

# Package ‘igvShiny’

May 10, 2024

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

**Version** 1.0.0

**Date** 2024-04-23

**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

**Imports** BiocGenerics, checkmate, futile.logger, GenomeInfoDbData, htmlwidgets, httr, jsonlite, randomcoloR, RCurl, utils

**Suggests** BiocStyle, GenomicAlignments, knitr, Rsamtools, rtracklayer, RUnit, shinytest2, VariantAnnotation

**biocViews** Software, ShinyApps, Sequencing, Coverage

**License** MIT + file LICENSE

**Encoding** UTF-8

**Collate** 'constants.R' 'genomeSpec.R' 'GWASTrack.R' 'igvShiny.R'  
'packages.R' 'test\_igvShiny\_package.R' 'utils.R' 'zzz.R'

**RoxygenNote** 7.3.1

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/igvShiny>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 75da2db

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-10

**Author** Paul Shannon [aut],  
 Arkadiusz Gladki [aut, cre] (<<https://orcid.org/0000-0002-7059-6378>>),  
 Karolina Scigocka [aut]

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

## Contents

display,GWASTrack-method . . . . .	2
getUrl,GWASTrack-method . . . . .	3
get_basic_genomes . . . . .	4
get_cas_genomes . . . . .	5
get_css_genomes . . . . .	5
get_tracks_dir . . . . .	6
GWASTrack-class . . . . .	6
igvShiny . . . . .	8
igvShinyOutput . . . . .	9
loadBamTrackFromLocalData . . . . .	10
loadBamTrackFromURL . . . . .	11
loadBedGraphTrack . . . . .	12
loadBedGraphTrackFromURL . . . . .	13
loadBedTrack . . . . .	14
loadCramTrackFromURL . . . . .	15
loadGFF3TrackFromLocalData . . . . .	16
loadGFF3TrackFromURL . . . . .	17
loadGwasTrack . . . . .	18
loadSegTrack . . . . .	19
loadVcfTrack . . . . .	20
parseAndValidateGenomeSpec . . . . .	21
removeTracksByName . . . . .	22
removeUserAddedTracks . . . . .	23
renderIgvShiny . . . . .	24
showGenomicRegion . . . . .	24
<b>Index</b>	<b>26</b>

---

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

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

### Value

nothing

### Examples

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

---

`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

<code>obj</code>	An object of class <code>GWASTrack</code>
------------------	-------------------------------------------

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

---

**Description**

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

**Usage**

```
get_cas_genomes()
```

**Value**

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

**Examples**

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

---

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

---

**Description**

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

**Usage**

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

**Arguments**

test                   logical(1) defaults to FALSE

**Value**

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

**Examples**

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

---

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

---

### Description

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

### Usage

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

### Arguments

<code>env_var</code>	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")
```

---

<code>GWASTrack-class</code>	<i>Constructor for GWASTrack</i>
------------------------------	----------------------------------

---

### Description

GWASTrack creates an IGV manhattan track from GWAS data

**Usage**

```
GWASTrack(
  trackName,
  data,
  chrom.col,
  pos.col,
  pval.col,
  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
trackHeight	numeric in pixels
autoscale	logical
minY	numeric for explicit (non-auto) scaling
maxY	numeric for explicit (non-auto) scaling

**Value**

A GWAS*Track* object

**Examples**

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

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

```

    pval.col = 28
  )
  getUrl(track)

url <- "https://s3.amazonaws.com/igv.org/demo/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)

```

---

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



**Value**

the created widget

**Examples**

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

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

the created widget's html

**Examples**

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

---

```
loadBamTrackFromLocalData
  load GenomicAlignments data as an igv.js alignment track
```

---

**Description**

load GenomicAlignments data as an igv.js alignment track

**Usage**

```
loadBamTrackFromLocalData(
  session,
  id,
  trackName,
  data,
  deleteTracksOfSameName = TRUE,
  displayMode = "EXPANDED"
)
```

**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"

**Value**

nothing

**Examples**

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

---

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  
)
```

### 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

### Value

nothing

### Examples

```
library(igvShiny)  
demo_app_file <-  
  system.file(package = "igvShiny", "demos", "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
)
```

### 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

### Value

nothing

## Examples

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "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
)
```

## 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

**Value**

nothing

**Examples**

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

---

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

---

**Description**

load a bed track provided as a data.frame

**Usage**

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

**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

**Value**

nothing

**Examples**

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "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
)
```

**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

**Value**

nothing

**Examples**

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "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
)
```

**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

**Value**

nothing

**Examples**

```

library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "igvShinyDemo-GFF3.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
)

```

**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

**Value**

nothing

**Examples**

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "igvShinyDemo-GFF3.R")
if (interactive()) {
  shiny::runApp(demo_app_file)
}
```

---

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

---

**Description**

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

**Usage**

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

**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

**Value**

nothing

**Examples**

```
library(igvShiny)  
demo_app_file <-  
  system.file(package = "igvShiny", "demos", "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)
```

**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
```

**Value**

nothing

**Examples**

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

---

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

---

**Description**

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

**Usage**

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

**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
```

**Value**

nothing

**Examples**

```
library(igvShiny)
demo_app_file <-
  system.file(package = "igvShiny", "demos", "igvShinyDemo-withVCF.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
base.url <-
  "https://gladki.pl/igvr/testFiles/sarsGenome"
fasta.file <-
  sprintf("%s/%s", base.url, "Sars_cov_2.ASM985889v3.dna.toplevel.fa")
fastaIndex.file <-
  sprintf("%s/%s",
    base.url,
    "Sars_cov_2.ASM985889v3.dna.toplevel.fa.fai")
annotation.file <-
  sprintf("%s/%s", base.url, "Sars_cov_2.ASM985889v3.101.gff3")
custom.genome.title <- "SARS-CoV-2"
genomeOptions <-
  parseAndValidateGenomeSpec(
    genomeName = custom.genome.title,
    initialLocus = "all",
    stockGenome = FALSE,
    dataMode = "http",
    fasta = fasta.file,
    fastaIndex = fastaIndex.file,
    genomeAnnotation = annotation.file
  )
```

---

removeTracksByName     *remove tracks from the browser*

---

**Description**

delete tracks on the browser

**Usage**

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

**Arguments**

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

**Value**

nothing

**Examples**

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

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

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

**Examples**

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

---

showGenomicRegion	<i>focus igv on a region</i>
-------------------	------------------------------

---

**Description**

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

**Usage**

```
showGenomicRegion(session, id, region)

getGenomicRegion(session, id)
```



**Arguments**

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

**Value**

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

**Examples**

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

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

# Index

- \* **GWASTrack\_class**
  - display, GWASTrack-method, 2
  - getUrl, GWASTrack-method, 3
  - GWASTrack-class, 6
- \* **igvShiny**
  - igvShiny, 8
  - igvShinyOutput, 9
  - parseAndValidateGenomeSpec, 21
  - removeTracksByName, 22
  - removeUserAddedTracks, 23
  - renderIgvShiny, 24
  - showGenomicRegion, 24
- \* **track\_loaders**
  - loadBamTrackFromLocalData, 10
  - loadBamTrackFromURL, 11
  - loadBedGraphTrack, 12
  - loadBedGraphTrackFromURL, 13
  - loadBedTrack, 14
  - loadCramTrackFromURL, 15
  - loadGFF3TrackFromLocalData, 16
  - loadGFF3TrackFromURL, 17
  - loadGwasTrack, 18
  - loadSegTrack, 19
  - loadVcfTrack, 20
- \* **utils**
  - get\_basic\_genomes, 4
  - get\_cas\_genomes, 5
  - get\_css\_genomes, 5
  - get\_tracks\_dir, 6
- .GWASTrack (GWASTrack-class), 6
- display (display, GWASTrack-method), 2
- display, GWASTrack-method, 2
- get\_basic\_genomes, 4
- get\_cas\_genomes, 5
- get\_css\_genomes, 5
- get\_tracks\_dir, 6
- getGenomicRegion (showGenomicRegion), 24
- getUrl (getUrl, GWASTrack-method), 3
- getUrl, GWASTrack-method, 3
- GWASTrack (GWASTrack-class), 6
- GWASTrack-class, 6
- igvShiny, 8
- igvShinyOutput, 9
- loadBamTrackFromLocalData, 10
- loadBamTrackFromURL, 11
- loadBedGraphTrack, 12
- loadBedGraphTrackFromURL, 13
- loadBedTrack, 14
- loadCramTrackFromURL, 15
- loadGenomeAnnotationTrack
  - (loadBedGraphTrack), 12
- loadGFF3TrackFromLocalData, 16
- loadGFF3TrackFromURL, 17
- loadGwasTrack, 18
- loadSEGTrack (loadSegTrack), 19
- loadSegTrack, 19
- loadVcfTrack, 20
- parseAndValidateGenomeSpec, 21
- removeTracksByName, 22
- removeUserAddedTracks, 23
- renderIgvShiny, 24
- showGenomicRegion, 24