Package 'igvShiny'

November 13, 2026

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

Version 1.9.52

Date 2026-11-05

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' 'tracks.R' 'igvShiny.R' 'packages.R' 'utils.R' 'zzz.R'

RoxygenNote 7.3.3

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/igvShiny

git_branch devel

git_last_commit 69da476

git_last_commit_date 2026-11-05

Repository Bioconductor 3.24

Date/Publication 2026-11-12

Author Paul Shannon [aut],
Arkadiusz Gladki [aut, cre] (ORCID:
<https://orcid.org/0000-0002-7059-6378>),
Karolina Scigocka [aut],
Carolina Heimann [ctb],
Steffen Klasberg [ctb],
Vincent Carey [ctb],
Parv Sachdeva [ctb],
Mateusz Gladki [ctb],
R Consortium [fnd]

Maintainer Arkadiusz Gladki <gladki.arkadiusz@gmail.com>

Contents

.alignmentConfig 4
.bamLocalDataTrackPayload 4
.bamUrlTrackPayload 5
.bed12Columns 6
.bedGraphTrackPayload 6
.bedGraphUrlTrackPayload 7
.bedpeTrackPayload 8
.bedTrackPayload 9
.configListTrackPayload 9
.cramUrlTrackPayload 10
.extractBamContigs 11
.extractBedContigs 11
.extractCramContigs 12
.extractVcfContigs 12
.getGenomeSpec 13
.getReferenceContigs 13
.gff3TrackPayload 14
.gff3UrlTrackPayload 15
.granges2bedGraphTable 16
.granges2bedTable 16
.gwasTrackObjectPayload 17
.gwasTrackPayload 17
.identifyLikelyAssembly 18
.normalizeBedColumns 18
.normalizeBedGraphColumns 19
.normalizeJunctionColumns 19
.packJunctionAttributes 20
.registerGenomeSpec 20
.registerTrackFile 21
.sanitizeAndMergeOptions 21
.sanitizeChromosomeColorMap 22
.sanitizeTrack 22
.sanitizeTracks 23
.segTrackPayload 23
.sendTrack 24
.serveFileWithHttpRange 24
.slotsOverConfig 25

.spliceJunctionTrackPayload . . . 25
.spliceJunctionUrlTrackPayload . . . 26
.stageTrackFile . . . 27
.trackFile . . . 28
.trackPayload . . . 28
.trackReferenceData . . . 29
.tracksDir . . . 29
.trackTable . . . 30
.validateTrackReference . . . 30
.vcfTrackPayload . . . 31
BedpeInteractionsTrack-class . . . 31
checkReferenceCompatibility . . . 32
display,GWASTrack-method . . . 33
GenomicAlignmentTrack-class . . . 34
getUrl,GWASTrack-method . . . 35
get_basic_genomes . . . 36
get_cas_genomes . . . 37
get_css_genomes . . . 37
get_tracks_dir . . . 38
GFF3Track-class . . . 38
GWASTrack-class . . . 40
igvAnnotationTrack . . . 41
igvAnnotationTrack-class . . . 42
igvShiny . . . 44
igvShinyOutput . . . 45
loadBamTrackFromLocalData . . . 46
loadBamTrackFromLocalFile . . . 47
loadBamTrackFromURL . . . 48
loadBedGraphTrack . . . 49
loadBedGraphTrackFromURL . . . 50
loadBedTrack . . . 52
loadCramTrackFromLocalData . . . 53
loadCramTrackFromURL . . . 54
loadGFF3TrackFromLocalData . . . 55
loadGFF3TrackFromURL . . . 56
loadGwasTrack . . . 57
loadSegTrack . . . 58
loadSpliceJunctionTrackFromLocalData . . . 59
loadSpliceJunctionTrackFromURL . . . 61
loadTrack . . . 62
loadVcfTrack . . . 65
parseAndValidateGenomeSpec . . . 66
QuantitativeTrack-class . . . 67
removeTracksByName . . . 69
removeUserAddedTracks . . . 70
renderIgvShiny . . . 71
serveLocalFile . . . 71
show,GWASTrack-method . . . 72
show,Track-method . . . 73
showGenomicRegion . . . 73
Track-class . . . 75
trackInfo . . . 76

VariantTrack-class 77

Index **79**

`.alignmentConfig` *Slot options and display mode of a BAM alignment track*

Description

The BAM builders set `displayMode` themselves, with "EXPANDED" as default, so a `displayMode` in trackConfig would be a conflict there and get dropped. It goes to the builder argument instead; the remaining keys are merged under the slot options once, by `.slotsOverConfig`, and sanitized once, by the builder.

Usage

```
.alignmentConfig(options, trackConfig)
```

Arguments

`options`	named list of slot options
`trackConfig`	named list of user track options

Value

a list with `displayMode` and `trackConfig`

`.bamLocalDataTrackPayload`
Build the payload of a bam track from in-memory alignments

Description

The browser-free half of `loadBamTrackFromLocalData`: exports `data` to `path` as bam with `rtracklayer::export`, which also writes the index `paste0(path, ".bai")`, and returns the message body the `loadBamTrackFromLocalData` handler in igvShiny.js reads. Needs rtracklayer and Rsamtools (Suggests).

Usage

```
.bamLocalDataTrackPayload(
  trackName,
  data,
  path,
  displayMode = "EXPANDED",
  trackHeight = NULL,
  trackConfig = list()
)
```

Arguments

trackName	character string
data	a GAlignments object (GenomicAlignments)
path	character string, the bam file to write; its basename is what the browser fetches from the tracks/ resource path
displayMode	character string, "EXPANDED" (default), "SQUISHED" or "COLLAPSED"
trackHeight	numeric track height in pixels, or NULL for the igv.js default
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, the bam and its index

---

.bamUrlTrackPayload *Build the payload of a bam track read by url*

Description

The browser-free half of loadBamTrackFromURL and loadBamTrackFromLocalFile: returns the message body the loadBamTrackFromURL handler in igvShiny.js reads. The urls are used as given (remote, or app-relative from serveLocalFile()); nothing is written, so files is empty.

Usage

```
.bamUrlTrackPayload(
  trackName,
  bamURL,
  indexURL,
  displayMode = "EXPANDED",
  showAllBases = FALSE,
  trackHeight = NULL,
  trackConfig = list()
)
```

Arguments

trackName	character string
bamURL	character string, url of the bam file
indexURL	character string, url of the bam index
displayMode	character string, "EXPANDED" (default), "SQUISHED" or "COLLAPSED"
showAllBases	logical, or NULL to leave the key out of the message (what loadBamTrackFromLocalFile has always sent)
trackHeight	numeric track height in pixels, or NULL for the igv.js default
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, character(0)

.bed12Columns *The BED columns 7-12 of a GRanges imported from BED12*

Description

rtracklayer keeps BED columns 7-12 as the metadata columns `thick` (IRanges), `itemRgb` (character colors) and `blocks` (IRangesList, positions relative to the feature start). Writing them back keeps the exon structure of a transcript; without them igv.js draws one solid feature. Missing columns get the BED defaults.

Usage

```
.bed12Columns(gr, meta, tbl)
```

Arguments

`gr`	a GRanges
`meta`	its metadata columns
`tbl`	the BED6 table already built from `gr`

Value

a data.frame with thickStart, thickEnd, itemRgb, blockCount, blockSizes, blockStarts

.bedGraphTrackPayload *Build the payload of a bedGraph track from a data.frame*

Description

The browser-free half of `loadBedGraphTrack`: checks and sorts the table and returns the message body the `loadBedGraphTrack` handler in igvShiny.js reads. The features travel inline as JSON, so `files` is empty.

Usage

```
.bedGraphTrackPayload(
  trackName,
  tbl,
  color = "gray",
  trackHeight = 30,
  autoscale,
  autoscaleGroup = -1,
  min = NA_real_,
  max = NA_real_,
  trackConfig = list()
)
```

Arguments

trackName	character string
tbl	data.frame, with at least "chrom" "start" "end" "score" columns
color	character string, a legal CSS color, or "random"
trackHeight	an integer, 30 (pixels) by default
autoscale	logical
autoscaleGroup	numeric(1), -1 (no group) by default
min	numeric, consulted when autoscale is FALSE
max	numeric, consulted when autoscale is FALSE
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with `config`, the message body without `elementID`, and `files`, `character(0)`

.bedGraphUrlTrackPayload

Build the payload of a quantitative track read from a URL

Description

The browser-free half of `loadBedGraphTrackFromURL`: returns the message body the `loadBedGraphTrackFromURL` handler in igvShiny.js reads. Nothing is written, so `files` is empty.

Usage

```
.bedGraphUrlTrackPayload(
  trackName,
  url,
  color = "gray",
  trackHeight = 30,
  autoscale = TRUE,
  min = 0,
  max = 1,
  autoscaleGroup = -1,
  trackConfig = list()
)
```

Arguments

trackName	character string
url	character string, a bedGraph, wig or bigwig URL
color	character string, a legal CSS color, or "random"
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), -1 (no group) by default
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, character(0)

.bedpeTrackPayload *Build the payload of a bedpe interaction track from a data.frame*

Description

The browser-free builder of BedpeInteractionsTrack: sorts the table by chrom1 and start1, writes it to path as headerless bedpe, and returns the message body the loadBedpeTrackFromFile handler in igvShiny.js reads. It has no exported data.frame loader; loadTrack() is its sender.

Usage

```
.bedpeTrackPayload(
  trackName,
  tbl,
  path,
  color = "darkBlue",
  trackHeight = 50,
  trackConfig = list()
)
```

Arguments

trackName	character string
tbl	data.frame, bedpe: chrom1, start1, end1, chrom2, start2, end2, then optional columns
path	character string, the file to write; its basename is what the browser fetches from the tracks/ resource path
color	character string, a legal CSS color, or "random"
trackHeight	numeric, pixels
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, the paths written (here: path)

.bedTrackPayload *Build the payload of a bed track from a data.frame*

Description

The browser-free half of `loadBedTrack`: checks and sorts the table, writes it to `path` as headerless bed, and returns the message body the `loadBedTrackFromFile` handler in igvShiny.js reads. It touches no session; the caller picks the path (in a sender, with `.trackFile()`) and adds the element id.

Usage

```
.bedTrackPayload(
  trackName,
  tbl,
  path,
  color = "",
  trackHeight = 50,
  trackConfig = list()
)
```

Arguments

`trackName`	character string
`tbl`	data.frame, with at least "chrom" (or "chr") "start" "end" columns
`path`	character string, the file to write; its basename is what the browser fetches from the `tracks/` resource path
`color`	character string, a legal CSS color, or "random"
`trackHeight`	an integer, 50 (pixels) by default
`trackConfig`	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with `config`, the message body without `elementID`, and `files`, the paths written (here: `path`)

.configListTrackPayload *Build the payload of a track given as a plain igv.js track config*

Description

The browser-free half of `loadTrack(<list>)`. The config goes through `.sanitizeAndMergeOptions()` like every loader's `trackConfig`: keys igv.js does not read are dropped with a warning. The genotype colors are allowed for a variant track: `type "variant"`, or no type and a vcf `format`, as igv.js infers it.

Usage

```
.configListTrackPayload(track)
```

Arguments

track	a named list, an igv.js track configuration with at least name, url and one of type or format

Value

list(config = the message body, files = character(0))

.cramUrlTrackPayload *Build the payload of a cram track read by url*

Description

The browser-free half of loadCramTrackFromURL and loadCramTrackFromLocalData: returns the message body the loadCramTrackFromURL handler in igvShiny.js reads. The urls are used as given (remote, or app-relative from serveLocalFile()); nothing is written, so files is empty.

Usage

```
.cramUrlTrackPayload(
  trackName,
  cramURL,
  indexURL,
  trackHeight = NULL,
  trackConfig = list()
)
```

Arguments

trackName	character string
cramURL	character string, url of the cram file
indexURL	character string, url of the crai index
trackHeight	numeric track height in pixels, or NULL for the igv.js default
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, character(0)

.extractBamContigs *Extract contig information from a BAM file*

Description

Extract contig information from a BAM file

Usage

```
.extractBamContigs(bamFile)
```

Arguments

bamFile	Path to a readable .bam file

Value

List with contigs (named integer vector) and assembly (string or NULL)

.extractBedContigs *Extract contig and coordinate information from a BED data.frame or file*

Description

Extract contig and coordinate information from a BED data.frame or file

Usage

```
.extractBedContigs(tbl)
```

Arguments

tbl	A data.frame with chr/chrom, start, end columns, or file path

Value

List with contigs (named integer vector) and maxCoords (named integer vector)

.extractCramContigs *Extract contig information from a CRAM file*

Description

Extract contig information from a CRAM file

Usage

```
.extractCramContigs(cramFile)
```

Arguments

cramFile	Path to a readable .cram file

Value

List with contigs (named integer vector) and assembly (string or NULL)

.extractVcfContigs *Extract contig information from a VCF object or file*

Description

Extract contig information from a VCF object or file

Usage

```
.extractVcfContigs(vcfDataOrFile)
```

Arguments

vcfDataOrFile	A VariantAnnotation VCF object, or a path to a .vcf / .vcf.gz file

Value

List with contigs (named integer vector) and assembly (string or NULL)

.getGenomeSpec *Retrieve active genome spec for an igvShiny widget instance*

Description

Retrieve active genome spec for an igvShiny widget instance

Usage

```
.getGenomeSpec(session, id = NULL)
```

Arguments

session	Shiny session or NULL
id	Character string, the widget element id

Value

Named list of genome options, or NULL

.getReferenceContigs *Extract contig names and lengths from genome options*

Description

Extract contig names and lengths from genome options

Usage

```
.getReferenceContigs(genomeSpec)
```

Arguments

genomeSpec	Named list of genome options, or character string (genome name)

Value

Named integer vector of contig lengths, or NULL

`.gff3TrackPayload` *Build the payload of a GFF3 track from a data.frame*

Description

The browser-free half of `loadGFF3TrackFromLocalData`: writes the table, header line included as before, to `path` and returns the message body the `loadGFF3TrackFromLocalData` handler in igvShiny.js reads.

Usage

```
.gff3TrackPayload(
  trackName,
  tbl.gff3,
  path,
  color = "gray",
  colorTable,
  colorByAttribute,
  displayMode,
  trackHeight = 50,
  visibilityWindow,
  trackConfig = list()
)
```

Arguments

`trackName`	character string
`tbl.gff3`	data.frame in standard 9-column GFF3 format
`path`	character string, the file to write; its basename is what the browser fetches from the `tracks/` resource path
`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
`trackConfig`	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with `config`, the message body without `elementID`, and `files`, the paths written (here: `path`)

`.gff3UrlTrackPayload` *Build the payload of a GFF3 track read by url*

Description

The browser-free half of `loadGFF3TrackFromURL`: returns the message body the `loadGFF3TrackFromURL` handler in igvShiny.js reads. Nothing is written, so `files` is empty. Like the loader, it has no defaults for the colour, display and window arguments.

Usage

```
.gff3UrlTrackPayload(
  trackName,
  gff3URL,
  indexURL,
  color = "gray",
  colorTable,
  colorByAttribute,
  displayMode,
  trackHeight = 50,
  visibilityWindow,
  trackConfig = list()
)
```

Arguments

`trackName`	character string
`gff3URL`	character string, url of the gff3 file
`indexURL`	character string, url of its index
`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
`trackConfig`	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with `config`, the message body without `elementID`, and `files`, `character(0)`

.granges2bedGraphTable *GRanges with one numeric metadata column to a bedGraph data.frame*

Description

GRanges with one numeric metadata column to a bedGraph data.frame

Usage

```
.granges2bedGraphTable(gr)
```

Arguments

gr	a GRanges (or UCSCData) with exactly one numeric metadata column

Value

a data.frame with chr, start (0-based), end, value

.granges2bedTable *GRanges to a six column bed data.frame*

Description

GRanges to a six column bed data.frame

Usage

```
.granges2bedTable(gr)
```

Arguments

gr	a GRanges (or UCSCData)

Value

a data.frame with chr, start (0-based), end, name, score, strand

.gwasTrackObjectPayload *Build the payload of a GWASTrack object*

Description

The browser-free half of `display()` (and of `loadTrack()` for a GWASTrack): returns the message body the `loadGwasTrackFlexibleSource` handler in igvShiny.js reads. The constructor already wrote a data.frame to the tracks directory; a local file is sent by its `tracks/` path. The builder writes nothing, so `files` is empty.

Usage

```
.gwasTrackObjectPayload(obj)
```

Arguments

`obj`	a GWASTrack object

Value

a list with `config`, the message body without `elementID`, and `files`, `character(0)`

.gwasTrackPayload *Build the payload of a GWAS track from a data.frame*

Description

The browser-free half of `loadGwasTrack`: writes the table with its header to `path` and returns the message body the `loadGwasTrack` handler in igvShiny.js reads.

Usage

```
.gwasTrackPayload(
  trackName,
  tbl.gwas,
  path,
  ymin = 0,
  ymax = 35,
  trackConfig = list()
)
```

Arguments

`trackName`	character string
`tbl.gwas`	data.frame, the GWAS table
`path`	character string, the file to write; its basename is what the browser fetches from the `tracks/` resource path
`ymin`	numeric defaults to 0

ymax	numeric defaults to 35
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, the paths written (here: path)

`.identifyLikelyAssembly` *Infer the likely assembly of a set of contigs from lengths or metadata*

Description

Infer the likely assembly of a set of contigs from lengths or metadata

Usage

```
.identifyLikelyAssembly(contigs, assemblyStr = NULL)
```

Arguments

contigs	Named integer vector of contig lengths
assemblyStr	Optional string from header

Value

String describing likely assembly or NULL

`.normalizeBedColumns` *Give a bed-like data.frame the chr/start/end column names loaders expect*

Description

Renames a leading "chrom" column to "chr" and fails when the first three columns are not chr, start and end.

Usage

```
.normalizeBedColumns(tbl)
```

Arguments

tbl	data.frame, with at least "chrom" (or "chr") "start" "end" columns

Value

the data.frame, its first column possibly renamed

.normalizeBedGraphColumns

Give a bedGraph-like data.frame the chr/start/end/value column names

Description

Renames a leading "chrom" column to "chr" and the fourth column to "value", then applies the chr/start/end check of `.normalizeBedColumns`.

Usage

```
.normalizeBedGraphColumns(tbl)
```

Arguments

`tbl`	data.frame, with at least four columns: chrom (or chr), start, end and a numeric score

Value

the data.frame, with its columns renamed

.normalizeJunctionColumns

Give a junction data.frame the chr/start/end column names and check them

Description

Renames a "chrom" column to "chr" and fails when chr, start or end is missing or of the wrong type.

Usage

```
.normalizeJunctionColumns(tbl)
```

Arguments

`tbl`	data.frame, with at least "chrom" (or "chr") "start" "end" columns

Value

the data.frame, its "chrom" column possibly renamed

.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.

.registerGenomeSpec *Register active genome spec for an igvShiny widget instance*

Description

Register active genome spec for an igvShiny widget instance

Usage

```
.registerGenomeSpec(session, elementId, genomeOptions)
```

Arguments

session	Shiny session or NULL
elementId	Character string or NULL
genomeOptions	Named list of genome options

.registerTrackFile *Have a path removed when the session ends*

Description

The companion of .trackFile, for the files a loader does not name itself: rtracklayer::export(format = "BAM") writes an index next to the bam it was given, and that index is served and leaks the same way. The path need not exist yet, or ever - unlink() does not mind.

Usage

```
.registerTrackFile(session, path)
```

Arguments

session	a shiny session object
path	character string, the path to remove when the session ends

Value

the path, unchanged

.sanitizeAndMergeOptions *Sanitize and merge track configuration options*

Description

Sanitize and merge track configuration options

Usage

```
.sanitizeAndMergeOptions(
  baseOptions,
  userOptions,
  allowed = .validIgvTrackOptions
)
```

Arguments

baseOptions	A list of default options set by the R function.
userOptions	A list of options provided by the user via trackConfig.
allowed	character, the option names igv.js reads for this kind of track; defaults to the global allowlist

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 *Sanitize a list of startup track specifications*

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.

.segTrackPayload *Build the payload of a seg track from a data.frame*

Description

The browser-free half of loadSegTrack: returns the message body the loadSegTrack handler in igvShiny.js reads. The segments travel inline as JSON, so files is empty.

Usage

```
.segTrackPayload(trackName, tbl, trackConfig = list())
```

Arguments

trackName	character string
tbl	data.frame of segments, written through as is
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, character(0)

.sendTrack *The sender shared by the loadTrack methods*

Description

The sender shared by the loadTrack methods

Usage

```
.sendTrack(
  session,
  id,
  track,
  handler,
  ext,
  deleteTracksOfSameName,
  trackConfig,
  validateReference
)
```

Arguments

session	a shiny session object
id	character string, the html element id of this widget instance
track	an object of a Track subclass
handler	character string, the igvShiny.js message handler
ext	character string, the extension of the file the builder writes, or NULL when it writes none
deleteTracksOfSameName	logical
trackConfig	a named list of additional igv.js track options
validateReference	logical

Value

nothing

.serveFileWithHttpRange *Internal HTTP Range request handler for local files*

Description

Internal HTTP Range request handler for local files

Usage

```
.serveFileWithHttpRange(filePath, req)
```

Arguments

filePath	character, normalized path to existing local file
req	environment, Rook request environment supplied by Shiny

Value

An httpResponse object (status 206 for Range requests, 200 otherwise)

.slotsOverConfig *Merge track slot options over a user trackConfig*

Description

Drops NA slot values, then drops (with a warning) the trackConfig keys the remaining slots set, so slot values win.

Usage

```
.slotsOverConfig(slotOptions, trackConfig)
```

Arguments

slotOptions	named list of track options taken from slots
trackConfig	named list of user track options

Value

a named list, the slot options followed by the user options

.spliceJunctionTrackPayload *Build the payload of a splice junction track from a data.frame*

Description

The browser-free half of loadSpliceJunctionTrackFromLocalData: writes the six-column junction bed to path and returns the message body of .spliceJunctionUrlTrackPayload pointed at it, which is what the loadSpliceJunctionTrackFromURL handler in igvShiny.js reads.

Usage

```
.spliceJunctionTrackPayload(
  trackName,
  tbl,
  path,
  trackHeight = 100,
  displayMode = "COLLAPSED",
  trackConfig = list()
)
```

Arguments

trackName	character string
tbl	data.frame, with at least "chrom" (or "chr") "start" "end" columns, in bed coordinates; see loadSpliceJunctionTrackFromLocalData for the optional columns
path	character string, the file to write; its basename is what the browser fetches from the tracks/ resource path
trackHeight	an integer, 100 (pixels) by default
displayMode	character, "COLLAPSED", "EXPANDED" or "SQUISHED"
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, the paths written (here: path)

.spliceJunctionUrlTrackPayload

Build the payload of a splice junction track read by url

Description

The browser-free half of loadSpliceJunctionTrackFromURL (and, with an app-relative url, of loadSpliceJunctionTrackFromLocalData): returns the message body the loadSpliceJunctionTrackFromURL handler in igvShiny.js reads. Nothing is written, so files is empty.

Usage

```
.spliceJunctionUrlTrackPayload(
  trackName,
  url,
  indexURL = "",
  trackHeight = 100,
  displayMode = "COLLAPSED",
  trackConfig = list()
)
```

Arguments

trackName	character string
url	character string, url of the junction bed file
indexURL	character string, url of a tabix index, or "" for none; anything but a single non-NA string is turned into "" with a warning
trackHeight	an integer, 100 (pixels) by default
displayMode	character, "COLLAPSED", "EXPANDED" or "SQUISHED"
trackConfig	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with config, the message body without elementID, and files, character(0)

.stageTrackFile *Make a file on disk reachable by igv.js*

Description

Files igv.js reads by url have to sit in the directory shiny serves as "tracks". Alignment files run to gigabytes, so the file is linked rather than copied where the filesystem allows it (windows, and any mount refusing links, falls back to a copy). The name is randomized: two loaders may be handed same-named files from different directories.

Usage

```
.stageTrackFile(session, path, ext)
```

Arguments

session	a shiny session object; the staged file is removed when that session ends
path	character string, an existing file
ext	character string, the extension of the served file

Value

string with the path to the file, relative to the shiny app

.trackFile — *Name a file in the served tracks directory, and have it removed with the session that asked for it*

Description

Loaders write a file per track load and nothing ever removes it. In an interactive session that is harmless - the default directory sits under `tempdir()` and goes with the process - but a deployed app keeps one R process across many user sessions, and `TRACKS_DIR` may point outside `tempdir()` altogether, where nothing removes the files at all. Alignment loads make it concrete: a bam export or a cram copy is gigabytes.

The paths handed out to one session are collected in that session's `userData`, and the first call registers a single `session$onSessionEnded()` hook to unlink the set. Assignment goes through a local binding to the `userData` environment on purpose: `session$userData$x <- value` would call the assignment method of a module's session proxy, which refuses to be written to.

Usage

```
.trackFile(session, ext)
```

Arguments

`session`	a shiny session object
`ext`	character string, the extension of the served file

Value

string with the path to the file

.trackPayload — *Build the payload of a track object*

Description

The builder half of `loadTrack`: hands the slots of `track` to the data.frame builder of its family (`.bedTrackPayload` for annotation tracks, `.bedGraphTrackPayload` for quantitative tracks) and returns its `list(config, files)`. It touches no session; a builder that writes a file writes it to `path`.

Usage

```
.trackPayload(track, path = NULL, trackConfig = list())
```

Arguments

`track`	an object of a Track subclass
`path`	character string, the file to write, or NULL for builders that write none
`trackConfig`	a named list of additional igv.js track configuration options; slot values win over it

Value

a list with `config`, the message body without `elementID`, and `files`, the paths written

---

.trackReferenceData *What loadTrack checks against the reference genome*

Description

What `.validateTrackReference` gets for a track: the feature table for the data.frame families, the Bioconductor object for alignment and variant tracks, and NULL (nothing checked, as with the url loaders) for tracks that only point at urls.

Usage

```
.trackReferenceData(track)
```

Arguments

`track`	an object of a Track subclass

Value

a data.frame, a GAlignments or VCF object, or NULL

.tracksDir *Get the tracks directory and make sure shiny serves that very directory*

Description

The "tracks" resource path is registered once, when the package is loaded. The default directory lives under `tempdir()`, which is not guaranteed to stay put for the life of the process: when it moves, files are written to the new directory while shiny still serves the old one, and igv.js gets a 404 for a file that exists on disk. Re-registering the path before each write keeps the served directory and the written directory the same.

Usage

```
.tracksDir()
```

Value

string with the path to the tracks directory.

.trackTable *A track's features as the data.frame its payload builder takes*

Description

Annotation tracks give a bed table (chr, start, end, and for GRanges name, score, strand), quantitative tracks a bedGraph table (chr, start, end, value). GRanges starts are 1-based and become 0-based here.

Usage

```
.trackTable(track)
```

Arguments

track	an object of a Track subclass with a coreObject slot

Value

a data.frame

.validateTrackReference *Internal validator dispatched by track loaders*

Description

Internal validator dispatched by track loaders

Usage

```
.validateTrackReference(
  session,
  id,
  trackName,
  target,
  validateReference = TRUE
)
```

Arguments

session	Shiny session or NULL
id	Widget element ID
trackName	Display name of track
target	Track file path or data object
validateReference	Logical, whether validation is enabled

.vcfTrackPayload *Build the payload of a VCF track from a VariantAnnotation object*

Description

The browser-free half of `loadVcfTrack`: writes `vcfData` to `path` with `VariantAnnotation::writeVcf` and returns the message body the `loadVcfTrack` handler in igvShiny.js reads. Needs VariantAnnotation (Suggests).

Usage

```
.vcfTrackPayload(trackName, vcfData, path, trackConfig = list())
```

Arguments

`trackName`	character string
`vcfData`	a `VCF` object from VariantAnnotation
`path`	character string, the file to write; its basename is what the browser fetches from the `tracks/` resource path
`trackConfig`	a named list of additional igv.js track configuration options, merged onto the defaults

Value

a list with `config`, the message body without `elementID`, and `files`, the paths written (here: `path`)

BedpeInteractionsTrack-class *BedpeInteractionsTrack: paired-end interactions from a bedpe data.frame*

Description

A `DataFrameAnnotationTrack` whose table is bedpe: chrom1, start1, end1, chrom2, start2, end2 (0-based starts), optionally followed by name and score. igv.js draws it as an "interact" track of arcs between the two anchors. Display it with `loadTrack`; color and height are sent to igv.js.

Usage

```
BedpeInteractionsTrack(
  trackName,
  table,
  color = "darkBlue",
  trackHeight = 50,
  displayMode = "EXPANDED",
  visibilityWindow = 1e+05
)
```

Arguments

trackName	A character string, used as track label by igv
table	a bedpe data.frame, at least 6 columns
color	A CSS color name (e.g., "red" or "#FF0000"), or "random"
trackHeight	track height in pixels
displayMode	"COLLAPSED", "SQUISHED" or "EXPANDED"; stored, not sent
visibilityWindow	Maximum window size in base pairs; stored, not sent

Value

An object of class BedpeInteractionsTrack

Examples

```
f <- system.file(package = "igvShiny", "extdata", "sixColumn-demo1.bedpe")
tbl <- utils::read.table(f, sep = "\t", header = TRUE,
                         stringsAsFactors = FALSE)
track <- BedpeInteractionsTrack("interactions", tbl)
trackSize(track)
```

---

checkReferenceCompatibility

Check reference compatibility between track data and genome specification

---

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)
}
```

`GenomicAlignmentTrack-class`
Alignment tracks: in-memory reads, a bam url or a cram url

Description

`GenomicAlignmentTrack` holds a `GAlignments` object (GenomicAlignments, written to a bam file at display time; needs the suggested packages GenomicAlignments, rtracklayer and Rsamtools). `RemoteAlignmentTrack` and `CramTrack` point igv.js at a served bam or cram file and its index. Display them with `loadTrack`; color, height and visibilityWindow are sent to igv.js.

Usage

```
GenomicAlignmentTrack(
  trackName,
  alignment,
  trackHeight = 50,
  visibilityWindow = 30000,
  color = "gray"
)

RemoteAlignmentTrack(
  trackName,
  bamUrl,
  bamIndex = paste0(bamUrl, ".bai"),
  trackHeight = 50,
  visibilityWindow = 30000,
  color = "gray"
)

CramTrack(
  trackName,
  cramUrl,
  indexUrl,
  trackHeight = 50,
  visibilityWindow = 30000,
  color = "gray"
)
```

Arguments

trackName	A character string, used as track label by igv
alignment	a GAlignments object
trackHeight	track height in pixels
visibilityWindow	Maximum window size in base pairs for which reads are displayed
color	A CSS color name (e.g., "red" or "#FF0000"), or "random"
bamUrl	url of a bam file
bamIndex	url of its index; by default bamUrl with ".bai" appended
cramUrl	url of a cram file
indexUrl	url of its crai index

Value

An object of the named class

Slots

alignment a GAlignments object

bamUrl character, url of the bam file

bamIndex character, url of its index

cramUrl character, url of the cram file

indexUrl character, url of its crai index

Examples

```
track <- CramTrack("cram demo", "https://example.com/reads.cram",
                   "https://example.com/reads.cram.crai")
trackInfo(track)
```

---

getUrl,GWASTrack-method

the url of the gwas table

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)
```

`get_basic_genomes` *get_basic_genomes*

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 *get_cas_genomes*

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 *get_css_genomes*

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` *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*

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")
```

`GFF3Track-class` *GFF3Track: gene models from a GFF3 data.frame or url*

Description

A `GFF3Track` holds either a 9-column GFF3 table (`tbl`), served as a file, or the url of a GFF3 file and optionally its index. Features are colored by `trackColor`, or by the `colorByAttribute` attribute through `colorTable`. Display it with `loadTrack`.

Usage

```
GFF3Track(
  trackName,
  tbl.track = data.frame(),
  url = NA_character_,
  indexURL = NA_character_,
  trackColor = "black",
  colorByAttribute = NA_character_,
  colorTable = list(),
  displayMode = "EXPANDED",
  trackHeight = 50,
  visibilityWindow = 1e+05
)
```

Arguments

`trackName`	A character string, used as track label by igv
`tbl.track`	a data.frame in standard 9-column GFF3 format
`url`	url of a GFF3 file, used when `tbl.track` is empty
`indexURL`	url of its index
`trackColor`	A CSS color, used unless colorTable is given
`colorByAttribute`	name of a GFF3 attribute in column 9, e.g. "biotype"
`colorTable`	a named list mapping attribute values to colors
`displayMode`	"COLLAPSED", "SQUISHED" or "EXPANDED"
`trackHeight`	track height in pixels
`visibilityWindow`	Maximum window size in base pairs for which features are displayed

Value

An object of class GFF3Track

Slots

`tbl` data.frame, the GFF3 table, or an empty data.frame

`url` character, the GFF3 url, or NA

`indexURL` character, the index url, or NA

`colorByAttribute` character, a column 9 attribute, or NA

`colorTable` list, attribute value to color; a "default" entry is renamed "*", igv.js's fallback key

Examples

```
f <- system.file(package = "igvShiny", "extdata", "GRCh38.94.NDUFS2.gff3")
tbl <- utils::read.table(f, sep = "\t", header = FALSE, quote = "",
                         stringsAsFactors = FALSE)
track <- GFF3Track("NDUFS2", tbl, colorByAttribute = "biotype",
                   colorTable = list(protein_coding = "blue",
                                     default = "gray"))
trackSize(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. `list("1" = "red", "2" = "blue")`. 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 GWASTrack 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")
)
```

`igvAnnotationTrack` *Build an annotation track from whatever annotation object is at hand*

Description

igvR's igvAnnotationTrack() built a track that held no data. Here it picks the subclass that fits annotation: a data.frame gives a DataFrameAnnotationTrack, a UCSCData a UCSCBedAnnotationTrack and any other GRanges a GRangesAnnotationTrack.

Usage

```
igvAnnotationTrack(trackName, annotation, ...)
```

Arguments

trackName	A character string, used as track label by igv
annotation	a data.frame, GRanges or UCSCData
...	further arguments of the subclass constructor, e.g. color, displayMode, trackHeight

Value

an object of a subclass of igvAnnotationTrack

Examples

```
tbl <- data.frame(chrom = "chr1", start = 100, end = 200,
                  stringsAsFactors = FALSE)
class(igvAnnotationTrack("demo", tbl))
```

---

igvAnnotationTrack-class

Annotation tracks: bed-like features from a data.frame, GRanges or UCSCData

---

Description

igvAnnotationTrack adds the slots of igv.js annotation tracks to Track. Its subclasses carry the features in a coreObject slot: DataFrameAnnotationTrack (a bed-like data.frame, written as is), GRangesAnnotationTrack (a GRanges; 1-based starts become 0-based bed starts) and UCSCBedAnnotationTrack (an rtracklayer UCSCData, needs the suggested package rtracklayer). Display one with loadTrack, which sends displayMode, expandedRowHeight, squishedRowHeight, maxRows, searchable and visibilityWindow to igv.js. nameField is stored for igvR compatibility, not sent.

Usage

```
DataFrameAnnotationTrack(
  trackName,
  annotation,
  color = "",
  displayMode = "SQUISHED",
  trackHeight = 50,
  expandedRowHeight = 30,
  squishedRowHeight = 15,
```

```
  maxRows = 500,
  searchable = FALSE,
  visibilityWindow = 1e+05
)

GRangesAnnotationTrack(
  trackName,
  annotationData,
  color = "darkGrey",
  displayMode = "SQUISHED",
  trackHeight = 50,
  expandedRowHeight = 30,
  squishedRowHeight = 15,
  maxRows = 500,
  searchable = FALSE,
  visibilityWindow = 1e+05
)

UCSCBedAnnotationTrack(
  trackName,
  annotation,
  color = "darkGrey",
  displayMode = "SQUISHED",
  trackHeight = 50,
  expandedRowHeight = 30,
  squishedRowHeight = 15,
  maxRows = 500,
  searchable = FALSE,
  visibilityWindow = 1e+05
)
```

Arguments

trackName	A character string, used as track label by igv; we recommend unique names per track
annotation	for DataFrameAnnotationTrack, a data.frame with at least chrom (or chr), start and end columns, in bed (0-based) coordinates; for UCSCBedAnnotationTrack, a UCSCData object, e.g. from rtracklayer::import(format = "bed")
color	A CSS color name (e.g., "red" or "#FF0000"), or "random"; for a data.frame, leave the default "" if supplying bed9 with itemRgb
displayMode	"COLLAPSED", "SQUISHED" or "EXPANDED"
trackHeight	track height in pixels
expandedRowHeight	Height of each row of features in "EXPANDED" mode
squishedRowHeight	Height of each row of features in "SQUISHED" mode
maxRows	maximum number of feature rows to display
searchable	If TRUE, labels on annotation elements may be used in search
visibilityWindow	Maximum window size in base pairs for which features are displayed

annotationData a GRanges; mcols "name" and "score", when present, fill the bed name and score columns

Value

An object of the named class

Slots

classTypeOfSuppliedObject character

displayMode character, "COLLAPSED", "SQUISHED" or "EXPANDED"

expandedRowHeight numeric, row height in "EXPANDED" mode

squishedRowHeight numeric, row height in "SQUISHED" mode

nameField character; stored for igvR compatibility, not sent

maxRows numeric

searchable logical

coreObject the features: data.frame, GRanges or UCSCData

Examples

```
tbl <- data.frame(chrom = rep("chr5", 3),
                  start = c(88883100, 88883200, 88883350),
                  end = c(88883150, 88883220, 88883390),
                  name = c("a", "b", "c"),
                  stringsAsFactors = FALSE)
track <- DataFrameAnnotationTrack("data.frame demo", tbl, color = "red")
gr <- GenomicRanges::GRanges(c("chr5:100-149", "chr5:300-349"))
track2 <- GRangesAnnotationTrack("GRanges demo", gr)
trackSize(track2)
```

---

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: `list(list(name="genes", type="annotation", format="gff3",url="https://...` Unrecognized keys are dropped with a warning; see `.validIgvTrackOptions` 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 *create the UI for the widget*

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 *load a scored genome annotation track provided as a data.frame*

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` *load a bed track provided as a data.frame*

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

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 `columns = list(chromosome = 3, position = 4, value = 10)`; `colorTable` 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 *load a seg track provided as a data.frame*

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_o` 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`, `maxFractionMultiMappedReac` `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)
}
```

`loadTrack` *Display a track object in an igvShiny widget*

Description

`loadTrack` is the one loader for every track class: it turns `track` into a payload with the builder of its class and sends it to the browser, removing an earlier track of the same name first. Annotation tracks are served as a bed file, quantitative tracks travel inline, exactly as with `loadBedTrack` and `loadBedGraphTrack`.

`track` can also be a plain igv.js track configuration, a named list with at least `name`, `url` and one of `type` or `format`. It loads any track type igv.js reads from a url, e.g. a bigBed file, without a dedicated loader. Keys igv.js does not read are dropped with a warning, as with `trackConfig`; a `trackConfig` passed with a config list is ignored with a logged warning.

Usage

```
loadTrack(session, id, track, ...)

## S4 method for signature 'Track'
loadTrack(session, id, track, ...)

## S4 method for signature 'igvAnnotationTrack'
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'QuantitativeTrack'
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'GenomicAlignmentTrack'
```

```
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'RemoteAlignmentTrack'
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'CramTrack'
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'VariantTrack'
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'GFF3Track'
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'BedpeInteractionsTrack'
loadTrack(
  session,
```

```
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'GWASTrack'
loadTrack(
  session,
  id,
  track,
  deleteTracksOfSameName = TRUE,
  trackConfig = list(),
  validateReference = TRUE
)

## S4 method for signature 'list'
loadTrack(
  session,
  id,
  track,
  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
track	an object of a Track subclass, e.g. DataFrameAnnotationTrack or GRangesQuantitativeTrack, or a named list holding an igv.js track configuration
...	arguments of the methods: deleteTracksOfSameName (logical, default TRUE), trackConfig (a named list of additional igv.js track configuration options, merged onto the track's own) and validateReference (logical, default TRUE: check the track contigs and coordinates against the active reference genome)
deleteTracksOfSameName	logical, default TRUE
trackConfig	a named list of additional igv.js track configuration options
validateReference	logical flag, default TRUE

Value

nothing

Examples

```
tbl <- data.frame(chrom = "chr1", start = c(100, 300), end = c(200, 400),
                  stringsAsFactors = FALSE)
track <- DataFrameAnnotationTrack("demo", tbl, color = "red")
```

```
if (interactive()) {
  # inside a shiny server function, with an igvShiny output "igvShiny_0":
  # loadTrack(session, "igvShiny_0", track)
}

# any igv.js track type from a url, here a bigBed file; nothing is
# fetched until the browser draws the track
bigBed <- list(name = "genes", type = "annotation", format = "bigBed",
               url = "https://example.org/genes.bb")
if (interactive()) {
  # loadTrack(session, "igvShiny_0", bigBed)
}
```

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

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
  )
```

QuantitativeTrack-class

Quantitative tracks: scored intervals from a data.frame, GRanges or UCSCData

---

Description

`QuantitativeTrack` adds y-axis scaling to `Track`; it carries no data, so build tracks with its subclasses: `DataFrameQuantitativeTrack` (a data.frame of chrom, start, end, value in bedGraph coordinates), `GRangesQuantitativeTrack` (a `GRanges` with exactly one numeric metadata column) and `UCSCBedGraphQuantitativeTrack` (an `rtracklayer` `UCSCData` imported from bedGraph, needs the suggested package rtracklayer). Display one with `loadTrack`, which also sends `visibilityWindow` to igv.js.

Usage

```
QuantitativeTrack(
  trackName,
  quantitativeData,
  fileFormat = c("wig", "bigWig", "bedGraph", "gwas"),
  color = "gray",
  sourceType = c("file", "url"),
  trackHeight = 50,
```

```
  autoscale = TRUE,
  min = NA_real_,
  max = NA_real_,
  visibilityWindow = 1e+05
)

DataFrameQuantitativeTrack(
  trackName,
  quantitativeData,
  color = "blue",
  trackHeight = 50,
  autoscale,
  min = NA_real_,
  max = NA_real_,
  visibilityWindow = 1e+05
)

GRangesQuantitativeTrack(
  trackName,
  quantitativeData,
  color = "blue",
  trackHeight = 50,
  autoscale = TRUE,
  min = NA_real_,
  max = NA_real_,
  visibilityWindow = 1e+05
)

UCSCBedGraphQuantitativeTrack(
  trackName,
  quantitativeData,
  color = "blue",
  trackHeight = 50,
  autoscale = TRUE,
  min = NA_real_,
  max = NA_real_,
  visibilityWindow = 1e+05
)
```

Arguments

trackName	A character string, used as track label by igv; we recommend unique names per track
quantitativeData	the scored intervals; ignored by QuantitativeTrack, which only builds the shared part
fileFormat	One of "wig", "bigWig", "bedGraph", "gwas"
color	A CSS color name (e.g., "red" or "#FF0000"), or "random"
sourceType	"file" or "url"
trackHeight	track height in pixels
autoscale	Autoscale track to maximum value in view

`min` Sets the minimum value for the data (y-axis) scale

`max` Sets the maximum value for the data (y-axis) scale; ignored if autoscale is TRUE

`visibilityWindow`
Maximum window size in base pairs for which data is displayed

Value

An object of the named class

Slots

`autoscale` logical, scale the track to the maximum value in view

`min` numeric, y-axis minimum, used when autoscale is FALSE

`max` numeric, y-axis maximum, used when autoscale is FALSE

`coreObject` the scores: data.frame, GRanges or UCSCData

Examples

```
tbl <- data.frame(chrom = "chr1", start = c(100, 200), end = c(200, 300),
                  value = c(0.5, 2), stringsAsFactors = FALSE)
track <- DataFrameQuantitativeTrack("bedGraph demo", tbl, autoscale = TRUE)
gr <- GenomicRanges::GRanges(c("chr1:101-200", "chr1:201-300"),
                             score = c(0.5, 2))
track2 <- GRangesQuantitativeTrack("GRanges demo", gr)
trackSize(track2)
```

`removeTracksByName` *remove tracks from the browser*

Description

delete tracks on the browser

Usage

```
removeTracksByName(session, id, trackNames)

## S4 method for signature 'ANY'
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

Details

An S4 generic dispatching on `session`; the default method (signature `"ANY"`) sends the message through a shiny session.

Value

nothing

Examples

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

`removeUserAddedTracks` *remove only those tracks explicitly added by your app*

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 *draw the igv genome browser element*

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

session	Shiny session object (must be an active Shiny session)
filePath	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/<fi

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
```

---

show,Track-method *Print a summary of a track object*

Description

One method for the whole Track hierarchy: class, name, type, format, source and, when the track holds data, its size.

Usage

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

Arguments

object	a Track

Value

nothing; called for the side effect of printing a summary

Examples

```
DataFrameAnnotationTrack("demo",
  data.frame(chr = "chr1", start = 10, end = 20))
```

---

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)

## S4 method for signature 'ANY'
showGenomicRegion(session, id, region)

getGenomicRegion(session, id)

## S4 method for signature 'ANY'
getGenomicRegion(session, id)
```

Arguments

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

Details

showGenomicRegion, getGenomicRegion and removeTracksByName are S4 generics dispatching on their first argument. The default method (signature "ANY") handles a shiny session (a ShinySession, a module session_proxy or any object with a sendCustomMessage method), so packages that build on igvShiny can add methods for their own classes without masking these functions.

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)
}
```

`Track-class` *Track: the base class of igvShiny's S4 tracks*

Description

`Track` holds the slots every igv.js track shares. It is the parent of `igvAnnotationTrack` and `QuantitativeTrack`; build tracks with their subclass constructors, e.g. `DataFrameAnnotationTrack`, and display them with `loadTrack`.

Usage

```
Track(
  trackType = c("annotation", "quantitative", "alignment", "variant", "gwas",
    "genomicAlignment", "remoteAlignment", "pairedEndAnnotation"),
  sourceType = c("file", "url", "gcs", "ga4gh"),
 fileFormat = c("bed", "gff", "gff3", "gtf", "wig", "bigWig", "bedGraph", "bam", "vcf",
    "seg", "gwas", "cram", "bedpe"),
  trackName,
  onScreenOrder = NA_real_,
  color = "gray",
  height = 50,
  autoTrackHeight = FALSE,
  minTrackHeight = 50,
  maxTrackHeight = 500,
  visibilityWindow = 1e+05
)
```

Arguments

`trackType`	One of "annotation", "quantitative", "alignment", "variant", "gwas", "genomicAlignment", "remoteAlignment", "pairedEndAnnotation"
`sourceType`	One of "file", "url", "gcs", "ga4gh"
`fileFormat`	One of "bed", "gff", "gff3", "gtf", "wig", "bigWig", "bedGraph", "bam", "vcf", "seg", "gwas", "cram", "bedpe"
`trackName`	A character string, used as track label by igv; we recommend unique names per track
`onScreenOrder`	Numeric, for explicit placement of track within the current set
`color`	A CSS color name (e.g., "red" or "#FF0000"), or "random"
`height`	track height in pixels
`autoTrackHeight`	If TRUE, track height is adjusted dynamically, within the bounds set by minTrackHeight and maxTrackHeight
`minTrackHeight`	In pixels, minimum allowed
`maxTrackHeight`	In pixels, maximum allowed
`visibilityWindow`	Maximum window size in base pairs for which indexed annotations or variants are displayed

Value

An object of class Track

Slots

`trackType` character, e.g. "annotation" or "quantitative"

`sourceType` character, e.g. "file" or "url"

`fileFormat` character, e.g. "bed" or "bedGraph"

`trackName` character, the track label

`onScreenOrder` numeric, explicit placement of the track

`color` character, a CSS color

`height` numeric, track height in pixels

`autoTrackHeight` logical; stored, not sent to igv.js yet

`minTrackHeight` numeric, pixels; stored, not sent to igv.js yet

`maxTrackHeight` numeric, pixels; stored, not sent to igv.js yet

`visibilityWindow` numeric, base pairs; sent to igv.js by the annotation and quantitative `loadTrack` methods (NA: not sent)

References

https://igv.org/doc/igvjs/#tracks/Tracks/

Examples

```
track <- Track(trackType = "annotation", sourceType = "file",
               fileFormat = "bed", trackName = "demo", color = "red")
trackInfo(track)
```

---

trackInfo *Basic facts about a track*

Description

Basic facts about a track

Usage

```
trackInfo(obj)

trackSize(obj)

## S4 method for signature 'Track'
trackInfo(obj)

## S4 method for signature 'Track'
trackSize(obj)
```

```
## S4 method for signature 'GenomicAlignmentTrack'
trackSize(obj)

## S4 method for signature 'VariantTrack'
trackSize(obj)

## S4 method for signature 'GFF3Track'
trackSize(obj)
```

Arguments

obj	An object of base class Track

Value

trackInfo: a list with trackType, fileFormat, source and class. trackSize: the number of features, or NA when the track holds no data of its own

Examples

```
tbl <- data.frame(chrom = "chr1", start = c(100, 300), end = c(200, 400),
                  stringsAsFactors = FALSE)
track <- DataFrameAnnotationTrack("demo", tbl)
trackInfo(track)
trackSize(track)
```

VariantTrack-class *VariantTrack: variants from a VariantAnnotation VCF object*

Description

A VariantTrack holds a VCF object (needs the suggested package VariantAnnotation), written to a vcf file at display time, or the urls of a served vcf and its index. Display it with loadTrack, which sends color (from anchorColor), the three genotype colors, displayMode, trackHeight (as height) and visibilityWindow to igv.js. Url tracks have no igvShiny browser handler yet, so loadTrack refuses them.

Usage

```
VariantTrack(
  trackName,
  vcf,
  trackHeight = 50,
  anchorColor = "pink",
  homvarColor = "rgb(17,248,254)",
  hetvarColor = "rgb(34,12,253)",
  homrefColor = "rgb(200,200,200)",
  displayMode = "EXPANDED",
  visibilityWindow = 1e+05
)
```

Arguments

trackName	A character string, used as track label by igv
vcf	a VCF object, or a list with the urls data and index
trackHeight	track height in pixels
anchorColor	CSS color of the variant row
homvarColor	CSS color of homozygous variant genotypes
hetvarColor	CSS color of heterozygous variant genotypes
homrefColor	CSS color of homozygous reference genotypes
displayMode	"COLLAPSED", "SQUISHED" or "EXPANDED"
visibilityWindow	Maximum window size in base pairs for which variants are displayed

Value

An object of class VariantTrack

Slots

displayMode character, "COLLAPSED", "SQUISHED" or "EXPANDED"

vcf.obj a VCF object, or NULL for a url track

vcf.url list with the urls data and index, or empty

anchorColor character, the variant color

homvarColor character, homozygous variant genotype color

hetvarColor character, heterozygous variant genotype color

homrefColor character, homozygous reference genotype color

Examples

```
if (requireNamespace("VariantAnnotation", quietly = TRUE)) {
  f <- system.file(package = "igvShiny", "extdata",
                   "chr19-cebpaRegion.vcf.gz")
  vcf <- VariantAnnotation::readVcf(f, "hg38")
  track <- VariantTrack("cebpa variants", vcf)
  trackSize(track)
}
```

Index

∗ **GWASTrack_class**
.sanitizeChromosomeColorMap, 22
display,GWASTrack-method, 33
getUrl,GWASTrack-method, 35
GWASTrack-class, 40
show,GWASTrack-method, 72
∗ **igvShiny**
.packJunctionAttributes, 20
.sanitizeAndMergeOptions, 21
.sanitizeTrack, 22
.sanitizeTracks, 23
igvShiny, 44
igvShinyOutput, 45
parseAndValidateGenomeSpec, 66
removeTracksByName, 69
removeUserAddedTracks, 70
renderIgvShiny, 71
showGenomicRegion, 73
∗ **internal**
.alignmentConfig, 4
.bamLocalDataTrackPayload, 4
.bamUrlTrackPayload, 5
.bed12Columns, 6
.bedGraphTrackPayload, 6
.bedGraphUrlTrackPayload, 7
.bedTrackPayload, 9
.bedpeTrackPayload, 8
.configListTrackPayload, 9
.cramUrlTrackPayload, 10
.extractBamContigs, 11
.extractBedContigs, 11
.extractCramContigs, 12
.extractVcfContigs, 12
.getGenomeSpec, 13
.getReferenceContigs, 13
.gff3TrackPayload, 14
.gff3UrlTrackPayload, 15
.granges2bedGraphTable, 16
.granges2bedTable, 16
.gwasTrackObjectPayload, 17
.gwasTrackPayload, 17
.identifyLikelyAssembly, 18
.normalizeBedColumns, 18
.normalizeBedGraphColumns, 19
.normalizeJunctionColumns, 19
.registerGenomeSpec, 20
.registerTrackFile, 21
.segTrackPayload, 23
.sendTrack, 24
.serveFileWithHttpRange, 24
.slotsOverConfig, 25
.spliceJunctionTrackPayload, 25
.spliceJunctionUrlTrackPayload, 26
.stageTrackFile, 27
.trackFile, 28
.trackPayload, 28
.trackReferenceData, 29
.trackTable, 30
.tracksDir, 29
.validateTrackReference, 30
.vcfTrackPayload, 31
∗ **track_classes**
BedpeInteractionsTrack-class, 31
GenomicAlignmentTrack-class, 34
GFF3Track-class, 38
igvAnnotationTrack, 41
igvAnnotationTrack-class, 42
QuantitativeTrack-class, 67
show,Track-method, 73
Track-class, 75
trackInfo, 76
VariantTrack-class, 77
∗ **track_loaders**
loadBamTrackFromLocalData, 46
loadBamTrackFromLocalFile, 47
loadBamTrackFromURL, 48
loadBedGraphTrack, 49
loadBedGraphTrackFromURL, 50
loadBedTrack, 52
loadCramTrackFromLocalData, 53
loadCramTrackFromURL, 54
loadGFF3TrackFromLocalData, 55
loadGFF3TrackFromURL, 56
loadGwasTrack, 57
loadSegTrack, 58
loadSpliceJunctionTrackFromLocalData,

59
loadSpliceJunctionTrackFromURL, 61
loadTrack, 62
loadVcfTrack, 65
∗ **utils**
checkReferenceCompatibility, 32
get_basic_genomes, 36
get_cas_genomes, 37
get_css_genomes, 37
get_tracks_dir, 38
serveLocalFile, 71
.BedpeInteractionsTrack *(BedpeInteractionsTrack-class)*, 31
.CramTrack *(GenomicAlignmentTrack-class)*, 34
.DataFrameAnnotationTrack *(igvAnnotationTrack-class)*, 42
.DataFrameQuantitativeTrack *(QuantitativeTrack-class)*, 67
.GFF3Track *(GFF3Track-class)*, 38
.GRangesAnnotationTrack *(igvAnnotationTrack-class)*, 42
.GRangesQuantitativeTrack *(QuantitativeTrack-class)*, 67
.GWASTrack *(GWASTrack-class)*, 40
.GenomicAlignmentTrack *(GenomicAlignmentTrack-class)*, 34
.QuantitativeTrack *(QuantitativeTrack-class)*, 67
.RemoteAlignmentTrack *(GenomicAlignmentTrack-class)*, 34
.Track *(Track-class)*, 75
.UCSCBedAnnotationTrack *(igvAnnotationTrack-class)*, 42
.UCSCBedGraphQuantitativeTrack *(QuantitativeTrack-class)*, 67
.VariantTrack *(VariantTrack-class)*, 77
.alignmentConfig, 4
.bamLocalDataTrackPayload, 4
.bamUrlTrackPayload, 5
.bed12Columns, 6
.bedGraphTrackPayload, 6
.bedGraphUrlTrackPayload, 7
.bedTrackPayload, 9
.bedpeTrackPayload, 8
.configListTrackPayload, 9
.cramUrlTrackPayload, 10
.extractBamContigs, 11
.extractBedContigs, 11
.extractCramContigs, 12
.extractVcfContigs, 12
.getGenomeSpec, 13
.getReferenceContigs, 13
.gff3TrackPayload, 14
.gff3UrlTrackPayload, 15
.granges2bedGraphTable, 16
.granges2bedTable, 16
.gwasTrackObjectPayload, 17
.gwasTrackPayload, 17
.identifyLikelyAssembly, 18
.igvAnnotationTrack *(igvAnnotationTrack-class)*, 42
.normalizeBedColumns, 18
.normalizeBedGraphColumns, 19
.normalizeJunctionColumns, 19
.packJunctionAttributes, 20
.registerGenomeSpec, 20
.registerTrackFile, 21
.sanitizeAndMergeOptions, 21
.sanitizeChromosomeColorMap, 22
.sanitizeTrack, 22
.sanitizeTracks, 23
.segTrackPayload, 23
.sendTrack, 24
.serveFileWithHttpRange, 24
.slotsOverConfig, 25
.spliceJunctionTrackPayload, 25
.spliceJunctionUrlTrackPayload, 26
.stageTrackFile, 27
.trackFile, 28
.trackPayload, 28
.trackReferenceData, 29
.trackTable, 30
.tracksDir, 29
.validateTrackReference, 30
.vcfTrackPayload, 31

BedpeInteractionsTrack *(BedpeInteractionsTrack-class)*, 31
BedpeInteractionsTrack-class, 31

checkReferenceCompatibility, 32
CramTrack *(GenomicAlignmentTrack-class)*, 34
CramTrack-class *(GenomicAlignmentTrack-class)*, 34

DataFrameAnnotationTrack, *64*, *75*

DataFrameAnnotationTrack (igvAnnotationTrack-class), 42
DataFrameAnnotationTrack-class (igvAnnotationTrack-class), 42
DataFrameQuantitativeTrack (QuantitativeTrack-class), 67
DataFrameQuantitativeTrack-class (QuantitativeTrack-class), 67
display (display,GWASTrack-method), 33
display,GWASTrack-method, 33

GenomicAlignmentTrack (GenomicAlignmentTrack-class), 34
GenomicAlignmentTrack-class, 34
get_basic_genomes, 36
get_cas_genomes, 37
get_css_genomes, 37
get_tracks_dir, 38
getGenomicRegion (showGenomicRegion), 73
getGenomicRegion,ANY-method (showGenomicRegion), 73
getUrl (getUrl,GWASTrack-method), 35
getUrl,GWASTrack-method, 35
GFF3Track (GFF3Track-class), 38
GFF3Track-class, 38
GRangesAnnotationTrack (igvAnnotationTrack-class), 42
GRangesAnnotationTrack-class (igvAnnotationTrack-class), 42
GRangesQuantitativeTrack, *64*
GRangesQuantitativeTrack (QuantitativeTrack-class), 67
GRangesQuantitativeTrack-class (QuantitativeTrack-class), 67
GWASTrack, *58*
GWASTrack (GWASTrack-class), 40
GWASTrack-class, 40

igvAnnotationTrack, 41, *75*
igvAnnotationTrack-class, 42
igvShiny, 44
igvShinyOutput, 45

loadBamTrackFromLocalData, 46
loadBamTrackFromLocalFile, 47
loadBamTrackFromURL, *46*, 48, *53, 54*
loadBedGraphTrack, 49, *62*
loadBedGraphTrackFromURL, 50
loadBedTrack, 52, *62*
loadCramTrackFromLocalData, 53
loadCramTrackFromLocalFile (loadCramTrackFromLocalData), 53
loadCramTrackFromURL, 54
loadGenomeAnnotationTrack (loadBedGraphTrack), 49
loadGFF3TrackFromLocalData, 55
loadGFF3TrackFromURL, 56
loadGwasTrack, 57
loadSEGTrack (loadSegTrack), 58
loadSegTrack, 58
loadSpliceJunctionTrackFromLocalData, 59
loadSpliceJunctionTrackFromURL, *60*, 61
loadTrack, *28*, *31*, *34*, *38*, *42*, 62, *67*, *75*, *77*
loadTrack,BedpeInteractionsTrack-method (loadTrack), 62
loadTrack,CramTrack-method (loadTrack), 62
loadTrack,GenomicAlignmentTrack-method (loadTrack), 62
loadTrack,GFF3Track-method (loadTrack), 62
loadTrack,GWASTrack-method (loadTrack), 62
loadTrack,igvAnnotationTrack-method (loadTrack), 62
loadTrack,list-method (loadTrack), 62
loadTrack,QuantitativeTrack-method (loadTrack), 62
loadTrack,RemoteAlignmentTrack-method (loadTrack), 62
loadTrack,Track-method (loadTrack), 62
loadTrack,VariantTrack-method (loadTrack), 62
loadVcfTrack, 65

parseAndValidateGenomeSpec, *32*, 66

QuantitativeTrack, *75*
QuantitativeTrack (QuantitativeTrack-class), 67
QuantitativeTrack-class, 67

RemoteAlignmentTrack (GenomicAlignmentTrack-class), 34
RemoteAlignmentTrack-class (GenomicAlignmentTrack-class), 34
removeTracksByName, 69
removeTracksByName,ANY-method (removeTracksByName), 69
removeUserAddedTracks, 70
renderIgvShiny, 71

serveLocalFile, 71

show,GWASTrack-method, 72
show,Track-method, 73
showGenomicRegion, 73
showGenomicRegion,ANY-method
(showGenomicRegion), 73

Track, *42*, *67*, *73*
Track (Track-class), 75
Track-class, 75
trackInfo, 76
trackInfo,Track-method (trackInfo), 76
trackSize (trackInfo), 76
trackSize,GenomicAlignmentTrack-method
(trackInfo), 76
trackSize,GFF3Track-method (trackInfo),
76
trackSize,Track-method (trackInfo), 76
trackSize,VariantTrack-method
(trackInfo), 76

UCSCBedAnnotationTrack
(igvAnnotationTrack-class), 42
UCSCBedAnnotationTrack-class
(igvAnnotationTrack-class), 42
UCSCBedGraphQuantitativeTrack
(QuantitativeTrack-class), 67
UCSCBedGraphQuantitativeTrack-class
(QuantitativeTrack-class), 67

VariantTrack (VariantTrack-class), 77
VariantTrack-class, 77