cghub: An interface for chatting with Cancer Genomics Hub

April 7, 2015

Contents

1 Installing the package **1**

2 Examples: **1**

2.1 Querying 1

2.2 Fetching the data 2

cghub.ucsc.edu

this package is under active developement Feel free to fork, log issues and submit pull requests https://github.com/sahilseth/cghub

1 Installing the package

```
install.packages("devtools")
devtools::install_github("sahilseth/cghub")
```

2 Examples:

2.1 Querying

For more on querying the database, please look at: https://cghub.ucsc.edu/docs/user/query.html

2.1.1 Get hits for a tumor type

Query is based on: - Get all *WGS* bams which are *live* for tumor *CESC*, and provide the number available

```
library(cghub)
out <- gt.cgquery(search = "tumor=CESC&library_strategy=WGS&study=phs000178&state=live",
                  count.only = TRUE)
```

2.1.2 Query and parse hits into a table

```
s = c(
  study = 'phs000178',
  library_strategy = 'WGS',
  disease_abbr = 'CESC',
  state = 'live')
search = paste(names(s), s, collapse = "&", sep = "=")
#search
out <- gt.cgquery(search = search, count.only = FALSE)
names(out)
```

[1] “tab” “query” “hits”

```
out$query
```

[1] “study:phs000178 library_strategy:WGS disease_abbr:CESC state:live”

```
out$hits
```

[1] 142

study	disease	platform	centerName	barcode	aliquotID
phs000178	CESC	ILLUMINA	HMS-RK	TCGA-HG-A2PA-01A-11D-A20X-26	ECE06317-01F9-4EF2-ACE0-E9
phs000178	CESC	ILLUMINA	HMS-RK	TCGA-FU-A3TQ-01A-11D-A232-26	7018744f-49ff-4f47-aee5-07f0dea5
phs000178	CESC	ILLUMINA	HMS-RK	TCGA-JX-A3Q0-10A-01D-A245-26	3494d848-1830-49a0-b144-22550b
phs000178	CESC	ILLUMINA	HMS-RK	TCGA-JX-A3Q8-10A-01D-A245-26	dfede0ba-0e85-4d22-9a70-88e798
phs000178	CESC	ILLUMINA	HMS-RK	TCGA-C5-A2LZ-01A-11D-A20X-26	837B293C-0843-4002-A8AB-3A9
phs000178	CESC	ILLUMINA	HMS-RK	TCGA-EA-A3HU-10B-01D-A20X-26	4B33B319-40C6-4C3C-B259-5518

2.1.3 Based on a list of TCGA barcodes

```
require(parallel)
ids <- c("TCGA-BJ-A0Z5-01","TCGA-DJ-A1QG-01","TCGA-EM-A1YC-01",
         "TCGA-ET-A25J-01", "TCGA-H2-A26U-01","TCGA-H2-A3RI-01")

tab <- mclapply(ids,function(id){
  out <- gt.cgquery(search=sprintf("legacy_sample_id=%s*&state=live",id), count.only=FALSE)
  return(out$tab)
  },mc.cores=1)
tab=do.call(rbind,tab)
knitr::kable(head(out$tab), format = "markdown")
```

2.2 Fetching the data

2.2.1 Load BAM using GTFuse

```
## This contains all the uuids: out$tab$uuid
path=gt.loadBamFile(uuid='056a5a41-1be5-48f1-8ae5-bd8125e5b0c2', key="mykey.pem", force=TRUE)
```

2.2.2 Download BAM using gtdownload

- this supports providing multiple UUIDs

```
outBam <- gt.cgDownload(uuids=uuids, path="/scratch/gt",
                        GT="gtdownload", cores=1,
                        key="mykey.pem",params="")
```