IGV-webapp

kike

12/6/2019

IGV-WebApp

I have found IGV-webapp suited for situations in which we are interested in creating a genome session browser and then share that session online via a URL with the community IGV-WebApp.

Create session from public files in Google Cloud Storage (GCS)

We are creating a JSON file that helps us building the session when we've a lot of files. We can then upload directly this json file to IGVWebApp. For the moment it's only working for bigwig files.

We just need:
* Default session json file (mine is for hg19, but feel free to change and download it)
* Name of the GCS bucket
* Directory path of the folder with all files in the GCS bucket
* csv annotation file where each row contains the name of the file (all bigwigs), and it's ID
* Final json file name

```r
annotation_file_name <- "./hg19_browser_links.csv"
small_jason_file_name <- "./igv-app-test_hg19.json"
bucket_name <- "any_bucket_name"
dir_path <- "any_dir_path"
final_json <- "./final.json"

library(RJSONIO)

annotation <- read.csv(annotation_file_name)
start_json <- start_json <-RJSONIO::fromJSON(small_jason_file_name)
tracks <- start_json$tracks
```

Annotation

```r
head(annotation)
```

```
##   Library         ID   type          url
## 1      l1   D0_skhds bigWig file1.bigwig
## 2      l2 D0_fdfdfdf bigWig File2.bigwig
## 3      l3    D3_gfgf bigWig File3.bigwig
```

```r
for (i in 1:nrow(annotation))
{
  new_track <- annotation[i,]
  name <- as.character(new_track$ID)
  filename <- strsplit(as.character(new_track$url),dir_path)[[1]][2]
  format <- "bigwig"
  url <- paste("gs://",bucket_name,dir_path,filename,sep="")
  sourceType <- "file"
  type <- "wig"
  noSpinner <- TRUE
  color <- "rgb(150,150,150)"
  height <- 50
  autoscale <- TRUE
  order <- i

  new_track_info <- list(name=name,filename=filename,format=format,url=url,sourceType=sourceType,
                         type=type,noSpinner=noSpinner,color=color,height=height,autoscale=autoscale,or
  tracks[[i+2]] <- new_track_info

}

#save to json format
start_json$tracks <- tracks
exportJSON <-RJSONIO::toJSON(start_json)
write(exportJSON, final_json)  #this is the file I use to load json file to IGV
```