
cmapBQ

Release 1.0.0

Anup Jonchhe

Dec 17, 2020

CONTENTS

1	Installation Guide	1
2	Guide	3
2.1	cmapBQ	3
2.2	Using CMap's BQ Toolkit	8
2.3	Instructions for installing cmapBQ	9
2.4	Where to place your JSON service file	9
2.5	Credentials Setup	9
2.6	Need Help?	11
2.7	License	11
3	Indices and tables	13
	Python Module Index	15
	Index	17

INSTALLATION GUIDE

The **cmapBQ** toolkit is available on PyPi and can be installed using: `pip install cmapBQ`

This will install the package into your currently active environment.

From here, ensure you have a Google Service Account Credentials file, documentation can be found at [Getting started with authentication](#)

It is recommended to place the credentials in the `~/.cmapBQ` folder. To complete installation, run the following command from within a python session. This only needs to be done once, and will populate a `~/.cmapBQ/config.txt` file with default table values and correct credentials path.

```
import cmapBQ.query as cmap_query
import cmapBQ.config as cmap_config

cmap_config.setup_credentials(path_to_json)
```


- search

2.1 cmapBQ

2.1.1 cmapBQ.config module

class `cmapBQ.config.Configuration` (*credentials: str, tables: cmapBQ.config.TableDirectory*)

Data class for configuration of cmapBQ. Object for config.txt

credentials: `str`

tables: `cmapBQ.config.TableDirectory`

class `cmapBQ.config.TableDirectory` (*compoundinfo: str, genetic_pertinfo: str, geneinfo: str, cellinfo: str, instinfo: str, siginfo: str, level3: str, level4: str, level5: str*)

cellinfo: `str`

compoundinfo: `str`

geneinfo: `str`

genetic_pertinfo: `str`

instinfo: `str`

level3: `str`

level4: `str`

level5: `str`

siginfo: `str`

`cmapBQ.config.get_bq_client` (*config=None*)

Return authenticated BigQuery client object.

Parameters `config` – optional path to config if not default

Returns BigQuery Client

`cmapBQ.config.get_default_config` ()

Get configuration object from reading `~/cmapBQ/config.txt`

Returns `cmapBQ.config.Configuration` class.

`cmapBQ.config.set_default_config(input_config_path)`

Change configuration in `~/cmapBQ` to input config path. Overwrites `~/cmapBQ/config.txt`,

Parameters `input_config_path` – valid YAML formatted config file

Returns location in `~/cmapBQ`

`cmapBQ.config.setup_credentials(path_to_credentials)`

Setup script for pointing `config.txt` to a `GOOGLE_APPLICATION_CREDENTIALS` JSON key. Writes default tables if `~/cmapBQ/config.txt` does not exist.

Parameters `path_to_credentials` –

Returns None (side effect)

2.1.2 cmapBQ.query module

`cmapBQ.query.cmap_cell(client, cell_iname=None, cell_alias=None, ccle_name=None, primary_disease=None, cell_lineage=None, cell_type=None, table=None, verbose=False)`

Query cellinfo table

Parameters

- **client** – Bigquery Client
- **cell_iname** – List of cell_inames
- **cell_alias** – List of cell aliases
- **ccle_name** – List of ccle_names
- **primary_disease** – List of primary_diseases
- **cell_lineage** – List of cell_lineages
- **cell_type** – List of cell_types
- **table** – table to query. This by default points to the `siginfo` table and normally should not be changed.
- **verbose** – Print query and table address.

Returns Pandas DataFrame

`cmapBQ.query.cmap_compounds(client, pert_id=None, cmap_name=None, moa=None, target=None, compound_aliases=None, limit=None, verbose=False)`

Query compoundinfo table for various field by providing lists of compounds, moa, targets, etc. ‘AND’ operator used for multiple conditions.

Parameters

- **client** – BigQuery Client
- **pert_id** – List of pert_ids
- **cmap_name** – List of cmap_names
- **target** – List of targets
- **moa** – List of MoAs
- **compound_aliases** – List of compound aliases
- **limit** – Maximum number of rows to return

- **verbose** – Print query and table address.

Returns Pandas Dataframe matching queries

```
cmapBQ.query.cmap_genes (client, gene_id=None, gene_symbol=None, ensembl_id=None,
                        gene_title=None, gene_type=None, feature_space='aig', src=None,
                        table=None, verbose=False)
```

Query geneinfo table. Geneinfo contains information about genes including ids, symbols, types, ensembl_ids, etc.

Parameters

- **client** – Bigquery Client
- **gene_id** – list of gene_ids
- **gene_symbol** – list of gene_symbols
- **ensembl_id** – list of ensembl_ids
- **gene_title** – list of gene_titles
- **gene_type** – list of gene_types
- **feature_space** – Common featurespaces to extract. ‘rid’ overrides selection
Choices: ['landmark', 'bing', 'aig']
landmark: 978 landmark genes
bing: Best-inferred set of 10,174 genes
aig: All inferred genes including 12,328 genes
Default is aig.
- **src** – list of gene sources
- **table** – table to query. This by default points to the siginfo table and normally should not be changed.
- **verbose** – Print query and table address.

Returns Pandas DataFrame

```
cmapBQ.query.cmap_genetic_perts (client, pert_id=None, cmap_name=None, gene_id=None,
                                gene_title=None, ensemble_id=None, table=None,
                                verbose=False)
```

Query genetic_pertinfo table

Parameters

- **client** – Bigquery Client
- **pert_id** – List of pert_ids
- **cmap_name** – List of cmap_names
- **gene_id** – List of type INTEGER corresponding to gene_ids
- **gene_title** – List of gene_titles
- **ensemble_id** – List of ensumble_ids
- **table** – table to query. This by default points to the siginfo table and normally should not be changed.
- **verbose** – Print query and table address.

Returns

`cmapBQ.query.cmap_matrix` (*client*, *data_level='level5'*, *feature_space='landmark'*, *rid=None*,
cid=None, *verbose=False*, *chunk_size=1000*, *table=None*, *limit=4000*)
 Query for numerical data for signature-gene level data.

Parameters

- **client** – Bigquery Client
- **data_level** – Data level requested. IDs from siginfo file correspond to ‘level5’. Ids from instinfo are available in ‘level3’ and ‘level4’. Choices are [‘level5’, ‘level4’, ‘level3’]
- **rid** – Row ids
- **cid** – Column ids
- **feature_space** – Common featurespaces to extract. ‘rid’ overrides selection
 Choices: [‘landmark’, ‘bing’, ‘aig’]
 landmark: 978 landmark genes
 bing: Best-inferred set of 10,174 genes
 aig: All inferred genes including 12,328 genes
 Default is landmark.
- **chunk_size** – Runs queries in stages to avoid query character limit. Default 1,000
- **limit** – Soft limit for number of signatures allowed. Default is 4,000.
- **table** – Table address to query. Overrides ‘data_level’ parameter. Generally should not be used.
- **verbose** – Print query and table address.

Returns GCToo object

`cmapBQ.query.cmap_profiles` (*client*, *sample_id=None*, *pert_id=None*, *pert_type=None*,
cmap_name=None, *cell_iname=None*, *det_plate=None*,
build_name=None, *return_fields='priority'*, *limit=None*, *table=None*,
verbose=False)

Query per sample metadata, corresponds to level 3 and level 4 data, AND operator used for multiple conditions.

Parameters

- **client** – Bigquery client
- **sample_id** – list of sample_ids
- **pert_id** – list of pert_ids
- **pert_type** – list of pert_types. Avoid using only this parameter as the return could be very large.
- **cmap_name** – list of cmap_names
- **det_plate** – list of det_plates
- **build_name** – list of builds
- **return_fields** – [‘priority’, ‘all’]
- **limit** – Maximum number of rows to return
- **table** – table to query. This by default points to the siginfo table and normally should not be changed.

- **verbose** – Print query and table address.

Returns Pandas Dataframe

`cmapBQ.query.cmap_sig` (*client, sig_id=None, pert_id=None, pert_type=None, cmap_name=None, cell_iname=None, det_plates=None, build_name=None, return_fields='priority', limit=None, table=None, verbose=False*)

Query level 5 metadata table. Multiple parameters are filtered using the 'AND' operator

Parameters

- **client** – Bigquery Client
- **sig_id** – list of sig_ids
- **pert_id** – list of pert_ids
- **pert_type** – list of pert_types. Avoid using only this parameter as the return could be very large.
- **cmap_name** – list of cmap_name, formerly pert_iname
- **cell_iname** – list of cell names
- **det_plates** – list of det_plates. det_plates values are the concatenation of values from

instinfo det_plate field with the 'l' delimiter used. :param build_name: list of builds :param return_fields: ['priority', 'all'] :param limit: Maximum number of rows to return :param table: table to query. This by default points to the level 5 siginfo table and normally should not be changed. :param verbose: Print query and table address. :return: Pandas Dataframe

`cmapBQ.query.get_bq_client` ()

Return authenticated BigQuery client object.

Parameters **config** – optional path to config if not default

Returns BigQuery Client

`cmapBQ.query.get_table_info` (*client, table_id*)

Query a table address within client's permissions for schema.

Parameters

- **client** – Bigquery Client
- **table_id** – table address as {dataset}.{table_id}

Returns Pandas Dataframe of column names. Note: Not all column names are query-able but all will be returned from a given metadata table

`cmapBQ.query.list_cmap_compounds` (*client*)

List available compounds

Parameters **client** – BigQuery Client

Returns Single column Dataframe of compounds

`cmapBQ.query.list_cmap_moas` (*client*)

List available MoAs

Parameters **client** – BigQuery Client

Returns Single column Dataframe of MoAs

`cmapBQ.query.list_cmap_targets` (*client*)

List available targets

Parameters **client** – BigQuery Client

Returns Pandas DataFrame

`cmapBQ.query.list_tables()`

Print table addresses. Comes from defaults in config.

Returns None

`cmapBQ.query.run_query(client, query)`

Runs BigQuery queryjob

Parameters

- **client** – BigQuery client object
- **query** – Query to run as a string

Returns QueryJob object

2.1.3 cmapBQ.utils module

`cmapBQ.utils.csv_to_gctx(filepaths, outpath, use_gctx=True)`

Convert list of csv files to gctx. CSVs must have 'rid', 'cid' and 'value' columns No other columns or metadata is preserved.

Parameters

- **filepaths** – List of paths to CSVs
- **outpath** – output directory of file
- **use_gctx** – use GCTX HDF5 format. Default is True

Returns

`cmapBQ.utils.long_to_gctx(df)`

Converts long csv table to GCToo Object. Dataframe must have 'rid', 'cid' and 'value' columns No other columns or metadata is preserved.

Parameters **df** – Long form pandas DataFrame

Returns GCToo object

2.2 Using CMap's BQ Toolkit

Introduction The cmapBQ toolkit enables access to data hosted in Bigquery directly from a python session.

[Tutorial Notebook](#) and [Demo Notebook](#) available on Github

2.3 Instructions for installing cmapBQ

The cmapBQ package is available on Pypi and can be installed using the command: `pip install cmapBQ`

2.4 Where to place your JSON service file

The recommended location for service account credentials is within the `~/ .cmapBQ` folder. The following command will populate that folder with a `config.txt` file that points to your credentials file.

```
import cmapBQ.query as cmap_query
import cmapBQ.config as cmap_config

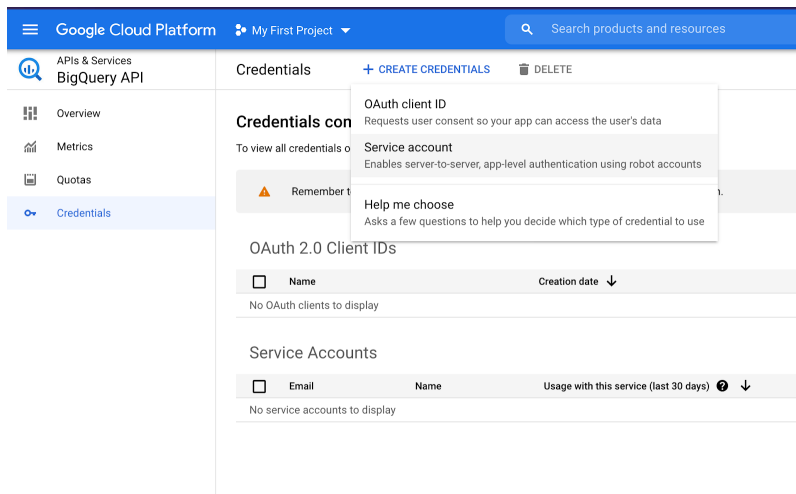
cmap_config.setup_credentials(path_to_json)
```

2.5 Credentials Setup

To be able to access the dataset, register for a Google Cloud account. After registration or if you already have an account, go to your Google Cloud console and then activate your Google Cloud BigQuery API ([link](#)).

When you have access to your Google Cloud Account, go to APIs & Services > Credentials. Find the +Create Credentials and select “Service Account”.

Note: Depending on your organization or project, you may not have access to the credentials page. If that is the case, discuss with the project admin to get your service account key, or create a new project in which you have permission.



Create service account

1 ✓ Service account details

2 Grant this service account access to project (optional)

Grant this service account access to cmapBQ-external-test so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role: BigQuery Job User Condition: Add condition

Access to run jobs

+ ADD ANOTHER ROLE

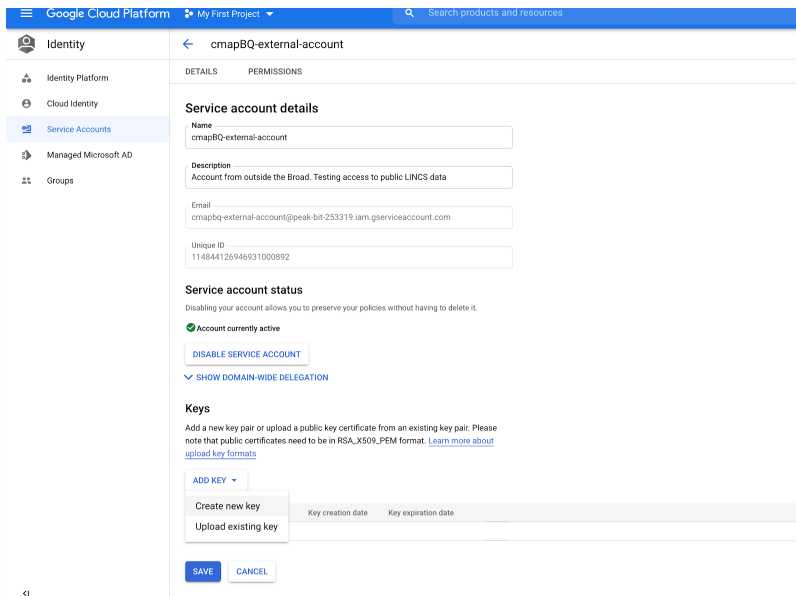
CONTINUE

3 Grant users access to this service account (optional)

DONE CANCEL

Fill out service account details, make sure to set the Role to “BigQuery Job User”

After the service account has been created, find the Section labeled “Keys” and go to Add Key > Create new key. Select “JSON” format.



Place the downloaded JSON file in a safe location, for example, ~/cmapBQ/ and run the following command in python once.

```
import cmapBQ.query as cmap_query
```

(continues on next page)

(continued from previous page)

```
import cmapBQ.config as cmap_config  
cmap_config.setup_credentials(path_to_json)
```

Note: For usage in Colab, JSON key can be uploaded and referenced from the file viewer in the left side menu

2.6 Need Help?

For questions please add an issue to the External hyperlinks, like [Github](#) or email anup@broadinstitute.org

2.7 License

MIT License

Copyright (c) 2020 Connectivity Map at the Broad Institute

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

C

`cmapBQ.config`, 3

`cmapBQ.query`, 4

C

cellinfo (*cmapBQ.config.TableDirectory attribute*), 3
 cmap_cell() (*in module cmapBQ.query*), 4
 cmap_compounds() (*in module cmapBQ.query*), 4
 cmap_genes() (*in module cmapBQ.query*), 5
 cmap_genetic_perts() (*in module cmapBQ.query*), 5
 cmap_matrix() (*in module cmapBQ.query*), 6
 cmap_profiles() (*in module cmapBQ.query*), 6
 cmap_sig() (*in module cmapBQ.query*), 7
 cmapBQ.config
 module, 3
 cmapBQ.query
 module, 4
 cmapBQ.utils
 module, 8
 compoundinfo (*cmapBQ.config.TableDirectory attribute*), 3
 Configuration (*class in cmapBQ.config*), 3
 credentials (*cmapBQ.config.Configuration attribute*), 3
 csv_to_gctx() (*in module cmapBQ.utils*), 8

G

geneinfo (*cmapBQ.config.TableDirectory attribute*), 3
 genetic_pertinfo (*cmapBQ.config.TableDirectory attribute*), 3
 get_bq_client() (*in module cmapBQ.config*), 3
 get_bq_client() (*in module cmapBQ.query*), 7
 get_default_config() (*in module cmapBQ.config*), 3
 get_table_info() (*in module cmapBQ.query*), 7

I

instinfo (*cmapBQ.config.TableDirectory attribute*), 3

L

level13 (*cmapBQ.config.TableDirectory attribute*), 3
 level14 (*cmapBQ.config.TableDirectory attribute*), 3
 level15 (*cmapBQ.config.TableDirectory attribute*), 3
 list_cmap_compounds() (*in module cmapBQ.query*), 7

list_cmap_moas() (*in module cmapBQ.query*), 7
 list_cmap_targets() (*in module cmapBQ.query*), 7
 list_tables() (*in module cmapBQ.query*), 8
 long_to_gctx() (*in module cmapBQ.utils*), 8

M

module
 cmapBQ.config, 3
 cmapBQ.query, 4
 cmapBQ.utils, 8

R

run_query() (*in module cmapBQ.query*), 8

S

set_default_config() (*in module cmapBQ.config*), 3
 setup_credentials() (*in module cmapBQ.config*), 4
 siginfo (*cmapBQ.config.TableDirectory attribute*), 3

T

TableDirectory (*class in cmapBQ.config*), 3
 tables (*cmapBQ.config.Configuration attribute*), 3