Package 'geomedb'

July 15, 2020

Type Package

Title Functions for Fetching 'GeOMe-db' Data

Version 2.0.1

Date 2020-06-26

Description
The Genomic Observatory Metadatabase (GeOMe Database) is an open access repository for geographic and ecological metadata associated with sequenced samples. This package is used to retrieve
GeOMe data for analysis. See <http://www.geome-db.org> for more information regarding GeOMe.

URL http://www.geome-db.org,
https://github.com/biocodellc/fimsR-access

BugReports https://github.com/biocodellc/fimsR-access/issues

License GPL-3

LazyData TRUE

Encoding UTF-8

Imports ape, httr, jsonlite

Depends utils

RoxygenNote 7.1.1

NeedsCompilation no

Author RJ Ewing [aut, cre],
Eric Crandall [aut]

Maintainer RJ Ewing <rj@rjewing.com>

Repository CRAN

Date/Publication 2020-07-15 13:30:16 UTC

R topics documented:

`fasterqDump` *Download or convert fastq data from NCBI Sequence Read Archive using multiple threads*

Description

'fasterqDump()' uses the SRAtoolkit command-line function 'fasterq-dump' to download fastq files from all samples returned by a `queryMetadata` query of GEOME, when one of the entities queried was 'fastqMetadata'

Usage

```
fasterqDump(queryMetadata_object, sratoolkitPath = "",
  outputDirectory = "./", arguments = "-p", filenames = "accessions",
  source = "sra", cleanup = FALSE, fasterqDumpHelp = FALSE)
```

Arguments

`queryMetadata_object`
: A list object returned from 'queryMetadata' where one of the entities queried was 'fastqMetadata'.

`sratoolkitPath`
: String. A path to a local copy of sratoolkit. Only necessary if sratoolkit is not on your $PATH. Assumes executables are inside 'bin'.

`outputDirectory`
: String. A path to the directory where you would like the files to be stored.

`arguments`
: A string variable of arguments to be passed directly to 'fasterq-dump'. Defaults to "-p" to show progress. Use fasterqDumpHelp = TRUE to see a list of arguments.

`filenames`
: String. How would you like the downloaded fastq files to be named? "accessions" names files with SRA accession numbers "IDs" names files with their materialSampleID "locality_IDs" names files with their locality and materialSampleID.

`source` String. ‘fasterq-dump‘ can retrieve files directly from SRA, or it can convert .sra files previously downloaded with ‘prefetch‘ that are in the current working directory. "sra" downloads from SRA "local" converts .sra files in the current working directory.

`cleanup` Logical. cleanup = T will delete any intermediate .sra files.

`fasterqDumpHelp` Logical. fasterqDumpHelp = T will show the help page for ‘fasterq-dump‘ and then quit.

Details

The ‘fasterq-dump‘ tool uses temporary files and multi-threading to speed up the extraction of FASTQ from SRA-accessions. This function works best with sratoolkit functions of version 2.9.6 or greater. SRAtoolkit functions can (ideally) be in your $PATH, or you can supply a path to them using the sratoolkitPath argument.

‘fasterqDump()‘ downloads files to the current working directory unless a different one is assigned through outputDirectory.

Change the number of threads by adding "-e X" to arguments where X is the number of threads.

‘fasterq-dump‘ will automatically split paired-end data into three files with:

- file_1.fastq having read 1
- file_2.fastq having read 2
- file.fastq having unmatched reads

‘fasterqDump()‘ can then rename these files based on their materialSampleID and locality.

Note that ‘fasterq-dump‘ will store temporary files in ~/ncbi/public/sra by default unless you pass "-t /path/to/temp/dir" to arguments. *Make sure to periodically delete these temporary files.*

Value

This function will not return anything within r. It simply downloads fastq files. It will print command line stdout to the console, and also provide a start and end time and amount of time elapsed during the download.

See Also

https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc to download pre-compiled executables for sratoolkit or https://github.com/ncbi/sra-tools/wiki/Building-and-Installing-from-S to install from source

This function will not work on Windows systems because fasterq-dump is not currently available for Windows. See `fastqDump` if you use Windows. See `prefetch` to download .sra files prior to converting them locally.

Examples

```
## Not run:
# Run a query of GEOME first
acaoli <- queryMetadata(
    entity = "fastqMetadata",
    query = "genus = Acanthurus AND specificEpithet = olivaceus AND _exists_:bioSample",
    select=c("Event"))

#trim to 3 entries for expediency
acaoli$fastqMetadata<-acaoli$fastqMetadata[1:3,]
acaoli$Event<-acaoli$Event[1:3,]

# Download straight from SRA, naming files with their locality and materialSampleID
fasterqDump(queryMetadata_object = acaoli, filenames = "IDs", source = "sra")

# A generally faster option is to run prefetch first, followed by fasterqDump, with cleanup = T to
# remove the prefetched .sra files.
prefetch(queryMetadata_object = acaoli)
fasterqDump(queryMetadata_object = acaoli, filenames = "IDs", source = "local", cleanup = T)

## End(Not run)
```

`fastqDump` *Download or convert fastq data from NCBI Sequence Read Archive in a single thread (Windows compatible)*

Description

'fastqDump()' uses the SRAtoolkit command-line function 'fastq-dump' to download fastq files from all samples returned by a `queryMetadata` query of GEOME, when one of the entities queried was 'fastqMetadata'

Usage

```
fastqDump(queryMetadata_object, sratoolkitPath = "",
  outputDirectory = ".", arguments = "-v --split-3",
  filenames = "accessions", source = "sra", cleanup = FALSE,
  fastqDumpHelp = FALSE)
```

Arguments

`queryMetadata_object`
: A list object returned from 'queryMetadata' where one of the entities queried was 'fastqMetadata'.

`sratoolkitPath`
: String. A path to a local copy of sratoolkit. Only necessary if sratoolkit is not on your $PATH. Assumes executables are inside 'bin'.

`outputDirectory`
: String. A path to the directory where you would like the files to be stored.

arguments	A string variable of arguments to be passed directly to 'fastq-dump'. Defaults to "-v –split 3" to show progress and split paired-end data. Use fastqDumpHelp = TRUE to see a list of arguments.
filenames	String. How would you like the downloaded fastq files to be named? "accessions" names files with SRA accession numbers "IDs" names files with their materialSampleID "locality_IDs" names files with their locality and materialSampleID.
source	String. 'fastq-dump' can retrieve files directly from SRA, or it can convert .sra files previously downloaded with 'prefetch' that are in the current working directory. "sra" downloads from SRA "local" converts .sra files in the current working directory.
cleanup	Logical. cleanup = T will delete any intermediate .sra files.
fastqDumpHelp	Logical. fastqDumpHelp = T will show the help page for 'fastq-dump' and then quit.

Details

This function works best with sratoolkit functions of version 2.9.6 or greater. SRAtoolkit functions can (ideally) be in your $PATH, or you can supply a path to them using the sratoolkitPath argument.

'fastqDump()' downloads files to the current working directory unless a different one is assigned through outputDirectory.

'fastq-dump' will automatically split paired-end data into three files with:

- file_1.fastq having read 1
- file_2.fastq having read 2
- file.fastq having unmatched reads

Value

This function will not return anything within r. It simply downloads fastq files. It will print command line stdout to the console, and also provide a start and end time and amount of time elapsed during the download.

See Also

https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc to download precompiled executables for sratoolkit or https://github.com/ncbi/sra-tools/wiki/Building-and-Installing-from-S to install from source

See prefetch to download .sra files prior to converting them locally. This two step process works faster than just using 'fastqDump()'. See fasterqDump for a faster, multithreaded version of 'fastqDump()' that does not work on Windows.

Examples

```
## Not run:
# Run a query of GEOME first
acaoli <- queryMetadata(
```

```
    entity = "fastqMetadata",
    query = "genus = Acanthurus AND specificEpithet = olivaceus AND _exists_:bioSample",
    select=c("Event"))

#trim to 3 entries for expediency
acaoli$fastqMetadata<-acaoli$fastqMetadata[1:3,]
acaoli$Event<-acaoli$Event[1:3,]

# Download straight from SRA, naming files with their locality and materialSampleID
fastqDump(queryMetadata_object = acaoli, filenames = "locality_IDs", source = "sra")

# A generally faster option is to run prefetch first, followed by fastqDump, with cleanup = T to
# remove the prefetched .sra files.
prefetch(queryMetadata_object = acaoli)
fastqDump(queryMetadata_object = acaoli, filenames = "IDs", source = "local", cleanup = T)

## End(Not run)
```

`listEntities` *Get a list of entities (i.e. tables) available to query*

Description

Get a list of entities (i.e. tables) available to query

Usage

```
listEntities(projectId = NA)
```

Arguments

`projectId`	(optional) The project to fetch entities for. If not provided, the network entities will be returned.

Examples

```
## Not run:
entities <- listEntities(projectId)
entities <- listEntities()

## End(Not run)
```

listExpeditions *Get a list of expeditions for a GEOME project*

Description

Get a list of expeditions for a GEOME project

Usage

```
listExpeditions(projectId)
```

Arguments

projectId	The project to list expeditions for.

Examples

```
## Not run:
expeditions <- listExpeditions(projectId)

## End(Not run)
```

listLoci *Get a list of loci that are stored in FASTA format directly in GEOME (not in the SRA)*

Description

Get a list of loci that are stored in FASTA format directly in GEOME (not in the SRA)

Usage

```
listLoci()
```

Examples

```
## Not run:
markers <- listLoci()

## End(Not run)
```

listProjects *Get a list of projects in GEOME*

Description

Get a list of projects in GEOME

Usage

```
listProjects()
```

Examples

```
## Not run:
projects <- listProjects()

## End(Not run)
```

prefetch *Download data from NCBI Sequence Read Archive in .sra format using FASP or HTTPS protocols*

Description

'prefetch()' uses the SRAtoolkit command-line function 'prefetch' to download .sra files from all samples returned by a queryMetadata query of GEOME, when one of the entities queried was 'fastqMetadata'

Usage

```
prefetch(queryMetadata_object, sratoolkitPath = "",
  outputDirectory = ".", arguments = "-p 1", prefetchHelp = FALSE)
```

Arguments

queryMetadata_object	A list object returned from 'queryMetadata' where one of the entities queried was 'fastqMetadata'.
sratoolkitPath	String. A path to a local copy of sratoolkit. Only necessary if sratoolkit is not on your $PATH. Assumes executables are inside 'bin'.
outputDirectory	String. A path to the directory where you would like the files to be stored.
arguments	A string variable of arguments to be passed directly to 'prefetch'. Defaults to "-p 1" to show progress. Use prefetchHelp = TRUE to see a list of arguments.
prefetchHelp	Logical. prefetchHelp = T will show the help page for 'prefetch' and then quit.

Details

This function works best with SRAtoolkit functions of version 2.9.6 or greater. SRAtoolkit functions can (ideally) be in your $PATH, or you can supply a path to them using the sratoolkitPath argument.

It downloads files to the current working directory unless a different one is assigned through outputDirectory.

'prefetch' will automatically use the Fast and Secure Protocol (FASP) in the Aspera Connect package if the 'ascp' executable is in your $PATH. Otherwise it will use HTTPS.

You can alternatively pass the path to 'ascp' by using arguments="-a path/to/ascp"

Value

This function will not return anything within r. It simply downloads .sra files. It will print command line stdout to the console, and also provide a start and end time and amount of time elapsed during the download.

See Also

https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc to download precompiled executables for sratoolkit or https://github.com/ncbi/sra-tools/wiki/Building-and-Installing-from-S to install from source

Use 'prefetch' in combination with fastqDump or fasterqDump to convert .sra files to .fastq.

Examples

```
## Not run:
# Run a query of GEOME first
acaoli <- queryMetadata(
   entity = "fastqMetadata",
   query = "genus = Acanthurus AND specificEpithet = olivaceus AND _exists_:bioSample",
   select=c("Event"))

#trim to 3 entries for expediency
acaoli$fastqMetadata<-acaoli$fastqMetadata[1:3,]
acaoli$Event<-acaoli$Event[1:3,]


prefetch(queryMetadata_object = acaoli)

fastqDump(queryMetadata_object = acaoli, filenames = "IDs", source = "local", cleanup = T)

## End(Not run)
```

queryMetadata *Query metadata from the GEOME database.*

Description

'queryMetadata' uses HTTP to query metadata from the GEOME database. If you are looking to download associated sequences from the SRA, you must include 'fastqMetadata' as one of the entities searched (this is done by default) and you should include "_exists_:bioSample" within your query statement to find only samples with associated SRA sequences

Usage

```
queryMetadata(entity = "Sample", projects = list(),
  expeditions = list(), select = list("fastqMetadata"), query = "",
  source = NULL, page = 0, limit = "10000")
```

Arguments

entity	String. The entity or entities (tables) to query. One of ('Event', 'Sample', 'Tissue', 'Sample_Photo', 'Event_Photo','fastqMetadata'). Default is to query 'Sample'
projects	List of projects to include in the query. The default is all projects
expeditions	Only applicable if projects are specified. list of expeditions to include in the query. The default is all expeditions
select	List of entities to include in the response. One of ('Event', 'Sample', 'Tissue', 'Sample_Photo', 'Event_Photo','fastqMetadata') The @param 'entity' will always be included in the response. 'fastqMetadata' included by default.
query	FIMS Query statement http://fims.readthedocs.io/en/latest/fims/query.html query string. Ex. 'yearCollected >= 2017 and country = "Indonesia"'. Your query must include "_exists_:bioSample" to find samples that have associated data in the SRA.
source	List of column names to include in the data.frame results. If there is no entity prefix, the column is assumed to belong to the @param 'entity'. Ex. list('Event.eventID', 'Event.locality', 'materialSampleID', 'bcid', 'Event.bcid') 'materialSampleID' and 'bcid' in the above list are assumed to belong to the @param 'entity'
page	The results page to return. Used to offset the page for large result sets. Defaults to 0.
limit	The number of results to include in the response. Defaults to 10000

Value

a list object with each entity (table) as a dataframe object

Examples

```
## Not run:
df <- queryMetadata('Sample', projects=list(1), expeditions=list("acaach_CyB_JD", "acajap_CyB_JD"))
df <- queryMetadata('Sample', names=list("materialSampleID", "bcid"), query="Chordata")
df <- queryMetadata('Sample', projects=list(1), expeditions=list("acajap_CyB_JD"),
                    names=list("bcid"), query="yearCollected=2008")
df <- queryMetadata('Sample', select=list('Event', 'Tissue'), names=list("bcid"),
                    query="yearCollected=2008")
df <- queryMetadata('fastqMetadata', select=list('Event', 'Sample', 'Tissue'),
                    query="_exists_:bioSample")
acaoli <- queryMetadata(
    entity = "fastqMetadata",
    query = "genus = Acanthurus AND specificEpithet = olivaceus AND _exists_:bioSample",
    select=c("Event"))

## End(Not run)
```

---

querySanger *Query Sanger sequences directly from the GEOME database*

Description

For Sanger sequence data (typically of mitochondrial origin), it is possible to store the sequence directly within GEOME. ‘querySanger()‘ allows you to download this sequence data into a DNAbin object, as well as to your working directory as a FASTA-formatted file.

Usage

```
querySanger(locus, projects = list(), expeditions = list(),
  query = "")
```

Arguments

locus	the locus to fetch. list of markers can be found by calling ‘listLoci()‘
projects	list of projects to include in the query. The default is all projects
expeditions	Only applicable if projects are specified. list of expeditions to include in the query. The default is all expeditions
query	FIMS Query DSL http://fims.readthedocs.io/en/latest/fims/query.html query string. Ex. ’yearCollected >= 2017 and country = "Indonesia"’

Value

a DNAbin object, which is a fairly standard form for storing DNA data in binary format. It will also download a FASTA-formatted file to your working directory.

Examples

```
## Not run:
data <- querySanger(
    locus = 'CYB', projects=list(1),
    expeditions=list("acaach_CyB_JD", "acajap_CyB_JD"),
    query="yearCollected >= 2008")

data <- querySanger(locus = 'CO1', query = "genus = Linckia AND specificEpithet = laevigata" )

## End(Not run)
```

Index