

dbGaP2x

October 19, 2018

1 Using dbGaP2x, R package to explore and sort phenotypics 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, jupyter = FALSE)
browse.study : function (phs, jupyter = 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 (phs)
parent.study : function (phs)
phs.version : function (phs)
search.dbgap : function (term, jupyter = 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.

The dbGap search engine can be tricky, that's why we created the function "browse.dbgap", who helps you find the studies related to the term that you search on your web browser.

Note that if you run this function in a jupyterhub environment, it will return a url since jupyterhub doesn't have access to your local browser.

```
In [3]: search.dbgap("Jackson", jupyter = TRUE)
```

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

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 aka 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.v5.p1'
```

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.v5.p1' 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
phs000499.v3.p1	NHLBI Jackson Heart Study Candidate Gene Association Resource (CARE)
phs000498.v3.p1	Jackson Heart Study Allelic Spectrum Project
phs000402.v3.p1	NHLBI GO-ESP: Heart Cohorts Exome Sequencing Project (JHS)
phs001098.v1.p1	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", jupyter = TRUE)
        browse.study("286", jupyter = TRUE)

'https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs000286.v5.p1'
'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 there 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	1860	2504	4549
HMB-IRB-NPU	264	505	802
DS-FDO-IRB-NPU	63	107	180
HMB-IRB	784	1232	2131
DS-FDO-IRB	173	289	489
TOTAL	3144	4637	8151

8151

Use `n.tables()` and `n.variables()` to get the number of datatables in your study and the total number of variables (n.variables goes into the study files to count the actual number of variables)

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

66
4326

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
pht002539.v2	ESP_HeartGO_JHS_Subject_Phenotypes	Subject ID, ESP cohort, target capture used in seq
pht001948.v1	CSTA	Agatston score of all coronary section among par
pht001947.v1	CSIA	Approach to life B. Life style among participants
pht001968.v1	PPAA	Post physical activity monitoring among particip
pht001955.v1	ECHA	Echocardiographic abnormalities among particip
pht001952.v1	DPASS_DIET1	Dietary data (DPASS) among participants of the J

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
ESP_HeartGO_JHS_Subject_Phenotypes	phv00165323.v2	SUBJID	Subject ID
ESP_HeartGO_JHS_Subject_Phenotypes	phv00165322.v2	ESP_Cohort	Cohort name [JHS]
ESP_HeartGO_JHS_Subject_Phenotypes	phv00165324.v2	ESP_phenotype	ESP Phenotype group (p
ESP_HeartGO_JHS_Subject_Phenotypes	phv00181282.v1	Sequence_center	Indicates where the sam
ESP_HeartGO_JHS_Subject_Phenotypes	phv00181283.v1	Target	Indicates target capture
ESP_HeartGO_JHS_Subject_Phenotypes	phv00181284.v1	ESP_race_selfreport	Self report race [African

Now that we have explore our datasets, let's use sandboxR in order to clean our variables, and to gather them into a tree that will be easier to use for researchers. Note that for chapter 3, we will need to move and create a lot of files on your environment. It will be easier to use on your local computer than in the Jupyterhub environment.

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:

a. Go to <https://www.ncbi.nlm.nih.gov/gap> and click on “controlled access data”

The screenshot shows the dbGaP homepage. At the top, there's a navigation bar with 'dbGaP' and a search bar. Below this, a large banner features an eye graphic and the text: 'The database of Genotypes and Phenotypes (dbGaP) was developed to archive and distribute the data and results from studies that have investigated the interaction of genotype and phenotype in Humans.' Below the banner, there are three main sections: 'Access dbGaP Data', 'Resources', and 'Important Links'. In the 'Access dbGaP Data' section, the 'Controlled Access Data' link is highlighted with a red box. Other links in this section include 'Advanced Search', 'Public FTP Download', 'Collections', and 'Summary Statistics'. The 'Resources' section includes 'dbGaP Data Browser', 'Phenotype-Genotype Integrator', 'dbGaP RSS Feed', 'Software', and 'dbGaP Tutorial'. The 'Important Links' section includes 'How to Submit', 'FAQ', 'Code of Conduct', 'Security Procedures', and 'Contact Us'. Below these sections, there's a 'Latest Studies' section with a table of studies.

Study	Embargo Release	Details	Participants	Type Of Study	Links	Platform
phs001228.v1.p1 Kids First Pediatric Research Program in Susceptibility to Ewing Sarcoma Based on Germline Risk and Familial History of Cancer	Version 1: passed embargo	V D A S	1112	Parent-Offspring Trios	Links	HiSeq X
phs000220.v2.p2	Versions 1-2: passed embargo	V D A S	27995	Cohort, Case-Control		TagMan OpenArrays: MEC_HI_SNPASSAYONLY_04-14-10 TagMan OpenArrays: MEC_HI_SNPASSAYONLY_09-18-11 TagMan OpenArrays: MEC_HI_SNPASSAYONLY_12-7-09 TagMan OpenArrays: MEC_LA_SNPASSAYONLY_12-7-09 TagMan OpenArrays: MEC_HI_SNPASSAYONLY_09-15-11 (Select Custom Panel)
phs001326.v1.p1 A Retrospective and Cross-Sectional Analysis of Patients Treated for SCID Since January 1, 1969	Version 1: passed embargo	V D A S	748	Longitudinal, Observational, Cross-Sectional	Links	
phs001587.v1.p1	Version 1: passed	V D A S				HiSeq

b. Click on Log in to dbGaP

The screenshot shows the 'Authorized Access Portal' of dbGaP. At the top, there's a navigation bar with 'dbGaP' and links for 'Browse/Search', 'Authorized Access', and 'Help'. Below this, there's a section titled 'Authorized Access Portal' with a 'Log In to dbGaP' link highlighted by a red box. To the right of this link, there's a box titled 'dbGaP Data Browser – View Only' which contains information about the simplified controlled-access application and the purpose of the dbGaP Data Browser. Below the 'Log In to dbGaP' link, there's a box titled 'dbGaP Data Download' which contains information about the management portal to request and download individual level data. At the bottom of the page, there's a footer with the NIH logo and the text 'FIRSTGov.gov'.

dbGaP Data Download

The management portal to request and download individual level data

Click [here](#) to login to the dbGaP controlled-access portal and to begin a project request. For guidance on the development of a data access request to complete project requests, please see [Tips for preparing a successful Data Access Request](#).

Who can apply for access?

How does one apply?

Why is Access Controlled?

[Additional help.](#)

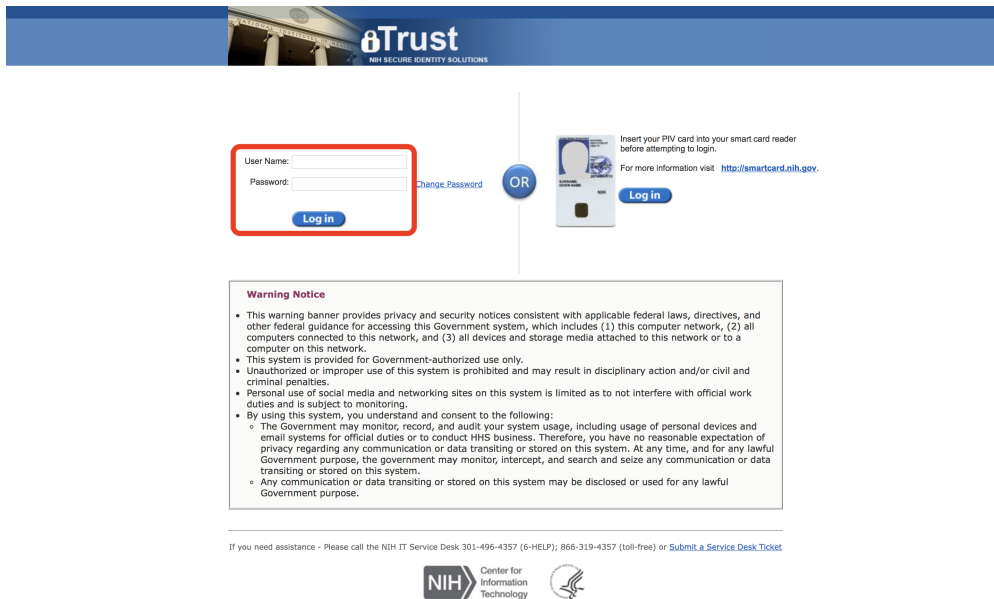
dbGaP Data Browser – View Only

With dbGaP Data Browser approval through the [simplified controlled-access application](#), users may view the collection “[Compilation of individual-level data from general research use \(GRU\)](#).”

What is the purpose of the dbGaP Data Browser; why is it useful?

How does one apply?

c. Identify yourself with your era common ID and password



d. 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, under **Get no password dbGaP repository key**

#	Project	Actions
7582	Increasing the power for G x E by using multi-locus predictors Chirag Patel	run selector file selector launch browser get embargo report get no password dbGaP repository key
10087	Development and evaluation of methods to estimate genetic and environmental influence in twins Chirag Patel	run selector file selector launch browser get embargo report get no password dbGaP repository key
10827	Increasing the power for G x E by using multi-locus predictors Chirag Patel	run selector file selector launch browser get embargo report get no password dbGaP repository key
13925	Testing the Reproducibility of the Associations between Rare Genetic Variations and Neuropsychiatric Diseases in Other Populations Paul Avillach	run selector file selector launch browser get embargo report get no password dbGaP repository key
16589	Large-scale meta-analysis of the human microbiome Chirag Patel	run selector file selector launch browser get embargo report get no password dbGaP repository key
16901	PIC-FAIR - KCB - Characterizing genetic variants in hypertrophic cardiomyopathy in diverse populations Paul Avillach	images selector run selector file selector launch browser get embargo report get no password dbGaP repository key
17011	NIH Data Commons Pilot Phase - Infrastructure and Methods Development Paul Avillach	images selector run selector file selector launch browser get embargo report 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. To try the function, we put some pre-encrypted files on the repo

```
In [15]: key <- "prj_17011.ngc"
files <- "encrypted_files"
dbgap.decrypt(files, key)
```

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

1.4.3 3.1. Download dbGaP files

a. 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.

The screenshot shows the dbGaP website interface. At the top, there's a navigation bar with links like 'Home', 'About', 'Help', 'Log out', etc. Below that, the 'My Research Projects' section is displayed. It lists several projects with their IDs and titles. For each project, there are links for 'run selector', 'file selector', 'launch browser', 'get embargo report', and 'get no password dbGaP repository key'. Project 17011, 'NIH Data Commons Pilot Phase - Infrastructure and Methods Development', is highlighted with a red box around its 'File selector' link.

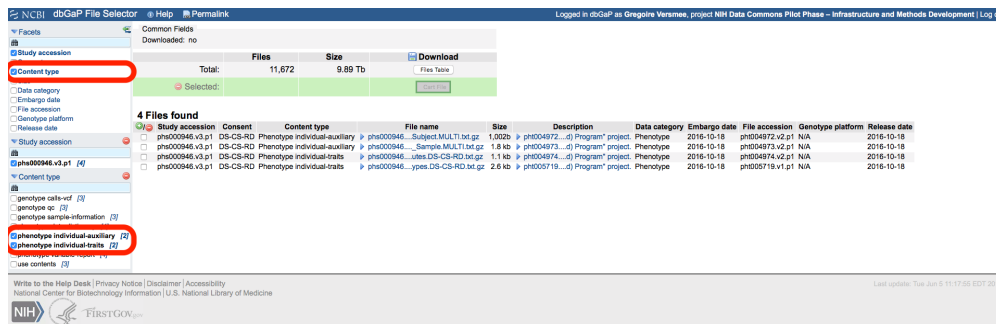
b. Filter by study accession

Here, we want to get the phenotypic data for the study “Early onset COPD”, so after checking Study accession, we select “phs000946”.

The screenshot shows the 'dbGaP File Selector' interface. On the left, there's a sidebar with filters like 'Study accession', 'Content type', 'Data category', etc. The 'Study accession' filter is selected, and a list of study accessions is shown. 'phs000946.v2.p1' is highlighted. The main area shows a table of files with columns for 'Study accession', 'Consent', 'Content type', 'File name', 'Size', 'Description', 'Data category', 'Embargo date', 'File accession', 'Genotype platform', and 'Release date'. The table lists various files associated with the selected study accession.

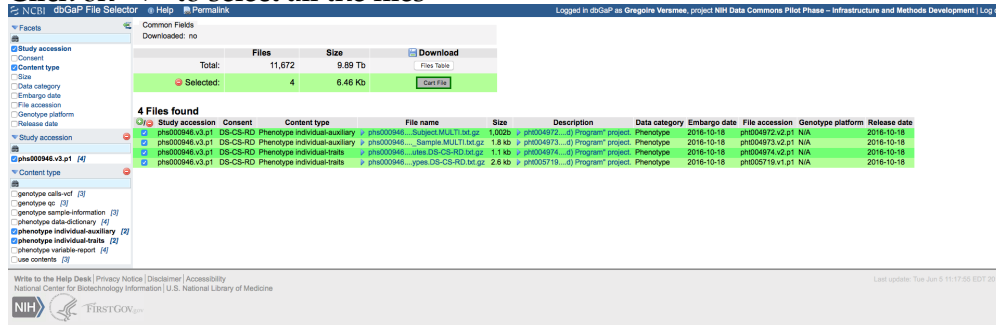
c. Filter again

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



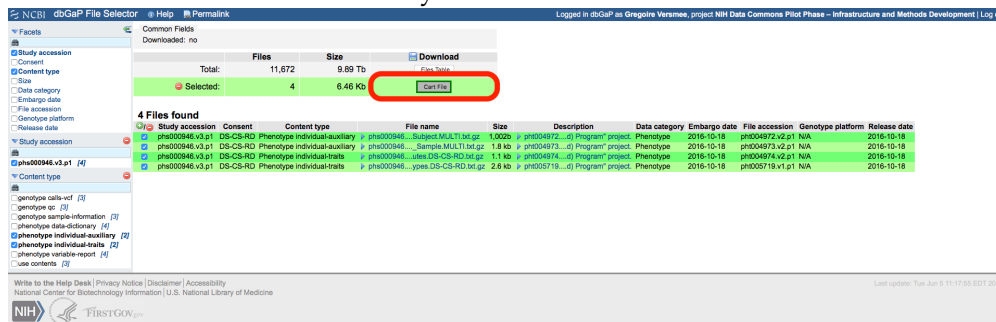
d. Select the files

Click on “+” to select all the files



e. Click on “Cart file”

This will download a .krt file in your download folder



f. Download and decrypt the files with a simple command

```
In [16]: key <- "prj_17011.ngc"
        cart <- "cart_prj17011_201810151143.krt"
        dbgap.download(cart, key)
```

You should see in your working directory a new one name dbGaP-*** that contains your files