv.js

Usage

```
get_tracks_dir(env_var = "TRACKS_DIR")
```

Arguments

env_var	The name of the environmental variable to use.
---------	--

Value

string with the path to the tracks directory.

Examples

```
gtd <- get_tracks_dir(env_var = "TRACKS_DIR")
```

getUrl, GWASTrack-method	<i>the url of the gwas table</i>
--------------------------	----------------------------------

Description

the url of the gwas table

Usage

```
## S4 method for signature 'GWASTrack'
getUrl(obj)
```

Arguments

obj An object of class GWASTrack

Value

character

Examples

```
file <-
  # a local gwas file
  system.file(package = "igvShiny", "extdata", "gwas-5k.tsv.gz")
tbl.gwas <- read.table(file,
                      sep = "\t",
                      header = TRUE,
                      quote = "")

track <-
  GWASTrack(
    "gwas 5k",
    tbl.gwas,
    chrom.col = 12,
    pos.col = 13,
    pval.col = 28
  )
getUrl(track)
```

GWASTrack-class

Constructor for GWASTrack

Description

GWASTrack creates an IGV manhattan track from GWAS data

Usage

```
GWASTrack(
  trackName,
  data,
  chrom.col,
  pos.col,
  pval.col,
  chromosomeColorMap = list(),
  trackHeight = 50,
  autoscale = TRUE,
  minY = 0,
  maxY = 30
)
```

Arguments

trackName	A character string, used as track label by igv, we recommend unique names per track.
data	a data.frame or a url pointing to one, whose structure is described by chrom.col, pos.col, pval.col
chrom.col	numeric, the column number of the chromosome column
pos.col	numeric, the column number of the position column
pval.col	numeric, the column number of the GWAS pvalue column
chromosomeColorMap	a named list or character vector mapping chromosome names, as they are spelled in the data, to colors, e.g. <code>list("1" = "red", "2" = "blue")</code> . A "*" entry colors every chromosome not named explicitly. Empty by default, which leaves the igv.js chromosome palette in place
trackHeight	numeric in pixels
autoscale	logical
minY	numeric for explicit (non-auto) scaling
maxY	numeric for explicit (non-auto) scaling

Value

A GWAS*Track* object

Examples

```
file <-
  # a local gwas file
  system.file(package = "igvShiny", "extdata", "gwas-5k.tsv.gz")
tbl.gwas <- read.table(file,
                      sep = "\t",
                      header = TRUE,
                      quote = "")

dim(tbl.gwas)
track <-
  GWASTrack(
    "gwas 5k",
    tbl.gwas,
    chrom.col = 12,
    pos.col = 13,
    pval.col = 28
  )
getUrl(track)

# a remote gwas file: the constructor checks that the url resolves, so this
# block reaches the network and stays out of R CMD check

url <- "https://gladki.pl/igvShiny/gwas_sample.tsv.gz"
track <- GWASTrack(
  "remote url gwas",
```

```

    url,
    chrom.col = 3,
    pos.col = 4,
    pval.col = 10,
    autoscale = FALSE,
    minY = 0,
    maxY = 300,
    trackHeight = 100
  )
  getUrl(track)

# colors picked per chromosome, with "*" covering the rest
track <- GWASTrack(
  "gwas 5k, custom colors",
  tbl.gwas,
  chrom.col = 12,
  pos.col = 13,
  pval.col = 28,
  chromosomeColorMap = list("1" = "red", "2" = "blue", "*" = "gray")
)

```

igvShiny

Create an igvShiny instance

Description

This function is called in the server function of your shiny app

Usage

```

igvShiny(
  genomeOptions,
  width = NULL,
  height = NULL,
  elementId = NULL,
  displayMode = "squished",
  tracks = list()
)

```

Arguments

genomeOptions	a list with these fields: genomeName, initialLocus, annotation, dataMode, fasta, fastaIndex, stockGenome, validated
width	a character string, standard css notations, either e.g., "1000px" or "95%"
height	a character string, needs to be an explicit pixel measure, e.g., "800px"

elementId	a character string, the html element id within which igv is created
displayMode	a character string, default "SQUISHED".
tracks	a list of track specifications to be created and displayed at startup. Each element is itself a named list of igv.js track options (e.g. name, type, format, url), for example: <code>list(list(name="genes", type="annotation", format="gff3",url="https://.../gen</code> Unrecognized keys are dropped with a warning; see <code>.validIgvTrackOptions</code> for the full allowlist.

Value

the created widget

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

igvShinyOutput	<i>create the UI for the widget</i>
----------------	-------------------------------------

Description

This function is called in the ui function of your shiny app

Usage

```
igvShinyOutput(outputId, width = "100%", height = NULL)
```

Arguments

outputId	a character string, specifies the html element id
width	a character string, standard css notations, either e.g., "1000px" or "95%", "100%" by default
height	a character string, needs to be an explicit pixel measure, e.g., "800px", "400px" by default

Value

the created widget's html

Examples

```
io <- igvShinyOutput("igvOut")
```

loadBamTrackFromLocalData

load GenomicAlignments data as an igv.js alignment track

Description

load GenomicAlignments data as an igv.js alignment track

Usage

```
loadBamTrackFromLocalData(
  session,
  id,
  trackName,
  data,
  deleteTracksOfSameName = TRUE,
  displayMode = "EXPANDED",
  trackConfig = list(),
  validateReference = TRUE,
  trackHeight = NULL
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
data	GenomicAlignments object
deleteTracksOfSameName	logical, default TRUE
displayMode	character string, possible values are "EXPANDED"(default), "SQUISHED" or "COLLAPSED"
trackConfig	a named list of additional igv.js track configuration options, sort among them; see loadBamTrackFromURL .
validateReference	logical flag, default TRUE: checks track contigs against the active reference genome and emits warnings/toasts on mismatch.
trackHeight	numeric, track height in pixels (default: NULL, igv.js defaults to 300)

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadBamTrackFromLocalFile

Load a BAM Track from a Local File via HTTP Range Requests

Description

Streams a local BAM file and its index directly from disk using HTTP 206 Partial Content requests handled by the Shiny session. This allows viewing multi-gigabyte BAM files without loading them into R memory.

Usage

```
loadBamTrackFromLocalFile(
  session,
  id,
  trackName,
  bamFile,
  indexFile = paste0(bamFile, ".bai"),
  deleteTracksOfSameName = TRUE,
  displayMode = "EXPANDED",
  trackConfig = list(),
  validateReference = TRUE,
  trackHeight = NULL
)
```

Arguments

session	an environment or list, provided and managed by shiny (ShinySession)
id	character string, the html element id of this widget instance
trackName	character string, display name for the track
bamFile	character string, path to an existing, readable .bam file
indexFile	character string, path to the corresponding .bai index file (default: paste0(bamFile, ".bai"))
deleteTracksOfSameName	logical, whether to delete any existing track with the same name (default: TRUE)
displayMode	character string, display mode for alignments ("EXPANDED", "COLLAPSED", or "SQUISHED"), default "EXPANDED"

trackConfig list, additional track options passed to igv.js

validateReference logical flag, default TRUE: checks track contigs against the active reference genome and emits warnings/toasts on mismatch.

trackHeight numeric, track height in pixels (default: NULL, igv.js defaults to 300)

Value

None, sends a message to the browser

Examples

```
## Not run:
# Inside a Shiny server function:
bamFile <- system.file(package = "igvShiny", "extdata",
                        "A_2_A24_02_01_01.nanopore.minimap.sorted.bam")
baiFile <- paste0(bamFile, ".bai")
loadBamTrackFromLocalFile(session, "igvShiny_0", "Nanopore Reads",
                          bamFile, baiFile, displayMode = "SQUISHED",
                          trackHeight = 100)

## End(Not run)
```

loadBamTrackFromURL *load a bam track which, with index, is served up by http*

Description

load a remote bam track

Usage

```
loadBamTrackFromURL(
  session,
  id,
  trackName,
  bamURL,
  indexURL,
  deleteTracksOfSameName = TRUE,
  displayMode = "EXPANDED",
  showAllBases = FALSE,
  trackConfig = list(),
  trackHeight = NULL
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
bamURL	character string http url for the bam file, typically very large
indexURL	character string http url for the bam file index, typically small
deleteTracksOfSameName	logical, default TRUE
displayMode	character string, possible values are "EXPANDED"(default), "SQUISHED" or "COLLAPSED"
showAllBases	logical, show all bases in the alignment, default FALSE
trackConfig	a named list of additional igv.js track configuration options. Alignment tracks read sort: list(sort = list(chr = "chr1", position = 155160540, option = "TAG", tag = "HP")) sorts the reads at that 1-based position by the HP tag. Any igv.js sort option works (BASE, STRAND, INSERT_SIZE, ...); reads sort descending unless direction = "ASC".
trackHeight	numeric, track height in pixels (default: NULL, igv.js defaults to 300)

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadBedGraphTrack	<i>load a scored genome annotation track provided as a data.frame</i>
-------------------	---

Description

load a genome annotation track provided as a data.frame

Usage

```
loadBedGraphTrack(
  session,
  id,
  trackName,
  tbl,
  color = "gray",
  trackHeight = 30,
  autoscale,
  autoscaleGroup = -1,
  min = NA_real_,
  max = NA_real_,
  deleteTracksOfSameName = TRUE,
  quiet = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
tbl	data.frame, with at least "chrom" "start" "end" "score" columns
color	character string, a legal CSS color, or "random", "gray" by default
trackHeight	an integer, 30 (pixels) by default
autoscale	logical
autoscaleGroup	numeric(1) defaults to -1
min	numeric, consulted when autoscale is FALSE
max	numeric, consulted when autoscale is FALSE
deleteTracksOfSameName	logical, default TRUE
quiet	logical, default TRUE, controls verbosity
trackConfig	a named list of additional igv.js track configuration options.
validateReference	logical flag, default TRUE: checks track contigs and coordinates against the active reference genome and emits warnings/toasts on mismatch.

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadBedGraphTrackFromURL

load a bedgraph track from a URL

Description

load a bedgraph track provided as a data.frame

Usage

```
loadBedGraphTrackFromURL(
  session,
  id,
  trackName,
  url,
  color = "gray",
  trackHeight = 30,
  autoscale = TRUE,
  min = 0,
  max = 1,
  autoscaleGroup = -1,
  deleteTracksOfSameName = TRUE,
  quiet = TRUE,
  trackConfig = list()
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
url	character
color	character string, a legal CSS color, or "random", "gray" by default
trackHeight	an integer, 30 (pixels) by default
autoscale	logical
min	numeric, consulted when autoscale is FALSE

max numeric, consulted when autoscale is FALSE
 autoscaleGroup numeric(1) defaults to -1
 deleteTracksOfSameName
 logical(1) defaults to TRUE
 quiet logical, default TRUE, controls verbosity
 trackConfig a named list of additional igv.js track configuration options.

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadBedTrack	<i>load a bed track provided as a data.frame</i>
--------------	--

Description

load a bed track provided as a data.frame

Usage

```
loadBedTrack(
  session,
  id,
  trackName,
  tbl,
  color = "",
  trackHeight = 50,
  deleteTracksOfSameName = TRUE,
  quiet = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
tbl	data.frame, with at least "chrom" "start" "end" columns
color	character string, a legal CSS color, or "random", "gray" by default
trackHeight	an integer, 50 (pixels) by default
deleteTracksOfSameName	logical, default TRUE
quiet	logical, default TRUE, controls verbosity
trackConfig	a named list of additional igv.js track configuration options.
validateReference	logical flag, default TRUE: checks track contigs and coordinates against the active reference genome and emits warnings/toasts on mismatch.

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

```
loadCramTrackFromLocalData
```

load a cram file sitting on the same machine as the shiny app

Description

load a local cram track. Unlike loadBamTrackFromLocalData this loader takes file paths, not an R object: no bioconductor package parses cram, so the file and its index are handed to igv.js untouched, through the directory shiny serves as "tracks".

Usage

```
loadCramTrackFromLocalData(
  session,
  id,
  trackName,
  cramFile,
  indexFile = paste0(cramFile, ".crai"),
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE,
  trackHeight = NULL
)

loadCramTrackFromLocalFile(
  session,
  id,
  trackName,
  cramFile,
  indexFile = paste0(cramFile, ".crai"),
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE,
  trackHeight = NULL
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
cramFile	character string, path to a cram file
indexFile	character string, path to its crai index, by default the cram file with ".crai" appended
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options, sort among them; see loadBamTrackFromURL .
validateReference	logical flag, default TRUE: checks track contigs against the active reference genome and emits warnings/toasts on mismatch.
trackHeight	numeric, track height in pixels (default: NULL, igv.js defaults to 300)

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadCramTrackFromURL *load a cram track which, with index, is served up by http*

Description

load a remote cram track

Usage

```
loadCramTrackFromURL(
  session,
  id,
  trackName,
  cramURL,
  indexURL,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  trackHeight = NULL
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
cramURL	character string http url for the bam file, typically very large
indexURL	character string http url for the bam file index, typically small
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options, sort among them; see loadBamTrackFromURL .
trackHeight	numeric, track height in pixels (default: NULL, igv.js defaults to 300)

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadGFF3TrackFromLocalData

load a GFF3 track defined by local data

Description

load a local GFF3 track file

Usage

```
loadGFF3TrackFromLocalData(
  session,
  id,
  trackName,
  tbl.gff3,
  color = "gray",
  colorTable,
  colorByAttribute,
  displayMode,
  trackHeight = 50,
  visibilityWindow,
  deleteTracksOfSameName = TRUE,
  trackConfig = list()
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
tbl.gff3	data.frame in standard 9-column GFF3 format
color	character #RGB or a recognized color name. ignored if colorTable and colorByAttribute provided
colorTable	list, mapping a gff3 attribute, typically biotype, to a color
colorByAttribute	character, name of a gff3 attribute in column 9, typically "biotype"

displayMode	character, "EXPANDED", "SQUISHED" or "COLLAPSED"
trackHeight	numeric defaults to 50
visibilityWindow	numeric, Maximum window size in base pairs for which indexed annotations or variants are displayed
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options.

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "local-data.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadGFF3TrackFromURL *load a GFF3 track which, with index, is served up by http*

Description

load a remote GFF3 track

Usage

```
loadGFF3TrackFromURL(
  session,
  id,
  trackName,
  gff3URL,
  indexURL,
  color = "gray",
  colorTable,
  colorByAttribute,
  displayMode,
  trackHeight = 50,
  visibilityWindow,
  deleteTracksOfSameName = TRUE,
  trackConfig = list()
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
gff3URL	character string http url for the bam file, typically very large
indexURL	character string http url for the bam file index, typically small
color	character #RGB or a recognized color name. ignored if colorTable and colorByAttribute provided
colorTable	list, mapping a gff3 attribute, typically biotype, to a color
colorByAttribute	character, name of a gff3 attribute in column 9, typically "biotype"
displayMode	character, "EXPANDED", "SQUISHED" or "COLLAPSED"
trackHeight	numeric defaults to 50
visibilityWindow	numeric, Maximum window size in base pairs for which indexed annotations or variants are displayed
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options.

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "local-data.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadGwasTrack	<i>load a GWAS (genome-wide association study) track provided as a data.frame</i>
---------------	---

Description

load a GWAS (genome-wide association study) track provided as a data.frame

Usage

```
loadGwasTrack(  
  session,  
  id,  
  trackName,  
  tbl.gwas,  
  ymin = 0,  
  ymax = 35,  
  deleteTracksOfSameName = TRUE,  
  trackConfig = list()  
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
tbl.gwas	data.frame, with at least "chrom" "start" "end" columns
ymin	numeric defaults to 0
ymax	numeric defaults to 35
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options. A table whose headers igv.js does not recognize needs its layout spelled out, 1-based, as <code>columns = list(chromosome = 3, position = 4, value = 10)</code> ; <code>colorTable</code> takes a chromosome-to-color map, see GWASTrack

Value

nothing

Examples

```
library(igvShiny)  
demo_app_file <-  
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")  
if (interactive()) {  
  shiny::runApp(demo_app_file)  
}
```

loadSegTrack	<i>load a seg track provided as a data.frame</i>
--------------	--

Description

load a SEG track provided as a data.frame. igv "displays segmented data as a blue-to-red heatmap where the data range is -1.5 to 1.5... The segmented data file format is the output of the Circular Binary Segmentation algorithm (Olshen et al., 2004)".

Usage

```
loadSegTrack(  
  session,  
  id,  
  trackName,  
  tbl,  
  deleteTracksOfSameName = TRUE,  
  trackConfig = list()  
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
tbl	data.frame, with at least "chrom" "start" "end" "score" columns
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options.

Value

nothing

Examples

```
library(igvShiny)  
demo_app_file <-  
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")  
if (interactive()) {  
  shiny::runApp(demo_app_file)  
}
```

loadSpliceJunctionTrackFromLocalData

load a splice junction track from a data.frame

Description

load splice junctions held in R. The table is written as the six-column bed igv.js draws junctions from, and served from the same directory as the other local-data tracks.

Coordinates are taken as bed coordinates and written out unchanged, as the other *FromLocalData loaders take theirs. STAR reports the first and last intron base of a junction 1-based in SJ.out.tab, so a table read straight from that file needs its start shifted by one first; a bed already converted from it, which is what STAR wrappers and the igv.js test fixtures ship, is passed in as is.

Columns chrom (or chr), start and end are required. score and strand are used when present. The per-junction attributes motif, uniquely_mapped, multi_mapped, maximum_spliced_alignment_overhang and annotated_junction are packed into the bed name column, which is where the track reads its filters and labels from; supply the ones you want to filter or label on. A name column, if you have already packed it yourself, is written through untouched.

Without uniquely_mapped or an explicit score the arcs all draw at the same thickness: igv.js has nothing to size them by.

Usage

```
loadSpliceJunctionTrackFromLocalData(
  session,
  id,
  trackName,
  tbl,
  trackHeight = 100,
  displayMode = "COLLAPSED",
  deleteTracksOfSameName = TRUE,
  trackConfig = list()
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
tbl	data.frame, with at least "chrom" "start" "end" columns, in bed coordinates: a 0-based start and an exclusive end
trackHeight	an integer, 100 (pixels) by default
displayMode	character, "COLLAPSED", "EXPANDED" or "SQUISHED"
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options, the junction ones among them; see loadSpliceJunctionTrackFromURL

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadSpliceJunctionTrackFromURL

load a splice junction track served up by http

Description

load splice junctions from a BED file reachable by URL, as written by STAR (SJ.out.tab converted to BED). Six columns, with the per-junction attributes packed into the name column as key=value pairs separated by semicolons: motif, uniquely_mapped, multi_mapped, maximum_spliced_alignment_overlap, and annotated_junction. The track reads its filters and labels from those attributes.

igv.js draws junctions from a file of that shape only - it derives none of them from a bam file, and it has no sashimi plot. Show this track above a coverage track (bigWig, bedGraph) for the same sample to get the arcs-over-coverage view sashimi plots are wanted for.

Usage

```
loadSpliceJunctionTrackFromURL(
  session,
  id,
  trackName,
  url,
  indexURL = "",
  trackHeight = 100,
  displayMode = "COLLAPSED",
  deleteTracksOfSameName = TRUE,
  trackConfig = list()
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string

url	character string http url for the bed file of junctions
indexURL	character string http url for a tabix index, needed only for a bgzipped bed; "" by default, which loads the file whole
trackHeight	an integer, 100 (pixels) by default
displayMode	character, "COLLAPSED", "EXPANDED" or "SQUISHED"
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options. The junction ones are read straight off it: minUniquelyMappedReads, minTotalReads, maxFractionMultiMappedReads, minSplicedAlignmentOverhang, thicknessBasedOn, bounceHeightBasedOn, colorBy, labelWith, hideAnnotatedJunctions, hideUnannotatedJunctions, hideMotifs among them.

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

loadVcfTrack	<i>load a VCF (variant) track provided as a Bioconductor VariantAnnotation object</i>
--------------	---

Description

load a VCF (variant) track provided as a Bioconductor VariantAnnotation object

Usage

```
loadVcfTrack(
  session,
  id,
  trackName,
  vcfData,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackName	character string
vcfData	VariantAnnotation object
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options.
validateReference	logical flag, default TRUE: checks track contigs against the active reference genome and emits warnings/toasts on mismatch.

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "local-data.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

parseAndValidateGenomeSpec

parseAndValidateGenomeSpec

Description

a helper function for internal use by the igvShiny constructor, but possible also of use to those building an igvShiny app, to test their genome specification for validity

Usage

```
parseAndValidateGenomeSpec(
  genomeName,
  initialLocus = "all",
  stockGenome = TRUE,
  dataMode = NA,
  fasta = NA,
  fastaIndex = NA,
  genomeAnnotation = NA
)
```

Arguments

genomeName	character usually one short code of a supported ("stock") genome (e.g., "hg38") or for a user-supplied custom genome, the name you wish to use
initialLocus	character default "all", otherwise "chrN:start-end" or a recognized gene symbol
stockGenome	logical default TRUE
dataMode	character either "stock", "localFile" or "http"
fasta	character when supplying a custom (non-stock) genome, either a file path or a URL
fastaIndex	character when supplying a custom (non-stock) genome, either a file path or a URL, essential for all but the very small custom genomes.
genomeAnnotation	character when supplying a custom (non-stock) genome, a file path or URL pointing to a genome annotation file in a gff3 format

Value

an options list directly usable by igvApp.js, and thus igv.js

See Also

[get_css_genomes()] for stock genomes we support.

Examples

```
genomeSpec <-
  parseAndValidateGenomeSpec("hg38", "APOE") # the simplest case
# a custom genome, read from the files shipped with the package: every
# resource is checked for existence, and "http" would check it over the
# network on each R CMD check
base.dir <-
  system.file(package = "igvShiny", "extdata", "sarsGenome")
fasta.file <-
  file.path(base.dir, "Sars_cov_2.ASM985889v3.dna.toplevel.fa")
fastaIndex.file <- paste0(fasta.file, ".fai")
annotation.file <-
  file.path(base.dir, "Sars_cov_2.ASM985889v3.101.gff3")
custom.genome.title <- "SARS-CoV-2"
genomeOptions <-
  parseAndValidateGenomeSpec(
    genomeName = custom.genome.title,
    initialLocus = "all",
    stockGenome = FALSE,
    dataMode = "localFiles",
    fasta = fasta.file,
    fastaIndex = fastaIndex.file,
    genomeAnnotation = annotation.file
  )
```

removeTracksByName	<i>remove tracks from the browser</i>
--------------------	---------------------------------------

Description

delete tracks on the browser

Usage

```
removeTracksByName(session, id, trackNames)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance
trackNames	a vector of character strings

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

removeUserAddedTracks	<i>remove only those tracks explicitly added by your app</i>
-----------------------	--

Description

remove only those tracks explicitly added by your app. stock tracks (i.e., #' Refseq Genes) remain

Usage

```
removeUserAddedTracks(session, id)
```

Arguments

session	an environment or list, provided and managed by shiny
id	character string, the html element id of this widget instance

Value

nothing

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

renderIgvShiny	<i>draw the igv genome browser element</i>
----------------	--

Description

This function is called in the server function of your shiny app

Usage

```
renderIgvShiny(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	an expression that generates an HTML widget
env	the environment in which to evaluate expr
quoted	logical flag indicating if expr a quoted expression

Value

an output or render function that enables the use of the widget within Shiny applications

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

serveLocalFile

Serve a local file via HTTP Range Requests from the Shiny session

Description

Registers a local file with the active Shiny session so that `igv.js` can stream chunks of the file using HTTP 206 Partial Content requests. This allows random-access binary formats (such as BAM, BAI, CRAM, CRAI, BigWig, and Tabix-indexed files) of arbitrary size to be viewed in `igvShiny` without loading them into R memory or copying them to temporary directories.

Usage

```
serveLocalFile(session, filePath)
```

Arguments

<code>session</code>	Shiny session object (must be an active Shiny session)
<code>filePath</code>	character string, path to an existing, readable local file

Details

The returned URL is relative to the Shiny application root and is routed through the same HTTP port as the Shiny session, ensuring compatibility with Posit Connect, shinyapps.io, reverse proxies, and local development.

Value

A character string containing the relative URL to access the file from the browser (e.g. `"session/<token>/dataobj/<filename>"`).

Examples

```
## Not run:
# Inside a Shiny server function:
server <- function(input, output, session) {
  bam_url <- serveLocalFile(session, "data/sample.bam")
  bai_url <- serveLocalFile(session, "data/sample.bam.bai")

  tracks <- list(
    list(
      name = "Sample BAM",
      type = "alignment",
      format = "bam",
      url = bam_url,
      indexURL = bai_url
    )
  )
  output$igv <- renderIgvShiny({
    igvShiny(genomeOptions, tracks = tracks)
  })
}
```

```
}
## End(Not run)
```

show, GWASTrack-method *Display a short summary of a GWASTrack object*

Description

Display a short summary of a GWASTrack object

Usage

```
## S4 method for signature 'GWASTrack'
show(object)
```

Arguments

object An object of class GWASTrack

Value

nothing; called for the side effect of printing a summary

Examples

```
file <-
  system.file(package = "igvShiny", "extdata", "gwas-5k.tsv.gz")
tbl.gwas <- read.table(file, sep = "\t", header = TRUE, quote = "")
track <-
  GWASTrack("gwas 5k", tbl.gwas, chrom.col = 12, pos.col = 13, pval.col = 28)
track
```

showGenomicRegion *focus igv on a region*

Description

zoom in or out to show the nominated region, by chromosome locus or gene symbol
return the current region displayed by your igv instance

Usage

```
showGenomicRegion(session, id, region)

getGenomicRegion(session, id)
```

Arguments

<code>session</code>	an environment or list, provided and managed by shiny
<code>id</code>	character string, the html element id of this widget instance
<code>region</code>	a character string, either e.g. "chr5:92,221,640-92,236,523" or "MEF2C"

Value

a character string of format "chrom:start-end"

Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "showcase", "igvShinyDemo.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

Index

- * **GWASTrack_class**
 - .sanitizeChromosomeColorMap, [4](#)
 - display, GWASTrack-method, [6](#)
 - getUrl, GWASTrack-method, [9](#)
 - GWASTrack-class, [10](#)
 - show, GWASTrack-method, [37](#)
- * **igvShiny**
 - .packJunctionAttributes, [3](#)
 - .sanitizeAndMergeOptions, [3](#)
 - .sanitizeTrack, [4](#)
 - .sanitizeTracks, [5](#)
 - igvShiny, [12](#)
 - igvShinyOutput, [13](#)
 - parseAndValidateGenomeSpec, [32](#)
 - removeTracksByName, [34](#)
 - removeUserAddedTracks, [34](#)
 - renderIgvShiny, [35](#)
 - showGenomicRegion, [37](#)
- * **track_loaders**
 - loadBamTrackFromLocalData, [14](#)
 - loadBamTrackFromLocalFile, [15](#)
 - loadBamTrackFromURL, [16](#)
 - loadBedGraphTrack, [17](#)
 - loadBedGraphTrackFromURL, [19](#)
 - loadBedTrack, [20](#)
 - loadCramTrackFromLocalData, [21](#)
 - loadCramTrackFromURL, [23](#)
 - loadGFF3TrackFromLocalData, [24](#)
 - loadGFF3TrackFromURL, [25](#)
 - loadGwasTrack, [26](#)
 - loadSegTrack, [28](#)
 - loadSpliceJunctionTrackFromLocalData, [29](#)
 - loadSpliceJunctionTrackFromURL, [30](#)
 - loadVcfTrack, [31](#)
- * **utils**
 - checkReferenceCompatibility, [5](#)
 - get_basic_genomes, [7](#)
 - get_cas_genomes, [8](#)
 - get_css_genomes, [8](#)
 - get_tracks_dir, [9](#)
 - serveLocalFile, [36](#)
 - .GWASTrack (GWASTrack-class), [10](#)
 - .packJunctionAttributes, [3](#)
 - .sanitizeAndMergeOptions, [3](#)
 - .sanitizeChromosomeColorMap, [4](#)
 - .sanitizeTrack, [4](#)
 - .sanitizeTracks, [5](#)
 - checkReferenceCompatibility, [5](#)
 - display (display, GWASTrack-method), [6](#)
 - display, GWASTrack-method, [6](#)
 - get_basic_genomes, [7](#)
 - get_cas_genomes, [8](#)
 - get_css_genomes, [8](#)
 - get_tracks_dir, [9](#)
 - getGenomicRegion (showGenomicRegion), [37](#)
 - getUrl (getUrl, GWASTrack-method), [9](#)
 - getUrl, GWASTrack-method, [9](#)
 - GWASTrack, [27](#)
 - GWASTrack (GWASTrack-class), [10](#)
 - GWASTrack-class, [10](#)
 - igvShiny, [12](#)
 - igvShinyOutput, [13](#)
 - loadBamTrackFromLocalData, [14](#)
 - loadBamTrackFromLocalFile, [15](#)
 - loadBamTrackFromURL, [14](#), [16](#), [22](#), [23](#)
 - loadBedGraphTrack, [17](#)
 - loadBedGraphTrackFromURL, [19](#)
 - loadBedTrack, [20](#)
 - loadCramTrackFromLocalData, [21](#)
 - loadCramTrackFromLocalFile
 - (loadCramTrackFromLocalData), [21](#)
 - loadCramTrackFromURL, [23](#)

loadGenomeAnnotationTrack
 (loadBedGraphTrack), [17](#)
loadGFF3TrackFromLocalData, [24](#)
loadGFF3TrackFromURL, [25](#)
loadGwasTrack, [26](#)
loadSEGTrack (loadSegTrack), [28](#)
loadSegTrack, [28](#)
loadSpliceJunctionTrackFromLocalData,
 [29](#)
loadSpliceJunctionTrackFromURL, [29](#), [30](#)
loadVcfTrack, [31](#)

parseAndValidateGenomeSpec, [6](#), [32](#)

removeTracksByName, [34](#)
removeUserAddedTracks, [34](#)
renderIgvShiny, [35](#)

serveLocalFile, [36](#)
show, GWASTrack-method, [37](#)
showGenomicRegion, [37](#)