

DataSpaceR

is an API for the CAVD DataSpace, written for the R statistical computing environment. The CAVD DataSpace (DataSpace) is a data sharing and discovery tool created for the HIV vaccine research community.

DataSpaceR provides a way for programmers to explore and access data within the CAVD DataSpace entirely through an R environment. DataSpace requires an account for accessing data from the API. Accounts can be set up for free by going to the CAVD DataSpace website and registering: <https://dataspace.cavd.org/>.



DataSpaceR

Getting started

Set up an account

New DataSpace users need to set up an account before using DataSpaceR.

- Go to <https://dataspace.cavd.org>
- Click "Register Here"
- Follow the on screen instructions

Install and load DataSpaceR

Download and load DataSpaceR from CRAN and load the library into an R session. Use the following commands.

```
install.packages("DataSpaceR")
library(DataSpaceR)
```

Set up credentials

Once loaded, DataSpaceR can help set up the credentials. DataSpaceR requires a netrc file to access your credentials from registering the DataSpace account. The following code will set up that file for you.

Replace "email@address.com" and "password" with the email address and password for the registered account.

```
writeNetrc(
  login = "email@address.com",
  password = "password",
  netrcFile = getNetrcPath()
)
```

See vignette ("DataSpaceR") for more details, or contact DataSpace support: dataspace.support@fredhutch.org

DataSpaceR - Main Workflows

```
con <- connectDS()
```

Creating a DataSpaceConnection object is the first step to using DataSpaceR. Once instantiated, data can be extracted using the 4 main pathways shown below.

Study datasets

Study datasets included integrated and non-integrated immunoassays such as BAMA, ICS, NAb, ADCC, ADPC, and PK mAb.

View all available studies

```
con$availableStudies
```

Create study object

```
stu <- con$getStudy( study_name )
```

Get dataset from study by name

```
stu$getDataset( dataset_name )
```

Curated and saved group datasets

DataSpace can store groups saved in the web application which can be accessed with R.

View all available groups

```
con$availableGroups
```

Create study object with get_group

```
grp <- con$getGroup( group_id )
```

Get dataset from by dataset name

```
grp$getDataset( dataset_name )
```

NAb mAb datasets

Neutralizing Antibody Monoclonal Antibody data uses different methods for access. See "Find NAb mAb data" for details.

View mAb grid

```
con$mabGrid
```

Filter mAb grid

```
con$filterMabGrid( mab_grid_column, value )
```

Create mAb object from filtered grid, then print object

```
mab <- con$getMab()
```

Extract NAb mAb data from mAb object

```
mab$mabMab
```

Publication datasets

DataSpace stores datasets used in certain publications. Those data can be accessed here as well.

View all available publication records

```
con$availablePublications
```

Download dataset for publication with available data

```
con$downloadPublicationData( publication_id )
```

Look for available data

```
con <- connectDS()
```

← The connection object provides tables for viewing available data.

```
con$availableStudies
con$availableGroups
con$mabGrid
con$mabGridSummary
con$availablePublications
```

These tables allow users to explore available data. The data printed to the screen from these commands can be used for finding valid inputs to create DataSpaceStudy or DataSpaceMab objects. These tables can also be queried just as one would query any data.frame type object and can be used directly as arguments for extracting data.

Find NAb mAb data

Print the mabGrid to the screen

```
con$mabGrid
```

Print the available columns to filter the mabGrid with

```
names(con$mabGrid)
```

Examples of how to find filter terms in the mabGrid

```
# print all clades found in grid
unique(con$mabGrid$clade)

# print all mab_mixture names containing "PGT"
con$mabGrid[grep("PGT", mab_mixture),]

# print mabGrid showing records with titer IC 50
# values < 0.01 ug/ml
con$mabGrid[curve_ic50 < 0.01,]
```

Examples of filtering the mabGrid. Resetting the grid is required between filtering. See last step of this section.

```
# filter for all mabs and clade B viruses
con$filterMabGrid("clade", "B")

# filter for all mabs with PGT in their names
con$filterMabGrid(
  "mab_mixture",
  con$mabGrid[
    grep("PGT", mab_mixture), mab_mixture
  ]
)

# filter for all mab virus combinations with titer
# IC50 values less than 0.01 ug/ml
con$filterMabGrid(
  "curve_ic50",
  con$mabGrid[curve_ic50 < 0.01, curve_ic50]
)
```

Extract data based on the state of the mabGrid

```
mab <- con$getMab()$mabMab
```

Reset the mabGrid to make a new filter

```
con$resetMabGrid()
```

Methods reference tables

DataSpaceConnection	
Created with <code>connectDS()</code>	
Method	Description
<code>availableGroups</code>	The table of saved and curated groups.
<code>availablePublications</code>	The table of tracked publications.
<code>availableStudies</code>	The table of tracked studies.
<code>downloadPublicationData</code>	Download publication data locally.
<code>getGroup</code>	Get <code>DataSpaceGroup</code> object.
<code>getMab</code>	Get <code>DataSpaceMab</code> object.
<code>getStudy</code>	Get <code>DataSpaceStudy</code> object.
<code>filterMabGrid</code>	Filter the <code>mabGrid</code> table.
<code>mabGrid</code>	The table of the current state of the mAb grid.
<code>mabGridSummary</code>	The summary table of the current state of the mAb grid.
<code>resetMabGrid</code>	Reset the state of the mAb grid.
<code>virusMetadata</code>	The table of virus metadata.

DataSpaceStudy	
Created from a <code>DataSpaceConnection</code> object with <code>getStudy()</code> or <code>getGroup()</code>	
Method	Description
<code>study</code>	The study name.
<code>group</code>	The group name.
<code>availableDatasets</code>	The table of datasets available in the study object.
<code>treatmentArm</code>	The table of treatment arm information for the connected study.
<code>dataDir</code>	The default target directory for downloading non-integrated datasets.
<code>studyInfo</code>	Store the information about the study.
<code>getDataset</code>	Get a dataset from the connection.
<code>getDatasetDescription</code>	Get variable information.
<code>setDataDir</code>	Set default target directory for downloading non-integrated datasets.

DataSpaceMab	
Created from a <code>DataSpaceConnection</code> object with <code>getMab()</code> after filtering with <code>filterMabGrid()</code>	
Method	Description
<code>studyAndMabs</code>	The table of available mAbs by study.
<code>mabs</code>	The table of available mAbs and their attributes.
<code>nabMab</code>	The table of NAb mAb assay results.
<code>studies</code>	The table of available studies.
<code>assays</code>	The table of assay status by study.
<code>variableDefinitions</code>	The table of variable definitions.

Explore DataSpaceR objects

```
con <- connectDS()
con
```

Create the connection object, assigned to the symbol **con** in this example. Typing **con** in the console will print the object. See methods reference table for available methods.

```
<DataSpaceConnection>
URL: https://dataspace.cavd.org
User: jmtaylor@fredhutch.org
Available studies: 365
- 78 studies with data
- 5073 subjects
- 439831 data points
Available groups: 3
Available publications: 1530
- 12 publications with data
```

This is a summary of what data is available to the user in DataSpace. Data is extracted via one of the 4 workflows outlined in this cheat sheet.

DataSpaceConnection

```
con <- connectDS()
stu <- con$getStudy("vtn505")
stu
```

Create the study object, assigned to the symbol **stu** in this example. Typing **stu** in the console will print the object. See methods table for available methods.

```
<DataSpaceStudy>
Study: vtn505
URL: https://dataspace.cavd.org/CAVD/vtn505
Available datasets:
- Binding Ab multiplex assay
- Demographics
- Intracellular Cytokine Staining
- Neutralizing antibody
Available non-integrated datasets:
- ADCP
- Demographics (Supplemental)
- Fc Array
```

The printed object shows basic information about the study

These dataset names are used to extract datasets using the **getDataset** method. This example shows integrated and non-integrated data.

DataSpaceStudy

```
con <- connectDS()
con$filterMabGrid(
  "mab_mixture", c("PGT125", "PGT128")
)
mab <- con$getMab()
mab
```

Create the study object, assigned to the symbol **mab** in this example. Typing **mab** in the console will print the object. See methods table for all the available methods.

```
<DataSpaceMab>
URL: https://dataspace.cavd.org
User: jmtaylor@fredhutch.org
Summary:
- 2 studies
- 2 mAb mixtures
- 7 neutralization tiers
- 13 clades
Filters:
- mab_mix_name_std: PGT125, PGT128
```

The printed object shows basic information about the mAb object

This is a summary of what mAb data is available in the mAb object. Neutralizing antibody data is extracted via the **nabmab** method.

DataSpaceMab