Package ‘GA4GHclient’

April 15, 2020

Type Package

Title A Bioconductor package for accessing GA4GH API data servers

Version 1.10.0

Description GA4GHclient provides an easy way to access public data servers through Global Alliance for Genomics and Health (GA4GH) genomics API. It provides low-level access to GA4GH API and translates response data into Bioconductor-based class objects.

License GPL (>= 2)

Depends S4Vectors

Imports BiocGenerics, Biostrings, dplyr, GenomeInfoDb, GenomicRanges, httr, IRanges, jsonlite, methods, VariantAnnotation

Suggests AnnotationDbi, BiocStyle, DT, knitr, org.Hs.eg.db, rmarkdown, testthat, TxDb.Hsapiens.UCSC.hg19.knownGene

LazyData TRUE

RoxygenNote 6.0.1

VignetteBuilder knitr

URL `https://github.com/labbcb/GA4GHclient`

BugReports `https://github.com/labbcb/GA4GHclient/issues`

biocViews DataRepresentation, ThirdPartyClient

git_url https://git.bioconductor.org/packages/GA4GHclient

git_branch RELEASE_3_10

git_last_commit e328ce6

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

Author Welliton Souza [aut, cre],
Benilton Carvalho [ctb],
Cristiane Rocha [ctb]

Maintainer Welliton Souza <`well309@gmail.com`>

R topics documented:

`GA4GHclient-package` *A Bioconductor package for accessing GA4GH API data server*

Description

GA4GHclient provides an easy way to access public data servers through Global Alliance for Genomics and Health (GA4GH) genomics API. It provides low-level access to GA4GH API and translates response data into Bioconductor-based class objects.

Author(s)

Welliton Souza, Benilton Carvalho, Cristiane Rocha

Maintainer: Welliton Souza <well309@gmail.com>

`getBiosample` *getBiosample function*

Description

Get a biosample by its ID.

Usage

```
getBiosample(host, biosampleId)
```

Arguments

`host`	URL of GA4GH API data server.
`biosampleId`	ID of the biosample requested.

Details

This function requests `GET host/datasets/biosampleId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchBiosamples`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
biosampleId <- searchBiosamples(host, datasetId, nrows = 1)$id
getBiosample(host, biosampleId)

## End(Not run)
```

`getCallSet` *getCallSet function*

Description

Get a call set by its ID.

Usage

```
getCallSet(host, callSetId)
```

Arguments

`host`	URL of GA4GH API data server.
`callSetId`	The ID of the CallSet to be retrieved.

Details

This request maps to GET `host/callsets/callSetId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchCallSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id
callSetId <- searchCallSets(host, variantSetId, nrows = 1)$id
getCallSet(host, callSetId)

## End(Not run)
```

`getDataset` *getDataset function*

Description

Get a dataset by its ID.

Usage

```
getDataset(host, datasetId)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	The ID of the dataset to be retrieved.

Details

This function requests `GET` `host/datasets/datasetId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchDatasets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
getDataset(host, datasetId)

## End(Not run)
```

`getExpressionLevel` *getExpressionLevel function*

Description

Get an expression level by its ID.

Usage

```
getExpressionLevel(host, expressionLevelId)
```

Arguments

`host`	URL of GA4GH API data server.
`expressionLevelId`	ID of the expression level.

Details

This function requests `GET host/expressionlevels/expressionLevelId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchExpressionLevels`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
rnaQuantificationSetId <- searchRnaQuantificationSets(host, datasetId, nrow = 1)$id
rnaQuantificationId <- searchRnaQuantifications(host, rnaQuantificationSetId, nrows = 1)$id
expressionLevelId <- searchExpressionLevels(host, rnaQuantificationId, nrows = 1)$id
getExpressionLevel(host, expressionLevelId)

## End(Not run)
```

`getFeature` *getFeature function*

Description

Get a feature set by its ID (a line of genomic feature file).

Usage

```
getFeature(host, featureId)
```

Arguments

`host`	URL of GA4GH API data server.
`featureId`	The ID of the feature to be retrieved.

Details

This function requests `GET host/features/featureId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchFeatures`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
featureSetId <- searchFeatureSets(host, datasetId, nrows = 1)$id
featureId <- searchFeatures(host, featureSetId, nrows = 1)$id
getFeature(host, featureId)

## End(Not run)
```

`getFeatureSet` *getFeatureSet function*

Description

Get a feature set by its ID.

Usage

```
getFeatureSet(host, featureSetId)
```

Arguments

`host`	URL of GA4GH API data server.
`featureSetId`	The ID of the FeatureSet to be retrieved.

Details

This function requests `GET host/featuresets/featureSetId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchFeatureSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
featureSetId <- searchFeatureSets(host, datasetId, nrows = 1)$id
getFeatureSet(host, featureSetId)

## End(Not run)
```

`getIndividual` *getIndividual function*

Description

Get an individual by its ID.

Usage

```
getIndividual(host, individualId)
```

Arguments

`host`	URL of GA4GH API data server.
`individualId`	ID of the individual requested.

Details

This function requests `GET host/individuals/individualId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchIndividuals`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
individualId <- searchIndividuals(host, datasetId, nrows = 1)$id
getIndividual(host, individualId)

## End(Not run)
```

`getReadGroupSet` *getReadGroupSet function*

Description

Get a read group set by its ID.

Usage

```
getReadGroupSet(host, readGroupSetId)
```

Arguments

`host`	URL of GA4GH API data server.
`readGroupSetId`	The ID of the ReadGroupSet to be retrieved.

Details

This function requests `GET host/readgroupsets/readGroupSetId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchReadGroupSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
readGroupSetId <- searchReadGroupSets(host, datasetId, nrows = 1)$id
getReadGroupSet(host, readGroupSetId)

## End(Not run)
```

`getReference` *getReference function*

Description

Get a reference by its ID.

Usage

```
getReference(host, referenceId)
```

Arguments

`host`	URL of GA4GH API data server.
`referenceId`	The ID of the Reference to be retrieved.

Details

This function requests `GET host/references/referenceId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchReferences`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
referenceSetId <- searchReferenceSets(host, nrows = 1)$id
referenceId <- searchReferences(host, referenceSetId, nrows = 1)$id
getReference(host, referenceId)

## End(Not run)
```

`getReferenceSet` *getReferenceSet function*

Description

Get a reference set by its ID.

Usage

```
getReferenceSet(host, referenceSetId)
```

Arguments

`host`	URL of GA4GH API data server.
`referenceSetId`	The ID of the ReferenceSet to be retrieved.

Details

This function requests `GET host/referencesets/referenceSetId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchReferenceSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
referenceSetId <- searchReferenceSets(host, nrows = 1)$id
getReferenceSet(host, referenceSetId)

## End(Not run)
```

`getRnaQuantification` *getRnaQuantification function*

Description

Get an RNA quantification by its ID.

Usage

```
getRnaQuantification(host, rnaQuantificationId)
```

Arguments

`host`	URL of GA4GH API data server.
`rnaQuantificationId`	ID of the RNA quantification requested.

Details

This function requests `GET host/rnaquantifications/rnaQuantificationId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchRnaQuantifications`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
rnaQuantificationSetId <- searchRnaQuantificationSets(host, datasetId, nrows = 1)$id
rnaQuantificationId <- searchRnaQuantifications(host, rnaQuantificationSetId, nrows = 1)$id
getRnaQuantification(host, rnaQuantificationId)

## End(Not run)
```

`getRnaQuantificationSet` *getRnaQuantificationSet function*

Description

Get an RNA quantification set by its ID.

Usage

```
getRnaQuantificationSet(host, rnaQuantificationSetId)
```

Arguments

`host`	URL of GA4GH API data server.
`rnaQuantificationSetId`	ID of the RNA quantification set requested.

Details

This function requests `GET host/rnaquantificationsets/rnaQuantificationSetId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchRnaQuantificationSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
rnaQuantificationSetId <- searchRnaQuantificationSets(host, datasetId, nrows = 1)$id
getRnaQuantificationSet(host, rnaQuantificationSetId)

## End(Not run)
```

`getVariant` *getVariant function*

Description

Get a variant by its ID with all call sets for this variant.

Usage

```
getVariant(host, variantId, asVCF = TRUE)
```

Arguments

`host`	URL of GA4GH API data server.
`variantId`	The ID of the Variant to be retrieved.
`asVCF`	If `TRUE` the function will return an `VCF` with header (default), otherwise it will return an `DataFrame`.

Details

This function requests `GET host/variants/variantId`.

Value

`VCF` object (when `asVCF = TRUE`) or `DataFrame` object (otherwise).

References

Official documentation.

See Also

`DataFrame`, `searchVariants`, `searchVariantsByGRanges`, `VCF`, `makeVCFFromGA4GHResponse`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id
variantId <- searchVariants(host, variantSetId, "1", 15031, 15031)$id
getVariant(host, variantId)

getVariant(host, variantId, asVCF = FALSE)

## End(Not run)
```

`getVariantAnnotationSet` *Gets getVariantAnnotationSet function*

Description

Get a variant annotation set by its ID.

Usage

```
getVariantAnnotationSet(host, variantAnnotationSetId)
```

Arguments

`host`	URL of GA4GH API data server.
`variantAnnotationSetId`	ID of variant annotation set.

Details

This function requests `GET host/variantannotationsets/variantAnnotationSetId`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `searchVariantAnnotationSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 2)$id[2]
id <- searchVariantAnnotationSets(host, variantSetId, nrows = 1)$id
getVariantAnnotationSet(host, variantAnnotationSetId = id)

## End(Not run)
```

`getVariantSet` *getVariantSet function.*

Description

Get a variant set by its ID.

Usage

```
getVariantSet(host, variantSetId, asVCFHeader = TRUE)
```

Arguments

`host`	URL of GA4GH API data server.
`variantSetId`	The ID of the VariantSet to be retrieved.
`asVCFHeader`	If `TRUE` the function will return an `VCFHeader` object (default), otherwise it will return an `DataFrame`.

Details

This function requests `GET` `host/variantsets/variantSetId`.

Value

`DataFrame` object. It can be converted into `VCFHeader` object.

See Also

`DataFrame`, `searchVariantSets`, `VCFHeader`, `makeVCFHeaderFromGA4GHResponse`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id
getVariantSet(host, variantSetId)

getVariantSet(host, variantSetId, asVCF = FALSE)

## End(Not run)
```

`HGVSnames` *Generate genomic variant data to HGVS nomenclature*

Description

This function follows the official reference HGVS nomenclature. At this moment, it supports only 'substitution' and 'indel' for DNA sequences.

Usage

```
HGVSnames(start, ref, alt, type = "g", seqnames = NA_character_)
```

Arguments

`start`	genomic location of start
`ref`	reference sequence
`alt`	alternate sequence
`type`	Sequence type to be used as prefix. Allowed options are: • `g` genomic (default); • `m` mitochondrial; • `c` coding DNA; • `n` non-coding DNA.
`seqnames`	name of sequence (e.g. chr1, 1). It is optional.

Value

Genomic coordinates of variants formatted as HGVS nomenclature.

References

Sequence Variant Nomenclature.

Examples

```
start <- c(45576, "88+1", 6775, 6775, 145, 9002, 4, 12345611, 58347698)
ref <- c("A", "G", "T", "TCA", "CGA", "AAAAAAAA", "GC", "G", "A")
alt <- c("C", "T", "GA", "C", "TGG", "TTT", "TG", "A", "*")
type <- c("g", "c", "g", "g", "c", "g", "g", "g", "g")
seqnames <- c("", "", NA, NA, NA, NA, NA, "chr11", NA)
HGVSnames(start, ref, alt, type, seqnames)
```

`listReferenceBases` *listReferenceBases function*

Description

Get the sequence bases of a reference genome by genomic range.

Usage

```
listReferenceBases(host, referenceId, start = 1, end = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`referenceId`	The ID of the Reference to be retrieved.
`start`	The start position (1-based) of this query. Defaults to 0. Genomic positions are non-negative integers less than reference length. Requests spanning the join of circular genomes are represented as two requests one on each side of the join (position 1).
`end`	The end position (1-based, inclusive) of this query. Defaults to the length of this Reference.

Details

This function requests `POST` `host/listreferencebases`.

Value

`BString` object.

See Also

`searchReferenceSets`, `searchReferences`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
referenceSetId <- searchReferenceSets(host, nrows = 1)$id
referenceId <- searchReferences(host, referenceSetId, nrows = 1)$id
listReferenceBases(host, referenceId, start = 1, end = 100)

## End(Not run)
```

`makeVCFFromGA4GHResponse`

makeVCFFromGA4GHResponse function

Description

Convert `DataFrame` output from `searchVariants` and `getVariant` functions to `VCF` class.

Usage

```
makeVCFFromGA4GHResponse(variants)
```

Arguments

`variants`	`DataFrame` generated by `searchVariants`.

Value

`VCF` object.

See Also

`searchVariants`, `getVariant`, `VCF`, `DataFrame`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id
variants <- searchVariants(host, variantSetId, referenceName = "1",
    start = 15000, end = 16000)
variants

makeVCFFromGA4GHResponse(variants)

## End(Not run)
```

`makeVCFHeaderFromGA4GHResponse`

makeVCFHeaderFromGA4GHResponse function

Description

Convert `DataFrame` output from `getVariantSet` function to `VCFHeader` class.

Usage

```
makeVCFHeaderFromGA4GHResponse(variantSet)
```

Arguments

`variantSet`	`DataFrame` generated by `getVariantSet` function.

Value

`VCFHeader` object.

See Also

`getVariantSet`, `VCFHeader`, `DataFrame`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id
variantId <- searchVariants(host, variantSetId, "1", 15031, 15031)$id
variant <- getVariant(host, variantId)

makeVCFFromGA4GHResponse(variant)

## End(Not run)
```

`searchBiosamples` *searchBiosamples function*

Description

This function gets Biosamples matching the search criteria.

Usage

```
searchBiosamples(host, datasetId, name = NA_character_,
  individualId = NA_character_, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	Id of the dataset to search.
`name`	Returns Biosamples with the given name found by case-sensitive string matching.
`individualId`	Returns Biosamples for the provided individual ID.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `/biosamples/search`.

Value

`DataFrame` object. `NULL` means no registry found.

References

Official documentation.

See Also

`DataFrame`, `getBiosample`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
searchBiosamples(host, datasetId, nrows = 10)

## End(Not run)
```

`searchCallSets` *searchCallSets function*

Description

Search for call sets (sample columns of VCF files).

Usage

```
searchCallSets(host, variantSetId, name = NA_character_,
  biosampleId = NA_character_, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`variantSetId`	The VariantSet to search.
`name`	Only return call sets with this name (case-sensitive, exact match).
`biosampleId`	Return only call sets generated from the provided BioSample ID.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST` `host/callsets/search`.

Value

`DataFrame` object.

See Also

`DataFrame`, `getCallSet`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id
searchCallSets(host, variantSetId)

## End(Not run)
```

`searchDatasets` *searchDatasets function*

Description

Search for datasets.

Usage

```
searchDatasets(host, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `POST` `/datasets/search`.

Value

`DataFrame` object. `NULL` means no registry found.

See Also

`DataFrame`, `getDataset`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
searchDatasets(host)

## End(Not run)
```

`searchExpressionLevels`

searchExpressionLevels function

Description

This function gets expression levels matching the search criteria.

Usage

```
searchExpressionLevels(host, rnaQuantificationId, nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`rnaQuantificationId`	Id of the rnaQuantification to restrict search to.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `/expressionlevels/search`.

Value

`DataFrame` object. `NULL` means no registry found.

References

Official documentation.

See Also

`DataFrame`, `getExpressionLevel`, `searchRnaQuantificationSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
rnaQuantificationSetId <- searchRnaQuantificationSets(host, datasetId, nrow = 1)$id
rnaQuantificationId <- searchRnaQuantifications(host, rnaQuantificationSetId, nrows = 1)$id
searchExpressionLevels(host, rnaQuantificationId, nrows = 10)

## End(Not run)
```

`searchFeatures` *searchFeatures function*

Description

Search for features (lines of genomic feature files).

Usage

```
searchFeatures(host, featureSetId, name = NA_character_,
  geneSymbol = NA_character_, parentId = NA_character_,
  referenceName = NA_character_, start = NA_integer_, end = NA_integer_,
  featureTypes = character(), nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`featureSetId`	The annotation set to search within. Either featureSetId or parentId must be non-empty.
`name`	Only returns features with this name (case-sensitive, exact match).
`geneSymbol`	Only return features with matching the provided gene symbol (case-sensitive, exact match). This field may be replaced with a more generic representation in a future version.
`parentId`	Restricts the search to direct children of the given parent feature ID. Either feature_set_id or parent_id must be non-empty.
`referenceName`	Only return features on the reference with this name (matched to literal reference name as imported from the GFF3).
`start`	Required, if name or symbol not provided. The beginning of the window (0-based, inclusive) for which overlapping features should be returned. Genomic positions are non-negative integers less than reference length. Requests spanning the join of circular genomes are represented as two requests one on each side of the join (position 0).

`end`	Required, if name or symbol not provided. The end of the window (0-based, exclusive) for which overlapping features should be returned.
`featureTypes`	TODO: To be replaced with a fully featured ontology search once the Metadata definitions are rounded out. If specified, this query matches only annotations whose feature_type matches one of the provided ontology terms.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST` `host/features/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `getFeature`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
featureSetId <- searchFeatureSets(host, datasetId, nrows = 1)$id
searchFeatures(host, featureSetId, nrows = 10)

## End(Not run)
```

`searchFeatureSets` *searchFeatureSets function*

Description

Search for feature sets (genomic features, e.g. GFF files).

Usage

```
searchFeatureSets(host, datasetId, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	The Dataset to search.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST host/featuresets/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `getFeatureSet`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
searchFeatureSets(host, datasetId)

## End(Not run)
```

`searchIndividuals` *searchIndividuals function*

Description

This function gets individuals matching the search criteria.

Usage

```
searchIndividuals(host, datasetId, name = NA_character_, nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	Id of the dataset to search.
`name`	Returns Individuals with the given name found by case-sensitive string matching.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `/individuals/search`.

Value

`DataFrame` object. `NULL` means no registry found.

References

Official documentation.

See Also

`DataFrame`, `getIndividual`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
searchIndividuals(host, datasetId, nrows = 10)

## End(Not run)
```

---

searchPhenotypeAssociations

searchPhenotypeAssociations function

Description

This function gets a list of phenotype associations matching the search criteria.

Usage

```
searchPhenotypeAssociations(host, phenotypeAssociationSetId,
  featureIds = character(), phenotypeIds = character(), nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`phenotypeAssociationSetId`	Id of the PhenotypeAssociationSet to search.
`featureIds`	Ids of the features. At least one `featureId` or `phenotypeId` must be provided.
`phenotypeIds`	Ids of the phenotypes. At least one `featureId` or `phenotypeId` must be provided.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `/featurephenotypeassociations/search`.

Value

`DataFrame` object. `NULL` means no registry found.

References

Official documentation.

See Also

`DataFrame`, `searchPhenotypeAssociationSets`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
id <- searchPhenotypeAssociationSets(host, datasetId, nrows = 1)$id
searchPhenotypeAssociations(host, id, nrows = 10)

## End(Not run)
```

`searchPhenotypeAssociationSets`

searchPhenotypeAssociationSets function

Description

This function gets a list of association sets matching the search criteria.

Usage

```
searchPhenotypeAssociationSets(host, datasetId, nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	Id of the dataset to search.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `/phenotypeassociationsets/search`.

Value

`DataFrame` object. `NULL` means no registry found.

References

Official documentation.

See Also

`DataFrame`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
searchPhenotypeAssociationSets(host, datasetId, nrows = 10)

## End(Not run)
```

`searchReadGroupSets` *searchReadGroupSets function*

Description

Search for read group sets (sequence alignement, e.g BAM files).

Usage

```
searchReadGroupSets(host, datasetId, name = NA_character_,
  biosampleId = NA_character_, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	The dataset to search.
`name`	Only return read group sets with this name (case-sensitive, exact match).
`biosampleId`	Specifying the id of a BioSample record will return only readgroups with the given biosampleId.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST` `host/readgroupsets/search`

Value

`DataFrame` object.

See Also

`DataFrame`, `getReadGroupSet`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
searchReadGroupSets(host, datasetId, nrows = 1)

## End(Not run)
```

`searchReads` *searchReads function*

Description

Search for reads by genomic range (bases of aligned sequences)

Usage

```
searchReads(host, readGroupIds, referenceId = NA_character_,
  start = NA_integer_, end = NA_integer_, nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`readGroupIds`	The ReadGroups to search. At least one id must be specified.
`referenceId`	The reference to query. Leaving blank returns results from all references, including unmapped reads - this could be very large.
`start`	The start position (1-based) of this query. If a reference is specified, this defaults to 0. Genomic positions are non-negative integers less than reference length. Requests spanning the join of circular genomes are represented as two requests one on each side of the join (position 1).
`end`	The end position (1-based, exclusive) of this query. If a reference is specified, this defaults to the reference's length.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST host/reads/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
readGroupIds <- "WyIxa2dlbm9tZXMiLCJyZ3MiLCJIRzAzMjcwIiwiRVJSMTgxMzI5Il0"
referenceSetId <- searchReferenceSets(host, nrows = 1)$id
referenceId <- searchReferences(host, referenceSetId, nrows = 1)$id
searchReads(host, readGroupIds, referenceId, start = 15000, end = 16000)

## End(Not run)
```

`searchReferences` *searchReferences function*

Description

Search for references (genome sequences, e.g. chromosomes).

Usage

```
searchReferences(host, referenceSetId, md5checksum = NA_character_,
  accession = NA_character_, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`referenceSetId`	The ReferenceSet to search.
`md5checksum`	If specified, return the references for which the md5checksum matches this string (case-sensitive, exact match). See ReferenceSet::md5checksum for details.
`accession`	If specified, return the references for which the accession matches this string (case-sensitive, exact match).
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST host/references/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `getReference`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
referenceSetId <- searchReferenceSets(host, nrows = 1)$id
searchReferences(host, referenceSetId)

## End(Not run)
```

`searchReferenceSets` *searchReferenceSets function*

Description

Search for reference sets (reference genomes).

Usage

```
searchReferenceSets(host, md5checksum = NA_character_,
  accession = NA_character_, assemblyId = NA_character_, nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`md5checksum`	If unset, return the reference sets for which the md5checksum matches this string (case-sensitive, exact match). See ReferenceSet::md5checksum for details.
`accession`	If unset, return the reference sets for which the accession matches this string (case-sensitive, exact match).
`assemblyId`	If unset, return the reference sets for which the assemblyId matches this string (case-sensitive, exact match).
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST` `host/references/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `getReferenceSet`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
searchReferenceSets(host)

## End(Not run)
```

`searchRnaQuantifications`

searchRnaQuantifications function

Description

This function gets a list of RnaQuantifications matching the search criteria.

Usage

```
searchRnaQuantifications(host, rnaQuantificationSetId,
  biosampleId = NA_character_, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`rnaQuantificationSetId`	IReturn only Rna Quantifications which belong to this set.
`biosampleId`	Return only RNA quantifications regarding the specified biosample.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `/rnaquantifications/search`.

Value

`DataFrame` object. `NULL` means no registry found.

References

Official documentation.

See Also

`DataFrame`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
id <- searchRnaQuantificationSets(host, datasetId, nrows = 1)$id
searchRnaQuantifications(host, rnaQuantificationSetId = id)

## End(Not run)
```

`searchRnaQuantificationSets` *searchRnaQuantificationSets function*

Description

This function gets a list of RNA quantification sets matching the search criteria.

Usage

```
searchRnaQuantificationSets(host, datasetId, nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	Id of the dataset to search.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests to `/rnaquantificationsets/search`.

Value

`DataFrame` object. `NULL` means no registry found.

References

Official documentation.

See Also

`DataFrame`, `getRnaQuantificationSet`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
searchRnaQuantificationSets(host, datasetId, nrows = 1)

## End(Not run)
```

`searchVariantAnnotations` *searchVariantAnnotations function*

Description

Search for annotated variants by genomic range.

Usage

```
searchVariantAnnotations(host, variantAnnotationSetId,
  referenceName = NA_character_, referenceId = NA_character_,
  start = NA_integer_, end = NA_integer_, effects = list(), nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`variantAnnotationSetId`	Required. The ID of the variant annotation set to search over.
`referenceName`	Only return variants with reference alleles on the reference with this name. One of this field or reference_id is required.
`referenceId`	Only return variants with reference alleles on the reference with this ID. One of this field or reference_name is required.

`start`	Required if reference_name or reference_id supplied. The beginning of the window (1-based, inclusive) for which variants with overlapping reference alleles should be returned. Genomic positions are non-negative integers less than reference length. Requests spanning the join of circular genomes are represented as two requests one on each side of the join (position 1).
`end`	Required if reference_name or reference_id supplied. The end of the window (1-based, exclusive) for which variants with overlapping reference alleles should be returned.
`effects`	This filter allows variant, transcript combinations to be extracted by effect type(s). Only return variant annotations including any of these effects and only return transcript effects including any of these effects. Exact matching across all fields of the Sequence Ontology OntologyTerm is required. (A transcript effect may have multiple SO effects which will all be reported.) If empty, return all variant annotations.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function requests `POST host/variantannotations/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 2)$id[2]
id <- searchVariantAnnotationSets(host, variantSetId, nrows = 1)$id
searchVariantAnnotations(host, variantAnnotationSetId = id,
    referenceName = "1", start = 15000, end = 16000)

## End(Not run)
```

`searchVariantAnnotationSets`

searchVariantAnnotationSets function

Description

Search for variant annotation sets (annotated VCF files).

Usage

```
searchVariantAnnotationSets(host, variantSetId, nrows = Inf,
  responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`variantSetId`	Required. The VariantSet to search.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This function maps to `POST` `host/variantannotationsets/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 2)$id[2]
searchVariantAnnotationSets(host, variantSetId)

## End(Not run)
```

`searchVariants` *searchVariants function*

Description

Search for variants by genomic ranges (lines of VCF files).

Usage

```
searchVariants(host, variantSetId, referenceName, start, end,
  callSetIds = character(), nrows = Inf, responseSize = NA_integer_,
  asVCF = TRUE)
```

Arguments

`host`	URL of GA4GH API data server.
`variantSetId`	The variant set to search.
`referenceName`	Required. Only return variants on this reference.
`start`	Required. The beginning of the window (1-based, inclusive) for which overlapping variants should be returned. Genomic positions are non-negative integers less than reference length. Requests spanning the join of circular genomes are represented as two requests one on each side of the join (position 1).
`end`	Required. The end of the window (1-based, inclusive) for which overlapping variants should be returned.
`callSetIds`	Only return variant calls which belong to callsets with these IDs. If unspecified, return all variants and no variant call objects.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.
`asVCF`	If `TRUE` the function will return an `VCF` with header (default), otherwise it will return an `DataFrame`.

Details

This function maps to `POST host/variants/search`.

Value

`VCF` object (when `asVCF = TRUE`) or `DataFrame` object (otherwise).

References

Official documentation.

See Also

`DataFrame`, `getVariant`, `searchVariantsByGRanges`, `VCF`, `makeVCFFromGA4GHResponse`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id
searchVariants(host, variantSetId, referenceName = "1",
    start = 15000, end = 16000)

searchVariants(host, variantSetId, referenceName = "1",
    start = 15000, end = 16000, asVCF = FALSE)

## End(Not run)
```

`searchVariantsByGRanges`

searchVariantsByGranges function

Description

Search for variants by genomic ranges (lines of VCF files)

Usage

```
searchVariantsByGRanges(host, variantSetId, granges, callSetIds = character(),
  nrows = Inf, responseSize = NA_integer_, asVCF = FALSE)
```

Arguments

`host`	URL of GA4GH API data server.
`variantSetId`	The variant set to search.
`granges`	A GRanges object containing one or more genomic ranges.
`callSetIds`	Only return variant calls which belong to callsets with these IDs. If unspecified, return all variants and no variant call objects.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.
`asVCF`	If `TRUE` the function will return a list of `VCF` object with headers (default), otherwise it will return a list of `DataFrame` objects.

Details

This function maps to the body of `POST host/variants/search`.

Value

List of `VCF` objects (when `asVCF = TRUE`) or a list of `DataFrame` objects (otherwise). Each row in GRanges object will be a element of the list.

References

Official documentation.

See Also

`DataFrame`, `searchVariants` `getVariant`, `VCF`

Examples

```
library(GenomicRanges)
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
variantSetId <- searchVariantSets(host, datasetId, nrows = 1)$id[1]
granges <- GRanges(seqnames = "1", IRanges(start = 15000, end = 16000))
searchVariantsByGRanges(host, variantSetId, granges)

## End(Not run)
```

`searchVariantSets` *searchVariantSets function*

Description

Search for for variant sets (VCF files).

Usage

```
searchVariantSets(host, datasetId, nrows = Inf, responseSize = NA_integer_)
```

Arguments

`host`	URL of GA4GH API data server.
`datasetId`	Id of the dataset to search.
`nrows`	Number of rows of the data frame returned by this function. If not defined, the function will return all entries. If the number of available entries is less than the value of this this parameter, the function will silently return only the available entries.
`responseSize`	Specifies the number of entries to be returned by the server until reach the number of rows defined in `nrows` parameter or until get all available entries. If not defined, the server will return the allowed maximum reponse size. Increasing this the value of this parameter will reduce the number of requests and reducing the time required. The will not respect this parameter if the value if larger than its maximum response size.

Details

This request maps to the body of `POST` `host/variantsets/search`.

Value

`DataFrame` object.

References

Official documentation.

See Also

`DataFrame`, `getVariantSet`

Examples

```
host <- "http://1kgenomes.ga4gh.org/"
## Not run:
datasetId <- searchDatasets(host, nrows = 1)$id
searchVariantSets(host, datasetId)

## End(Not run)
```

Index