

Using the Command Line

Installation

As described in the documentation, HGVA backend is powered by the OpenCGA project. The CLI is distributed with the rest of the OpenCGA code. The OpenCGA code can be cloned in your machine by executing in your terminal. Checkout the latest code (release-1.1.0 branch):

```
$ git clone
https://github.com/opencb/opencga.git
$ git checkout v1.3.6
```

Alternatively, you can download *tar.gz* files with the code for the latest tags/releases of OpenCGA from:

<https://github.com/opencb/opencga/releases>

Once you have downloaded the code, follow the instructions at the *How to Build* section of the OpenCGA repository:

<https://github.com/opencb/opencga>

The CLI interface is accessible through the *opencga.sh* script:

Table of Contents:

- Installation
- Configuration
- Examples
 - Getting information about variants
 - Getting information about projects
 - Getting information about studies
 - Getting information about samples
 - Getting information about cohorts

```

cd opencga
cd build
cd bin
opencga/build/bin$ ./opencga.sh

Program:      OpenCGA (OpenCB)
Version:      1.1.0
Git commit:
f2dace56fcdf491efee8ebb0cb43f981e31c320e
Description:  Big Data platform for processing
and analysing NGS data

Usage:        opencga.sh [-h|--help] [--version]
<command> [options]

Catalog commands:
    users    User commands
    projects Project commands
    studies  Study commands
    files    File commands
    jobs     Jobs commands
    individuals Individual commands
    families Family commands
    samples  Samples commands
    variables Variable set commands
    cohorts  Cohorts commands

Analysis commands:
    alignments Implement several tools for the
genomic alignment analysis
    variant    Variant commands

```

The CLI provides *commands*, *subcommands* and *parameters* to access its functionality. *Commands* of most interest for HGVA users are *projects*, *studies*, *cohorts* and *samples*. Please, find below a list of commands which can be of most interest for HGVA user. Further documentation on the OpenCGA CLI can be found at the [Command Lines](#) section of the OpenCGA documentation.

Configuration

As previously said, the CLI makes intensive use of the RESTful API. Thus, the only configuration detail needed for the CLI to work is a URL where the Web Services API is hosted. The configuration file *client-configuration.yml* is used for this purpose. You shall find a template of this file at the *build/conf* directory:

```

$ ll
opencga/build/conf/client-configuration.yml
-rw-r--r-- 1 fjlopez fjlopez 290 Oct 24 17:49
opencga/build/conf/client-configuration.yml

```

Edit this file with any text editor and set the `rest host` attribute to "<http://bioinfo.hpc.cam.ac.uk/hgva>" :

Configuration file client-configuration.yml

```
---  
## number of seconds that session remain open  
sessionDuration: 12000  
  
## REST client configuration options  
rest:  
  host: "http://bioinfo.hpc.cam.ac.uk/hgva"  
  batchQuerySize: 200  
  timeout: 30000  
  defaultLimit: 2000  
  
## gRPC configuration options  
grpc:  
  host: "localhost:9091"
```

Examples

Getting information about variants

You can query variants by using the *variant* command and *query* subcommand. An extensive list of filtering parameters allow great flexibility on the queries. Please, check inline help provided by *open cga.sh* for further details. For example, get TTN variants from the Genome of the Netherlands study, which is framed within the *reference_grch37* project. We will restrict studies data to those corresponding to GONL. Finally, we will also limit the number of returned results to 3:

```
./opencga.sh variant query --gene TTN --study  
GONL --limit 3 --of json --output-study GONL
```

Getting information about projects

You can use the command *projects* to query projects data.

For getting all metadata from a particular project you can use the *info* subcommand. For example, getting all metadata for the *cancer_grch37* project:

```
./opencga.sh projects info --project  
cancer_grch37
```

For getting all metadata from all studies associated to a particular project you can use the *studies* subcommand. For example, getting all studies and their metadata for the *cancer_grch37* project:

```
./opencga.sh projects studies --project  
cancer_grch37
```

Getting information about studies

You can use the command *studies* to query studies data.

For getting all available studies and their metadata you can use the *search* subcommand. For example, getting all metadata for all available studies (**please note, of special interest will be here the field *alias* which contains the study identifier to be used as an input whenever a study must be passed as a parameter**):

```
./opencga.sh studies search
```

For getting summary data from a particular study you can use the *summary* subcommand. For example, getting summary data for study *1kG_phase3* which is framed within project *reference_grch37*:

```
./opencga.sh studies summary --study  
reference_grch37:1kG_phase3
```

For getting all available metadata for a particular study you can use the *info* command. For example, getting all metadata for study *GONL* which is framed within the project *reference_grch37*:

```
./opencga.sh studies info --study GONL
```

For getting all samples metadata for a given study you can use the *samples* subcommand. For example, getting all samples metadata for study *1kG_phase3* which is framed within project *reference_grch37*. Please, note that not all studies contain samples data, e.g. *GONL*, *ExAC*, among others, only provide variant lists and aggregated frequencies, i.e. no sample genotypes.

```
./opencga.sh studies samples --study  
reference_grch37:1kG_phase3
```

Getting information about samples

You can use the command *samples* to query samples data.

For getting all metadata for a particular sample you can use the *info* subcommand. For example, get all metadata for sample *HG00096* of the *1kG_phase3* study which is framed within the *reference_grch37* project:

```
./opencga.sh samples info --sample HG00096  
--study reference_grch37:1kG_phase3
```

Getting information about cohorts

You can use the *cohorts* command to query cohorts data.

For getting all samples metadata in a given cohort you can use the *samples* subcommand. For example, get all samples metadata for cohort *GBR* from study *1kG_phase3* which is framed within project *reference_grch37*:

```
./opencga.sh cohorts samples --study  
reference_grch37:1kG_phase3 --cohort GBR
```