

dbgap2x

July 25, 2019

1 Using dbgap2x, R package to explore, download and decrypt phenotypic and genomic data from dbGaP

1.1 Introduction

1.1.1 Load the package

```
In [1]: #devtools::install_github("guersmee/dbgap2x", force = TRUE)
library(dbgap2x)
```

1.1.2 Get the list of the function for this new package

```
In [2]: lsf.str("package:dbgap2x")

browse.dbgap : function (phs, no.browser = FALSE)
browse.study : function (phs, no.browser = FALSE)
consent.groups : function (phs)
datatables.dict : function (phs)
dbgap.data_dict : function (xml, dest)
dbgap.decrypt : function (files, key = FALSE)
dbgap.download : function (krt, key = FALSE)
is.parent : function (phs)
n.pop : function (phs, consentgroups = TRUE, gender = TRUE)
n.tables : function (phs)
n.variables : function (...)
parent.study : function (phs)
phs.version : function (phs)
search.dbgap : function (term, no.browser = FALSE)
study.name : function (phs)
sub.study : function (phs)
variables.dict : function (phs)
```

1.2 Search for dbGaP studies

1.2.1 Let's try to explore the "Jackson Heart Study" cohort that exists on dbGaP.

We created the function "browse.dbgap", which helps you to find the studies related to the term that you search for

```
In [3]: search.dbgap("Jackson")
```

```
https://www.ncbi.nlm.nih.gov/gap/?term=Jackson%5BStudy+Name%5D
```

Study ID	Study Name	Release Date
phs001356.v1.p2	Exome Chip Genotyping: The Jackson Heart Study	2019-05-10
phs001098.v2.p2	T2D-GENES Multi-Ethnic Exome Sequencing Study: Jackson Heart Study	2019-05-10
phs000499.v4.p2	NHLBI Jackson Heart Study Candidate Gene Association Resource (CARE)	2019-05-10
phs000498.v4.p2	Jackson Heart Study Allelic Spectrum Project	2019-05-10
phs000286.v6.p2	The Jackson Heart Study (JHS)	2019-05-10
phs000964.v3.p1	NHLBI TOPMed: The Jackson Heart Study	2018-05-18

dbGaP returns the list of the studies related to your term. As you see, there are 6 studies associated with the “Jackson Heart Study” (JHS). One of these study is the main one a.k.a the “parent study”, whereas the other ones are substudies. In this case, phs000286.v5.p1 is the parent study. Firstly, we can use the `phs.version()` function in order to be sure that this is the latest version of the study. We can abbreviate the phs name by giving just the digit, or we can use the full dbGaP id.

```
In [4]: phs.version("286")
```

```
'phs000286.v6.p2'
```

The `is.parent()` function is useful to test if a study is a parent study or a substudy

```
In [5]: is.parent("000286") # JHS main cohort
        is.parent("phs499") # substudy "CARE" for JHS
```

```
TRUE
```

```
FALSE
```

If you don’t know the parent study of a substudy, try `parent.study()`

```
In [6]: parent.study("phs000499")
```

```
1. 'phs000286.v6.p2' 2. 'Jackson Heart Study (JHS) Cohort'
```

On the other side, use `sub.study()` to get the name and IDs of the substudies from a parent one

```
In [7]: sub.study("286")
```

phs	name
phs001356.v1.p2	Exome Chip Genotyping: The Jackson Heart Study
phs000498.v4.p2	Jackson Heart Study Allelic Spectrum Project
phs001069.v1.p2	MIGen_ExS: JHS
phs000402.v4.p2	NHLBI GO-ESP: Heart Cohorts Exome Sequencing Project (JHS)
phs000499.v4.p2	NHLBI Jackson Heart Study Candidate Gene Association Resource (CARE)
phs001098.v2.p2	T2D-GENES Multi-Ethnic Exome Sequencing Study: Jackson Heart Study

If you want to get the name of a study from its dbGaP id, use `study.name()`

```
In [8]: study.name("286")
'Jackson Heart Study (JHS) Cohort'
```

Finally, you can watch your study on dbGaP with `browse.dbgap()`.

If a website exists for this study, you can browse it using `browse.study()`

```
In [9]: browse.dbgap("286", no.browser = TRUE)
        browse.study("286", no.browser = TRUE)

'https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs000286.v6.p2'
'https://www.jacksonheartstudy.org'
```

1.3 Explore the characteristics of your study

For each dbGaP study, there can be multiple consent groups that will have their specificities. Use `consent.groups` to know the number and the name of the consent groups in the study that you are exploring. Let's keep focusing on JHS.

```
In [10]: JHS <- "phs000286"
         consent.groups(JHS)
```

	shortName	longName
0	NRUP	Subjects did not participate in the study, did not complete a consent document and
1	HMB-IRB-NPU	Health/Medical/Biomedical (IRB, NPU)
2	DS-FDO-IRB-NPU	Disease-Specific (Focused Disease Only, IRB, NPU)
3	HMB-IRB	Health/Medical/Biomedical (IRB)
4	DS-FDO-IRB	Disease-Specific (Focused Disease Only, IRB)

Use `n.pop()` to know the number of patient included in each groups

```
In [11]: n.pop(JHS)
         n.pop(JHS, consentgroups = FALSE)
```

consent_group	male	female	total
HMB-IRB	2409	3046	5885
HMB-IRB-NPU	265	511	883
DS-FDO-IRB-NPU	63	109	201
HMB-IRB	793	1249	2289
DS-FDO-IRB	174	295	516
TOTAL	3704	5210	9774

9774

Use `n.tables()` and `n.variables()` to get the number of datatables in your study and the total number of variables

```
In [12]: n.tables(JHS)
         n.variables(JHS)

112
4856
```

`datatables.dict()` will return a data frame with the datatables IDs (phtxxxxxx) and description of your study

```
In [13]: tablesdict <- datatables.dict(JHS)
        head(tablesdict)
```

pht	dt_study_name	dt_label
pht008811.v1	MIGen_JHS_AA_Subject_Phenotypes	Subject ID, age, sex, cohort, consortium, T2D affecti
pht008783.v1	sbpc	sbpc
pht008727.v1	allevthf	allevthf
pht001959.v2	loca	loca
pht001945.v2	cena	cena
pht001957.v2	hcaa	hcaa

`variables.dict()` will return a data frame with the variables IDs (phvxxxxxx), their name in the study, the datatable where they come from and their description

```
In [14]: vardict <- variables.dict(JHS)
        head(vardict)
```

dt_study_name	phv	var_name	var_desc
MIGen_JHS_AA_Subject_Phenotypes	phv00404354.v1	SUBJECT_ID	De-identified Subject ID
MIGen_JHS_AA_Subject_Phenotypes	phv00404355.v1	sex	Gender of participant
sbpc	phv00403830.v1	SUBJECT_ID	PARTICIPANT ID [Visit 1] [Sittin
sbpc	phv00403831.v1	VISIT	CONTACT OCCASION [Visit 1]
sbpc	phv00403832.v1	SBPC1	Q1. A. Temperature. Room temp
sbpc	phv00403833.v1	SBPC2	Q2. B. Tobacco and caffeine use, p

1.4 Extract your study

1.4.1 Get your dbGaP repository key

In order to download or decrypt your data from dbGaP, you will need to request an access and to get a decryption key. Follow those steps to access your dbGaP repository key: ##### - Go to <https://www.ncbi.nlm.nih.gov/gap> and click on controlled access data

- Click on Log in to dbGaP

- Identify yourself with your era common ID and password

- **Get a PI dbGaP repository key:** In order to download the files and to decrypt them, you will need a decryption key. This key can be found on a PI dbGaP account. Go to the Authorized Access and then My Projects tabs. Then, in the column Actions on the right of your screen, find Get no password dbGaP repository key.

1.4.2 Decrypt the .ncbi_enc files

On dbGaP, the phenotypic files are encrypted. We created a decryption function that uses a dockerized version on sratoolkit. To use that function, you need to have docker installed on your

device (www.docker.com). If you are using the dockerized version of this software (available at hub.docker.com/r/gversmee/dbgap2x), docker is already pre-installed, but you'll need to upload your key on the jupyter working directory.

```
In [ ]: key <- "path/to/your/key.ngc"
        files <- "path/to/directory/ofencrypted_files"
        dbgap.decrypt(files, key)
```

You should see a "decrypted_files" directory in the directory where your encrypted files are located

1.4.3 Download dbGaP files

- **Click on "file selector"** This gives you access to the dbGaP file selector where you can find all the files available for the selected project. To find it, go to the Authorized Access and then My Projects tabs. Then, in the column Actions on the right of your screen, find file selector.

- **Filter by study accession** Here, we want to get the phenotypic data for the study "Early onset COPD", so after checking Study accession, we select "phs000946".

- **Filter again** Since we are only interested in getting the phenotypic data, let's filter by Content type and select phenotype individual-auxiliary and phenotype individual-traits.

- **Select the files** Click on "+" to select all the files.

- **Click on "Cart file"** This will download a .krt file in your download folder.

1.4.4 Download and decrypt the files

```
In [ ]: key <- "path/to/your/key.ngc"
        cart <- "path/to/your/cartfile.krt"
        dbgap.download(cart, key)
```

You should see in your working directory a new folder named dbGaP-*** that contains your files