Package 'GoogleGenomics'

October 16, 2018

Version 2.2.0

Title R Client for Google Genomics API

Author Cassie Doll, Nicole Deflaux

Maintainer Siddhartha Bagaria <bagaria.siddhartha@gmail.com>

Depends R (>= 3.1.0), GenomicAlignments (>= 1.0.1), VariantAnnotation

Imports Biostrings, GenomeInfoDb, GenomicRanges, IRanges, httr, rjson, Rsamtools, S4Vectors (>= 0.9.25), Biobase, methods, utils

Suggests BiocStyle, RProtoBuf, httpuv, knitr, rmarkdown, testthat, ggbio, ggplot2, BSgenome.Hsapiens.UCSC.hg19, org.Hs.eg.db, TxDb.Hsapiens.UCSC.hg19.knownGene

Description Provides an R package to interact with the Google Genomics API.

SystemRequirements GNU make

VignetteBuilder knitr

License Apache License (== 2.0) | file LICENSE

URL https://cloud.google.com/genomics/

BugReports https://github.com/Bioconductor/GoogleGenomics/issues

biocViews DataImport, ThirdPartyClient, Genetics

RoxygenNote 6.0.1

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

git_branch RELEASE_3_7

git_last_commit 65aee17

git_last_commit_date 2018-04-30

Date/Publication 2018-10-15

R topics documented:

`authenticate` *Configure how to authenticate for Google Genomics API.*

Description

Follow the sign up instructions at https://cloud.google.com/genomics/install-genomics-tools.

Usage

```
authenticate(file, clientId, clientSecret, invokeBrowser,
  apiKey = Sys.getenv("GOOGLE_API_KEY"),
  gcloudCredsPath = defaultGcloudCredsPath(), tryGCEServiceAccount = TRUE)
```

Arguments

`file`	Client secrets file obtained from Google Developer Console. This file could be for a native application or a service account. If this file is not present, clientId and clientSecret must be provided for native application credentials.
`clientId`	Client ID from Google Developer Console, overridden if file is provided.
`clientSecret`	Client Secret from Google Developer Console, overridden if file is provided.
`invokeBrowser`	If TRUE or not provided, the default browser is invoked with the auth URL iff the `httpuv` package is installed (suggested). If FALSE, a URL is output which needs to be copy pasted in a browser, and the resulting token needs to be pasted back into the R session. With both the options, you will still need to login to your Google account if not logged in already.
`apiKey`	Public API key that can be used to call the Genomics API for public datasets. This method of authentication does not need you to login to your Google account. Providing this key overrides all other arguments.
`gcloudCredsPath`	Path to the generated json file with application default credentials.
`tryGCEServiceAccount`	If TRUE, will try checking if this is a GCE VM instance with a valid service account. If valid credentials are found, will use them over all other options.

Details

There are four primary ways, in order of preference, of authenticating:

1. When running on Google Compute Engine, configure your VM to be authenticated at the time of initial setup. See https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-ir using.

2. Use the gcloud tool to generate application default credentials. If the generated file is not in its standard location, you can set the environment variable GOOGLE_APPLICATION_CREDENTIALS with its path, or provide the gcloudCredsPath argument.

3. For public data, use a public API key from the project that you want to access. You can either set the GOOGLE_API_KEY environment variable, or provide the apiKey argument. Does not work with gRPC.

4. Download secrets file (native application or service account) or provide the clientId and clientSecret pair. See https://cloud.google.com/genomics/downloading-credentials-for-api-access. Native application credentials should only be used when accessing data for which your own account is not authorized.

This method is called with default arguments at package load time.

Value

TRUE if successful, FALSE if not.

Examples

```
apiKey <- Sys.getenv("GOOGLE_API_KEY")
if (!is.na(apiKey) && nchar(apiKey)>0) {
  authenticate(apiKey=apiKey)
}
## Not run:
authenticate()
authenticate(file="clientSecrets.json")
authenticate(file="clientSecrets.json", invokeBrowser=FALSE)
authenticate(clientId="abc", clientSecret="xyz", invokeBrowser=FALSE)

## End(Not run)
```

`callGRPCMethod` *Issues a gRPC call to the Google Genomics service and returns the response.*

Description

Needs gRPC support at package build time and the RProtoBuf package. See package README for instructions on installing gRPC.

Usage

```
callGRPCMethod(methodName, request)
```

Arguments

methodName	The RPC method name.
request	The request object for the RPC, either as a JSON object generated from rjson, or as a RProtoBuf object modified from the default instance obtained from getRProtoBufDefaultObject.

Details

In general, use higher level methods such as getReads and getVariants instead.

Value

The raw response converted from JSON to an R object, or the RProtoBuf object if the request was an RProtoBuf object.

See Also

Other page fetch functions: getReadsPage, getSearchPage, getVariantsPage

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
if (isGRPCAvailable()) {
  request <- list(readGroupSetIds=list("CMvnhpKTFhDnk4_9zcKO3_YB"),
                  referenceName="22",
                  start=16051400, end=16051500, pageToken=NULL)
  reads <- callGRPCMethod("SearchReads", request)
  summary(reads)
} else {
  message("gRPC support is disabled; package was not compiled with GRPC")
}
```

---

defaultGcloudCredsPath

Returns the standard location for application default credentials as generated by the gcloud CLI tool.

Description

Returns the standard location for application default credentials as generated by the gcloud CLI tool.

Usage

```
defaultGcloudCredsPath()
```

Value

File path for credentials json.

Examples

```
defaultGcloudCredsPath()
```

getReads *Get reads from Google Genomics.*

Description

This function will return all of the reads that comprise the requested genomic range, iterating over paginated results as necessary.

Usage

```
getReads(readGroupSetId = "CMvnhpKTFhDnk4_9zcKO3_YB", chromosome = "22",
  start = 16051400, end = 16051500, fields = NULL, converter = c,
  useGRPC = getOption("google_genomics_use_grpc"))
```

Arguments

readGroupSetId	The read group set ID.
chromosome	The chromosome.
start	Start position on the chromosome in 0-based coordinates.
end	End position on the chromosome in 0-based coordinates.
fields	A subset of fields to retrieve. The default (NULL) will return all fields.
converter	A function that takes a list of read R objects and returns them converted to the desired type.
useGRPC	Whether to use GRPC mechanism to query.

Details

By default, this function gets reads for a small genomic region for one sample in 1,000 Genomes.

Optionally pass a converter as appropriate for your use case. By passing it to this method, only the converted objects will be accumulated in memory. The converter function should return an empty container of the desired type if called without any arguments.

Value

By default, the return value is a list of R objects corresponding to the JSON objects returned by the Google Genomics Reads API. If a converter is passed, object(s) of the type returned by the converter will be returned by this function.

See Also

getVariants

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
reads <- getReads()
summary(reads)
summary(reads[[1]])
```

getReadsPage *Get one page of reads from Google Genomics.*

Description

In general, use the getReads method instead. It calls this method, returning reads from all of the pages that comprise the requested genomic range.

Usage

```
getReadsPage(readGroupSetId = "CMvnhpKTFhDnk4_9zcKO3_YB", chromosome = "22",
  start = 16051400, end = 16051500, fields = NULL, pageToken = NULL)
```

Arguments

readGroupSetId	The read group set ID.
chromosome	The chromosome.
start	Start position on the chromosome in 0-based coordinates.
end	End position on the chromosome in 0-based coordinates.
fields	A subset of fields to retrieve. The default (NULL) will return all fields.
pageToken	The page token. This can be NULL (default) for the first page.

Details

By default, this function gets reads for a small genomic region for one sample in 1,000 Genomes.

Note that the Global Alliance for Genomics and Health API uses a 0-based coordinate system. For more detail, please see GA4GH discussions such as the following:

- https://github.com/ga4gh/schemas/issues/168
- https://github.com/ga4gh/schemas/issues/121

Value

A two-element list is returned by the function.

reads: A list of R objects corresponding to the JSON objects returned by the Google Genomics Reads API.

nextPageToken: The token to be used to retrieve the next page of results, if applicable.

See Also

Other page fetch functions: callGRPCMethod, getSearchPage, getVariantsPage

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
readsPage <- getReadsPage()
summary(readsPage)
summary(readsPage$reads[[1]])
```

`getRProtoBufDefaultObject` *Returns a protocol buffer Message object from* `RProtoBuf`.

Description

Needs gRPC support at package build time and the RProtoBuf package. See package README for instructions on installing gRPC.

Usage

```
getRProtoBufDefaultObject(fullyQualifiedName)
```

Arguments

`fullyQualifiedName`	Type of the message object to return.

Value

Default instance of the Message.

Examples

```
if (isGRPCAvailable()) {
  getRProtoBufDefaultObject("google.genomics.v1.SearchReadsRequest")
}
```

`getSearchPage` *Get one page of search results for a particular entity type from Google Genomics.*

Description

In general, use higher level methods such as getReads and getVariants instead.

Usage

```
getSearchPage(entityType, body, fields, pageToken)
```

Arguments

`entityType`	Entities with a search API such as reads, variants, variantSets, etc...
`body`	The body of the message to POST to the search endpoint.
`fields`	The fields to be returned in the search response.
`pageToken`	The page token. This can be NULL for the first page.

Value

The raw response converted from JSON to an R object.

See Also

Other page fetch functions: callGRPCMethod, getReadsPage, getVariantsPage

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
body <- list(readGroupSetIds=list("CMvnhpKTFhDnk4_9zcKO3_YB"),
             referenceName="22",
             start=16051400, end=16051500, pageToken=NULL)
reads <- getSearchPage("reads", body, NULL, NULL)
summary(reads)
```

---

getVariantCalls *Elaborate the result of getVariants as a VRanges with all calls for all samples*

Description

Elaborate the result of getVariants as a VRanges with all calls for all samples

Usage

```
getVariantCalls(variantSetId = "10473108253681171589", chromosome = "22",
  start = 16051400, end = 16051500, fields = NULL, converter = c,
  oneBasedCoord = TRUE, nullAction = "stop")
```

Arguments

variantSetId	The dataset ID.
chromosome	The chromosome.
start	Start position on the chromosome in 0-based coordinates.
end	End position on the chromosome in 0-based coordinates.
fields	A subset of fields to retrieve. The default (NULL) will return all fields.
converter	A function that takes a list of variant R objects and returns them converted to the desired type.
oneBasedCoord	Convert returned addresses to 1-based address system
nullAction	either "stop" or "warn" telling how to deal with event in which request yields no variants; for "warn" we return NULL

Value

By default, the return value is a VRanges object. If a converter is passed, object(s) of the type returned by the converter will be returned by this function.

Examples

```
## Not run:
getVariantCalls()

## End(Not run)
```

`getVariants` *Get variants from Google Genomics.*

Description

This function will return all of the variants that comprise the requested genomic range, iterating over paginated results as necessary.

Usage

```
getVariants(variantSetId = "10473108253681171589", chromosome = "22",
  start = 16051400, end = 16051500, fields = NULL, converter = c,
  useGRPC = getOption("google_genomics_use_grpc"))
```

Arguments

`variantSetId`	The dataset ID.
`chromosome`	The chromosome.
`start`	Start position on the chromosome in 0-based coordinates.
`end`	End position on the chromosome in 0-based coordinates.
`fields`	A subset of fields to retrieve. The default (NULL) will return all fields.
`converter`	A function that takes a list of variant R objects and returns them converted to the desired type.
`useGRPC`	Whether to use GRPC mechanism to query.

Details

By default, this function gets variants from a small section of 1000 Genomes phase 1 variants.

Optionally pass a converter as appropriate for your use case. By passing it to this method, only the converted objects will be accumulated in memory. The converter function should return an empty container of the desired type if called without any arguments.

Value

By default, the return value is a list of R objects corresponding to the JSON objects returned by the Google Genomics Variants API. If a converter is passed, object(s) of the type returned by the converter will be returned by this function.

See Also

`getReads` for equivalent function for reads, and `variantsToVRanges` for a converter function.

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
variants <- getVariants()
summary(variants)
summary(variants[[1]])
```

`getVariantsPage` *Get one page of variants from Google Genomics.*

Description

In general, use the getVariants method instead. It calls this method, returning variants from all of the pages that comprise the requested genomic range.

Usage

```
getVariantsPage(variantSetId = "10473108253681171589", chromosome = "22",
  start = 16051400, end = 16051500, fields = NULL, pageToken = NULL)
```

Arguments

`variantSetId`	The dataset ID.
`chromosome`	The chromosome.
`start`	Start position on the chromosome in 0-based coordinates.
`end`	End position on the chromosome in 0-based coordinates.
`fields`	A subset of fields to retrieve. The default (NULL) will return all fields.
`pageToken`	The page token. This can be NULL (default) for the first page.

Details

By default, this function gets variants from a small section of 1000 Genomes phase 1 variants.

Value

A two-element list is returned by the function.

variants: A list of R objects corresponding to the JSON objects returned by the Google Genomics Variants API.

nextPageToken: The token to be used to retrieve the next page of results, if applicable.

See Also

Other page fetch functions: `callGRPCMethod`, `getReadsPage`, `getSearchPage`

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
variantsPage <- getVariantsPage()
summary(variantsPage)
summary(variantsPage$variants[[1]])
```

`GoogleGenomics` *A basic R package for Google Genomics API.*

Description

For more details, read the package README.

isGRPCAvailable *Returns if this package was built with gRPC support.*

Description

Returns if this package was built with gRPC support.

Usage

```
isGRPCAvailable()
```

Value

TRUE iff the package was built with gRPC support.

readsToGAlignments *Convert reads to GAlignments.*

Description

Note that the Global Alliance for Genomics and Health API uses a 0-based coordinate system. For more detail, please see GA4GH discussions such as the following:

- https://github.com/ga4gh/schemas/issues/168
- https://github.com/ga4gh/schemas/issues/121

Usage

```
readsToGAlignments(reads, oneBasedCoord = TRUE, slStyle = "UCSC")
```

Arguments

reads	A list of R objects corresponding to the JSON objects returned by the Google Genomics Reads API.
oneBasedCoord	Convert genomic positions to 1-based coordinates.
slStyle	The style for seqnames (chrN or N or...). Default is UCSC.

Value

GAlignments

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
alignments1 <- getReads(converter=readsToGAlignments)
summary(alignments1)
alignments2 <- readsToGAlignments(getReads())
print(identical(alignments1, alignments2))
```

variantsToGRanges *Convert variants to GRanges.*

Description

Note that the Global Alliance for Genomics and Health API uses a 0-based coordinate system. For more detail, please see GA4GH discussions such as the following:

- https://github.com/ga4gh/schemas/issues/168
- https://github.com/ga4gh/schemas/issues/121

Usage

```
variantsToGRanges(variants, oneBasedCoord = TRUE, slStyle = "UCSC")
```

Arguments

variants	A list of R objects corresponding to the JSON objects returned by the Google Genomics Variants API.
oneBasedCoord	Convert genomic positions to 1-based coordinates.
slStyle	The style for seqnames (chrN or N or...). Default is UCSC.

Value

GRanges

See Also

Other variants converter functions: variantsToVRanges

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
variants1 <- getVariants(converter=variantsToGRanges)
summary(variants1)
variants2 <- variantsToGRanges(getVariants())
print(identical(variants1, variants2))
```

variantsToVRanges *Convert variants to VRanges.*

Description

Note that the Global Alliance for Genomics and Health API uses a 0-based coordinate system. For more detail, please see GA4GH discussions such as the following:

- https://github.com/ga4gh/schemas/issues/168
- https://github.com/ga4gh/schemas/issues/121

Usage

```
variantsToVRanges(variants, oneBasedCoord = TRUE, slStyle = "UCSC")
```

Arguments

variants	A list of R objects corresponding to the JSON objects returned by the Google Genomics Variants API.
oneBasedCoord	Convert genomic positions to 1-based coordinates.
slStyle	The style for seqnames (chrN or N or...). Default is UCSC.

Value

VRanges

See Also

Other variants converter functions: variantsToGRanges

Examples

```
# Authenticated on package load from the env variable GOOGLE_API_KEY.
variants1 <- getVariants(converter=variantsToVRanges)
summary(variants1)
variants2 <- variantsToVRanges(getVariants())
print(identical(variants1, variants2))
```

Index