# GISAID CLI2

## Version 2 Command Line Interface (CLI) for batch uploading

## Description

CLI2 is version 2 of the Command Line Interface (CLI) for batch submissions of meta- and sequence-data to GISAID. Presently, the software allows upload to the EpiCoV database, but future iterations will allow uploading to any of the GISAID databases (EpiFlu, EpiRSV, etc.).

Queries, bug reports, feature requests, etc., should be emailed to `clisupport[at]gisaid.org`.

Use of this package requires some knowledge of the command line. To gain some basic skills using the `*nix` command line, please see e.g., here

## Installation

### pip

Download the compressed archive, then do:

```
tar zxvpf gisaid_cli2.tgz
pip install --user ./gisaid_cli2
cli2 -h
```

### uninstall

```
pip uninstall cli2 -y
```

### conda

### install

Download the latest miniconda installer from https://docs.conda.io/en/latest/miniconda.html and follow miniconda installation instructions.

```
cd gisaid_cli2
conda init --all
export PATH=${PATH}:/bin:/usr/bin:/sbin:/usr/sbin
conda update -y conda
conda env create -f environment.yml
conda activate cli2_env
cli2 -h
```

### uninstall

```
conda remove --name cli2_env --all
```

## Example Usage

### Get help, top level

The software tool is run with the `cli2` command. To get help, trail any command with `-h`.

```
cli2 -h
usage: cli2 [-h]  ...

Command Line Interface (CLI) v2 for uploading sequence and metadata to GISAID.

optional arguments:
-h, --help    show this help message and exit

```

```
Sub-commands help:

    authenticate
                Write the authentication token.
    upload      Upload sequences and metadata.
    version     Show version and exit.
```

There are three sub-commands are available to the user:

- `cli2 authenticate` (refer to section 1, below)
- `cli2 upload` (see section 2)
- `cli2 version` (see section 3)

### 1. Authenticate

Before uploading, you will need to authenticate your connection. This is done by creating an authentication token with `cli2 authenticate`. The generated token is specific to a client-ID, username and password combination. For running test uploads, use the test client-ID: `TEST-EA76875B00C3`. **To obtain a live client-ID, email clisupport[at]gisaid.org.** To reset a token, simply delete the token file and re-run `cli2 authenticate`. If the password field is left blank when running `cli2 authenticate`, the program will ask for it interactively (thus avoiding storing the password in the terminal history). By default, the authentication token file will be `./gisaid.authtoken`; however, you may specify a different path and filename using the `--token` option. Authentication tokens are valid for 100 days. New folders may contain new tokens. Old tokens may be safely deleted.

NB: Keep your unique client-ID in a safe place. Before authenticating with your client-ID, you will need to delete the `gisaid.authtoken` created at the `TEST` run.

To get help on the authentication sub-command, do `cli2 authenticate -h`:

```
usage: cli2 authenticate [-h] [--database {EpiCoV,EpiFlu,EpiRSV}] [--token TOKEN] --
    ↪ username USERNAME
                            [--password PASSWORD] --client_id CLIENT_ID [--proxy PROXY
                                ↪ ] [--debug] [--log LOG]

Write the authentication token.

optional arguments:
-h, --help            show this help message and exit
--database {EpiCoV,EpiFlu,EpiRSV}
                        Target GISAID database. (default: EpiCoV)
--token TOKEN         Authentication token. (default: ./gisaid.authtoken)
--username USERNAME   Your GISAID username. (default: None)
--password PASSWORD   Your GISAID password. Leave blank on shared computers. (default:
    ↪  None)
--client_id CLIENT_ID
                        Submitter's client-ID. Email clisupport@gisaid.org to request
                            ↪ client-ID. (default: None)
--proxy PROXY         Proxy-configuration for HTTPS-Request in the form: http(s)://
    ↪ username:password@proxy:port.
                        (default: None)
--debug               Switch off debugging information (dev purposes only). (default:
    ↪ True)
--log LOG             All output logged here. (default: ./logfile.log)
```

### 2. Upload

There are test data in the `cli2` repository at `cli2/data/{test.csv,test.fa}`. Before performing a test upload with these data:

- copy the example files to a working directory
- in the `test.csv` file, replace `yourusername` in the `submitter` column with your GISAID username
- replace `Anzark` in the `covv_virus_name` column with a random string of your choice; make the identical substitution in the `test.fa` nucleotide sequence headers (denoted by > characters.) The

fasta headers need to match exactly the `covv_virus_name` or the submission for the non-matched labels will not succeed.

After making and saving these changes, do:

```
cli2 upload --metadata test.csv --fasta test.fa
```

For the test upload, a message similar to the following will be printed by default to `stdout` and to `./` ↪ `logfile.log`:

```
missing_seq:    hCoV-19/x/x390/2021
epi_isl_id:     hCoV-19/Anzark/12/2021; EPI_ISL_4348356
epi_isl_id:     hCoV-19/Anzark/13/2021; EPI_ISL_4348357
epi_isl_id:     hCoV-19/Anzark/17/2021; EPI_ISL_4348358
epi_isl_id:     hCoV-19/Anzark/16/2021; EPI_ISL_4348359
validation_error:       hCoV-19/Anzark/17/2021; validation_error; {"covv_virus_name": "
    ↪ already exists"}
validation_error:       hCoV-19/x/x390/2021; validation_error; {"covv_sequence": "
    ↪ field_mandatory_error"}
epi_isl_id:     hCoV-19/Anzark/Melbourne473/2021; EPI_ISL_4348360
epi_isl_id:     hCoV-19/Anzark/Melbourne476/2021; EPI_ISL_4348361
epi_isl_id:     hCoV-19/Anzark/Melbourne477/2021; EPI_ISL_4348362
epi_isl_id:     hCoV-19/Anzark/Melbourne475/2021; EPI_ISL_4348363
upload_count:   submissions uploaded: 8
failed_count:   submissions failed: 2

Total runtime (HRS:MIN:SECS): 0:00:02.668939
```

Run logs will be saved to `logfile.log` with metadata for failed uploads appended `failed.out`.

The output above shows attempted submission of sequences already in the destination will result in `validation error:`, with `"covv_virus_name": "already exists"`. Additional runs will append logs to existing logs by default. All TEST submissions will be sent to a test-server and will not be released to the GISAID EpiCoV-live system.

All EPI_ISL accessions assigned at the test upload stage are dummy accessions, which should not be used in any publications. EPI_ISLs assigned under your live client-ID will be 'live' accession numbers which may be used in publications.

For large submissions, it is recommended to run the upload from inside a `tmux` or `screen` window.

You can manage any successful submissions, which are still in pre-release, by going to the "My Unreleased" tab:

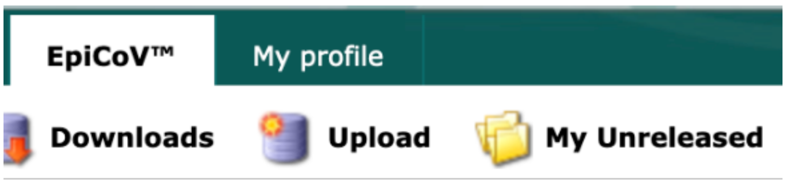

Figure 1: "my unreleased"

To get help on the upload sub-command, do `cli2 upload -h`:

```
usage: cli2 upload [-h] [--database {EpiCoV,EpiFlu,EpiRSV}] [--token TOKEN] --metadata
    ↪ METADATA --fasta FASTA
                   [--frameshift {catch_all,catch_novel,catch_none}] [--failed FAILED]
                       ↪ [--proxy PROXY] [--debug] [--log LOG]

Perform upload of sequences and metadata to GISAID's curation zone.

optional arguments:
-h, --help            show this help message and exit
--database {EpiCoV,EpiFlu,EpiRSV}
                      Target GISAID database. (default: EpiCoV)
--token TOKEN         Authentication token. (default: ./gisaid.authtoken)
```

```
--metadata METADATA   The csv-formatted metadata file. (default: None)
--fasta FASTA         The fasta-formatted nucleotide sequences file. (default: None)
--frameshift {catch_all,catch_novel,catch_none}
                        'catch_none': catch none of the frameshifts and release
                            ↪ immediately; 'catch_all': catch all frameshifts and
                            ↪ require email
                        confirmation; 'catch_novel': catch novel frameshifts and require
                            ↪  email confirmation. (default: catch_all)
--failed FAILED       Name of CSV output to contain failed records. (default: ./failed.
    ↪ out)
--proxy PROXY         Proxy-configuration for HTTPS-Request in the form: http(s)://
    ↪ username:password@proxy:port. (default: None)
--debug               Switch off debugging information (dev purposes only). (default:
    ↪ True)
--log LOG             All output logged here. (default: ./upload.log)
```

### 2.1 Frameshifts

The option `--frameshift` is used to set notification preferences for detected “open reading frame” shifts, or “frameshifts”. The choices for the `--frameshift` option corresponds with options in the drop down menu in the graphic web front-end batch upload submission page. The choices are:

```
catch_all = "Notify me about ALL DETECTED FRAMESHIFTS in this submission for
    ↪ reconfirmation of affected sequences".
catch_novel = "Notify me only about NOT PREVIOUSLY REPORTED FRAMESHIFTS in this
    ↪ submission for reconfirmation of affected sequences"
catch_none = "I confirm ANY FRAMESHIFTS in this submission and request their release
    ↪ without confirmation by a curator"
```

For choice `catch_none`, curators will not notify submitters of frameshifts in the sequence data: records will be released immediately to the live database if all other QC metrics meet release criteria.

Choices `catch_all` (default) and `catch_novel` will result in emails being sent to the submitter, requiring confirmation of frameshifts before release of records to the live database.

If a submission fails part-way through upload, re-run the `cli2 upload` command and the submission should continue from at or near the interruption.

For the correct `csv` and `fasta` file formats, consult the GISAID EpiCoV website. For this, login to gisaid.org, then go to EpiCoV -> Upload -> Batch Upload -> Download Instructions and Template. The first time batch upload is clicked, the user will need to solve a CAPTCHA. Example data formats are additionally provided at `cli2/data/{test.csv,test.fa}`.

### 3. version

To get the software version number, do:

```
cli2 version
```

This user manual is concurrent with version:

```
cli2 version:    2.1.2
```

## Support

Please direct all support enquiries to `clisupport[at]gisaid.org`.

## FAQ

*How do I set preferences for frameshifts notifications for my upload?* Use the `--frameshifts` option as outlined in section **2.1 Frameshifts**.

*I have uploaded via the command line with no error messages, my sequences are not visible in* `my` ↪ `unreleased` *folder or in the live database. Why?* Most likely, the submitter is using the `TEST` client-ID.

To upload to the live database, please request a client-ID at `clisupport[at]gisaid.org` and re-run `cli2 authenticate` with the live client-ID before re-attempting upload.

*What does the error “submitter_invalid” mean? Note, I have created a* `*.authtoken` *file successfully and the upload seemed to be running.* The `submitter` column in the `metadata.csv` file contains a username that is not authenticated for use with the username used to create the `*.authtoken` file. To fix, either: 1) email `clisupport[at]gisaid.org` to add the username/s to the list of authorised usernames for your account (e.g., if you are uploading on behalf of other users); 2) change the username to be equal to the username used to create the `*.authtoken` file, or; 3) re-create the `*.authtoken` file using `cli2 authenticate`, with `--username` equal to the value in the `metadata.csv` file.

*On attempting to upload data, what does the error* `validation_error;{"covv_location": "format_error"}` *mean?* This is caused by incorrect `covv_location` formatting. The following are acceptable: `continent/country/region` and `continent/country/region/sub-region`, but; `country/region/sub-region`, `country/region` and `region/sub-region` are not. Hence, `Europe/Germany/Bavaria/Munich` and `Europe/Germany/Bavaria` are acceptable, but `Mexico/Jalisco/Puerto Vallarta` needs to be corrected to `North America/Mexico/Jalisco/Puerto Vallarta`. Spaces and hyphens in the location are allowed.

*What does* `validation_error;{"covv_sequence": "field_mandatory_error"}` *mean?* This is caused by any of two possibilities. The first cause is actual missing sequence data, so the fasta file contains the sequence header (denoted by the > character) but no sequence data after the header. The second cause is a sequence indicated by `covv_virus_name` is in the `*.csv` file but not in the `*.fa`. Consult the stdout, the log and failed files to further diagnose the problem. If all else fails, email `clisupport[at]gisaid.org` for further assistance.

*I have submitted sequences for which the metadata or sequence data need correcting. How do I correct this*? If the data were recently submitted, make the changes in `My Unreleased`. Otherwise, email `service[at]gisaid.org` with the `EPI_ISL` accessions and the fields to update as a table. Add additional instructions in the email body to assist curators to make the changes as requested.

*I have re-infection samples from a patient for which I have previously submitted sequences. How do I link my new record to previous records?* In the `covv_add_host_info` column, add a note to reference the `EPI_ISL` accession of the previous record.

*How do I add additional information to the location field?* Use the `covv_add_location` field to add the extra data (e.g., LatLong, suburb, postcode etc.).