Altocumulus Documentation

Release 2.1.2

Yiming Yang, Joshua Gould, Bo Li

Aug 16, 2022

CONTENTS

Command line utilities for running workflows on Terra or Cromwell including:

- Run a Terra method, and bulk add/delete methods on Terra.
- Submit WDL workflow jobs to a sever running Cromwell, as well as check jobs' status, abort jobs, and get logs.
- Replace local file paths with remote Cloud (Google Cloud or Amazon AWS) bucket URIs, and automatically upload referenced files to Cloud buckets.
- Parse monitoring log files to determine optimal instance type and disk space.

Important tools used by Altocumulus:

- FireCloud Swagger
- Dockstore Swagger
- FireCloud Service Selector (FISS). In particular, fiss/firecloud/api.py.

CHAPTER

ONE

VERSION 2.1.2 AUGUST 15, 2022

- Bug fix on BCL folder and FASTQ file uploading. [PR #30]
- In `cromwell list_jobs` command, assign informative names for jobs with `nan` workflow name. [PR #31 and #32]

CHAPTER

TWO

VERSION 2.1.1 AUGUST 12,2022

- Add `--type` option to `query` command to specify query type.

CHAPTER

THREE

VERSION 2.1.0 AUGUST 4, 2022

- Altocumulus now only works with Python 3.8+.
- Improve FASTQ file uploading. [PR #28]
- Add `query` command to query project metadata from a LIMS (Laboratory Information Management System) via RESTful APIs.

CHAPTER

FOUR

VERSION 2.0.3 MAY 24, 2022

- Support uploading only the FASTQ files with filename prefix specified within the source folder, instead of the whole folder, to the Cloud. [PR #24]
- In `cromwell list_jobs` command, add `-n` option to show only top *n* jobs. [PR #21]
- Bug fix:
  - Make all the temporary files with filenames unique per process, and remove them even when submission fails.
  - Fix the issue in `cromwell list_jobs` command when workflows' names are not returned by Cromwell API. [PR #20 by Asma Bankapur]
  - Fix the issue in `cromwell get_logs` command when no subworkflow exists in a WDL subtask call. [PR #22]

CHAPTER

FIVE

VERSION 2.0.2 MARCH 16, 2022

- Fix the issue when submitting jobs using Dockstore workflow without specifying version (i.e. implicitly using default version):
  - Dockstore API points to an incorrect path in the top-level `workflow_path` value.
  - So always search through all versions to use the corresponding `workflow_path` inside the default version entry.

CHAPTER

SIX

VERSION 2.0.1 MARCH 1, 2022

- Add `--profile` option to allow use a specific AWS profile when dealing with AWS backend:
  - In **terra** command: `run` and `get_logs` sub-commands.
  - In **upload** command.
- In **cromwell** `run` sub-command:
  - Add `-d` option to allow attach dependency WDL files along with the main workflow WDL file specified in `-m` option.
  - Fix the issue on processing floating numbers in workflow input JSON files.

CHAPTER

SEVEN

VERSION 2.0.0 JANUARY 12, 2022

- Make method-related commands in legacy version as sub-commands under **terra** command, including:
  - `run`, `add_method`, `remove_method`, `storage_estimate`.
- Create sub-commands under **cromwell** command for interaction between users and Cromwell server, including:
  - `run`, `check_status`, `abort`, `get_metadata`, `get_logs`, `list_jobs`.
- Make uploading local data to Cloud buckets a separate command **upload**.
- Add **parse_monitoring_log** command to extract computing resource usage info from monitoring logs generated by Cumulus WDL workflows.

CHAPTER

EIGHT

VERSION 1.1.1 SEPTEMBER 3, 2021

Legacy version:

- Make sure that float values would look the same as in JSON input. For example, if `0.00005` is given, altocumulus should pass `0.00005` instead of `5e-05` to Terra.

8.1 Installation

Altocumulus is released on PyPI, and can be installed using pip:

```
pip install altocumulus
```

To install its development version, do the following:

```
git clone https://github.com/lilab-bcb/altocumulus.git
cd altocumulus
pip install -e .
```

8.2 Usage

8.2.1 Use alto as a command line tool

The alto tool can be used as a command line tool. Type:

```
alto -h
```

to see the help information:

```
Usage:
  alto command_args
  alto -h | --help
  alto -v | --version
```

8.2.2 Terra commands

Terra is a cloud-native platform for bioinformatics analysis workflow execution and biomedical data access. Altocumulus sub-commands under **terra** command are used for workflow operations on Terra workspaces.

alto terra run

Submit workflows to Terra for execution. Workflows can from either Dockstore or Broad Methods Repository. If local files are detected, automatically upload files to the workspace Google Cloud bucket. For Dockstore workflows, collection and name would be used as config namespace and name respectively. Otherwise, namespace and name would be used. After a successful submission, a URL pointing to the job status would be printed out.

Type:

```
alto terra run -h
```

to see the usage information:

```
Usage:
  alto terra run [-h] -m METHOD -w WORKSPACE [--bucket-folder <folder>] -i WDL_INPUTS [-
↪o <updated_json>] [--no-cache]
  alto terra run -h
```

- Options:

  -m METHOD, --method METHOD
  Workflow name. The workflow can come from either Dockstore or Broad Methods Repository. If it comes from Dockstore, specify the name as organization:collection:name:version (e.g. broadinstitute:cumulus:cumulus:1.5.0) and the default version would be used if version is omitted. If it comes from Broad Methods Repository, specify the name as namespace/name/version (e.g. cumulus/cumulus/43) and the latest snapshot would be used if version is omitted.

  -w WORKSPACE, --workspace WORKSPACE
  Workspace name (e.g. foo/bar). The workspace is created if it does not exist

  --bucket-folder <folder>
  Store inputs to <folder> under workspaces google bucket

  -i WDL_INPUTS, --input WDL_INPUTS
  WDL input JSON.

  -o <updated_json>, --upload <updated_json>
  Upload files/directories to the workspace Google Cloud bucket and output updated input json (with local path replaced by google bucket urls) to <updated_json>.

  --no-cache
  Disable call caching.

  -h, --help
  Show this help message and exit

- Outputs:

  URL pointing to the job status

- Examples:

```
alto terra run -m broadinstitute:cumulus:demultiplexing \
               -w "My Workspace Field/Workspace 01" \
               --bucket-folder analysis-01/uploads \
               -i inputs.json \
               -o inputs_updated.json
```

alto terra add_method

Add one or more methods to Broad Methods Repository.

Type:

```
alto terra add_method -h
```

to see the usage information:

```
Usage:
    alto terra add_method [-h] -n NAMESPACE [-p] wdl [wdl ...]
    alto terra add_method -h
```

- Arguments:

  wdl
  Path to WDL file.

- Options:

  -n NAMESPACE, --namespace NAMESPACE
  Methods namespace

  -p, --public
  Make methods publicly readable

  -h, --help
  Show this help message and exit

alto terra remove_method

Remove methods from Broad Methods Repository.

Type:

```
alto terra remove_method -h
```

to see the usage information:

```
Usage:
    alto terra remove_method [-h] -m METHOD
```

- Arguments:

  wdl
  Path to WDL file.

- Options:

-m METHOD, --method METHOD
Method takes the format of namespace/name/version. If only namespace is provided, delete all methods under that namespace. If both namespace and name are provided, delete all snapshots for that method. If namespace, name and version are provided, only delete the specific snapshot.

-h, --help
Show this help message and exit

alto terra storage_estimate

Export workspace storage cost estimates associated with the user to TSV

Type:

```
alto terra storage_estimate -h
```

to see the usage information:

```
Usage:
    alto terra storage_estimate [-h] --output OUTPUT [--access {owner,reader,writer}]
```

- Options:

--output OUTPUT
Output TSV path

--access [owner|reader|writer]
Workspace access levels

-h, --help
Show this help message and exit

8.2.3 Cromwell commands

Cromwell is a widely-used genomics workflow engine to schedule the execution of WDL jobs, running either on an HPC server or a Cloud VM instance. Altocumulus sub-commands under **cromwell** command are used for workflow operations between users and a (remote) server running Cromwell.

alto cromwell run

Submit WDL jobs to a Cromwell server for execution. Workflows should be from Dockstore. For Dockstore workflows, collection and name would be used as config namespace and name respectively. If local files are detected, automatically upload files to the workspace Google Cloud bucket. After a successful submission, a URL pointing to the job status would be printed out.

Type:

```
alto cromwell run -h
```

to see the usage information:

```
Usage:
    alto cromwell run [-h] -s SERVER [-p PORT] -m METHOD_STR -i INPUT [-o <updated_json>
↪] [-b [s3|gs]://<bucket-name>/<bucket-folder>] [--no-cache] [--no-ssl-verify] [--time-
↪out TIME_OUT]
```

- Options:

  -s SERVER, --server SERVER
  Server hostname or IP address.

  -p PORT, --port PORT
  Port number for Cromwell service. The default port is 8000.

  -m METHOD_STR, --method METHOD_STR
  Any of the three forms of workflow WDL file below is accepted:

  - Workflow name from Dockstore, with name specified as **"<organization>:<collection>:<name>:<version>"** (e.g. `broadinstitute:cumulus:cumulus:1.5.0`). If <*version*> part is not specified, the default version defined on Dockstore would be used.
  - An HTTP or HTTPS URL of a WDL file.
  - A local path to a WDL file.

  -d DEPENDENCY_STR, --dependency DEPENDENCY_STR
  ZIP file containing workflow source files that are used to resolve local imports. This zip bundle will be unpacked in a sandbox accessible to the workflow.

  -i INPUT, --input INPUT
  Path to a local JSON file specifying workflow inputs.

  -o <updated_json>, --upload <updated_json>
  Upload files/directories to the workspace cloud bucket and output updated input JSON (with local path replaced by cloud bucket urls) to <updated_json>.

  -b [s3|gs]://<bucket-name>/<bucket-folder>, --bucket [s3|gs]://<bucket-name>/<bucket-folder>
  Cloud bucket folder for uploading local input data. Start with `s3://` if an AWS S3 bucket is used, `gs://` for a Google bucket. Must be specified when **-o** option is used.

  --no-cache
  Disable call-caching, i.e. do not read from cache.

  --no-ssl-verify
  Disable SSL verification for web requests. Not recommended for general usage, but can be useful for intra-networks which don't support SSL verification.

  --time-out TIME_OUT
  Keep on checking the job's status until time_out (in hours) is reached. Notice that if this option is set, Altocumulus won't terminate until reaching *TIME_OUT* hour(s).

  --profile PROFILE
  AWS profile. Only works if dealing with AWS, and if not set, use the default profile.

  -h, --help
  Show this help message and exit

- Outputs:
  - **Case 1:** The ID of the submitted workflow job, which is a series of heximal numbers generated by Cromwell
  - **Case 2:** If **--time-out** option is set, The job ID, along with its final status when terminating, will be returned as a JSON-format string on screen.
- Examples:

```
alto cromwell run -s my-server.com \
                  -m broadinstitute:cumulus:cumulus \
                  -i inputs.json \
                  -o inputs_updated.json \
                  -b s3://my-bucket/analysis-01/uploads \
                  --no-ssl-verify
```

alto cromwell check_status

Check the current status for a workflow on a Cromwell server.

Type:

```
alto cromwell check_status -h
```

to see the usage information:

```
Usage:
    alto cromwell check_status [-h] -s SERVER [-p PORT] --id JOB_ID
```

- Options:

  -s SERVER, --server SERVER
  Server hostname or IP address.

  -p PORT, --port PORT
  Port number of Cromwell service on the server. The default port is `8000`.

  --id JOB_ID
  Workflow ID returned in **alto cromwell run** command.

  -h, --help
  Show this help message and exit

- Outputs:

  The current status of the job in query: *Submitted*, *Running*, *Succeeded*, *Aborting*, *Aborted*, or *Failed*.

- Examples:

```
alto cromwell check_status -s my-server.com --id 710ec6d3-882c-469c-8092-
↪a0b9d5f8dd90
```

alto cromwell abort

Abort a running workflow job on a Cromwell server.

Type:

```
alto cromwell abort -h
```

to see the usage information:

```
Usage:
    alto cromwell abort [-h] -s SERVER [-p PORT] --id JOB_ID
```

- Options:

-s SERVER, --server SERVER
Server hostname or IP address.

-p PORT, --port PORT
Port number for Cromwell service. The default port is `8000`.

--id JOB_ID
Workflow ID returned in **alto cromwell run** command.

-h, --help
Show this help message and exit

- Outputs:

  If the aborting request is sent to the server successfully, a message saying that the job is in status *Aborting* will be printed on screen.

- Examples:

```
alto cromwell abort -s my-server.com --id 710ec6d3-882c-469c-8092-a0b9d5f8dd90
```

alto cromwell get_metadata

Get workflow and call-level metadata for a submitted job.

Type:

```
alto cromwell get_metadata -h
```

to see the usage information:

```
Usage:
    alto cromwell get_metadata [-h] -s SERVER [-p PORT] --id JOB_ID
```

- Options:

  -s SERVER, --server SERVER
  Server hostname or IP address.

  -p PORT, --port PORT
  Port number for Cromwell service. The default port is `8000`.

  --id JOB_ID
  Workflow ID returned in **alto cromwell run** command.

  -h, --help
  Show this help message and exit

- Outputs:

  A local file named `<job-id>.metadata.json` will be created with the job's metadata info in JSON format, where *<job-id>* is the job's ID specified.

- Examples:

```
alto cromwell get_metadata -s my-server.com --id 710ec6d3-882c-469c-8092-
↪a0b9d5f8dd90
```

alto cromwell get_logs

Get the logs for a submitted job.

Type:

```
alto cromwell get_logs -h
```

to see the usage information:

```
Usage:
    alto cromwell get_logs [-h] -s SERVER [-p PORT] --id JOB_ID
```

- Options:

  -s SERVER, --server SERVER
  Server hostname or IP address.

  -p PORT, --port PORT
  Port number for Cromwell service. The default port is `8000`.

  --id JOB_ID
  Workflow ID returned in **alto cromwell run** command.

  --profile PROFILE
  AWS profile. Only works if dealing with AWS, and if not set, use the default profile.

  -h, --help
  Show this help message and exit

- Outputs:

  A local folder named by the job's ID is created. Inside the folder, *stdout* and *stderr* logs of all the WDL tasks and subworkflows of this job are fetched in the same hierarchy as stored on the server's execution folder.

- Examples:

  ```
  alto cromwell get_logs -s my-server.com --id 710ec6d3-882c-469c-8092-a0b9d5f8dd90
  ```

alto cromwell list_jobs

List jobs submitted to the server.

Type:

```
alto cromwell list_jobs -h
```

to see the usage information:

```
Usage:
    alto cromwell list_jobs [-h] -s SERVER [-p PORT] [-a] [-u USER] [--only-succeeded] [-
↪-only-running] [--only-failed] [-n NUM_SHOWN]
```

- Options:

  -s SERVER, --server SERVER
  Server hostname or IP address.

-p PORT, --port PORT
Port number for Cromwell service. The default port is `8000`.

-a, --all
List all the jobs on the server.

-u USER, --user USER
List jobs submitted by this user.

--only-succeeded
Only show jobs succeeded.

--only-running
Only show jobs that are running.

--only-failed
Only show jobs that have failed or have aborted.

-n NUM_SHOWN Only show the <num_shown> most recent jobs.

-h, --help
Show this help message and exit

- Outputs:

  A table of submitted jobs (possibly after filtering specified by options above) with Job ID, creator username, workflow name, status, as well as date and time on submission, start and end of the job. Moreover, jobs in *Succeeded* status are printed in Green color, those in *Failed* or *Aborted* status are in Red color, and those in all the rest statuses are in the default font color of the terminal. By default, *list_jobs* command shows only jobs submitted by the current user.

- Examples:

```
alto cromwell list_jobs -s my-server.com
alto cromwell list_jobs -s my-server.com -a
alto cromwell list_jobs -s my-server.com -u some-username --only-succeeded -n 10
```

8.2.4 Upload to cloud

`alto upload`

Upload files/directories to a Cloud (gcp or aws) bucket.

Type:

```
alto upload -h
```

to see the usage information:

```
Usage:
    alto upload [-h] (-b BUCKET | -w WORKSPACE) [--bucket-folder <folder>] [--dry-run] [-
↪o <updated_json>] input [input ...]
```

- Arguments:

  input
  Input JSONs or files (e.g. sample sheet).

- Options:

-b BUCKET, --bucket BUCKET
Cloud bucket url including scheme (e.g. `gs://my_bucket`). If bucket starts with `gs://`, backend is Google Cloud; otherwise, bucket should start with `s3://` and backend is Amazon AWS.

-w WORKSPACE, --workspace WORKSPACE
Terra workspace name (e.g. foo/bar).

--bucket-folder <folder>
Store inputs to <folder> under workspaces bucket

--dry-run
Causes upload to run in "dry run" mode, i.e., just outputting what would be uploaded without actually doing any uploading.

-o <updated_json>
Output updated input JSON file to <updated_json>

--profile PROFILE
AWS profile. Only works if dealing with AWS, and if not set, use the default profile.

-h, --help
Show this help message and exit

8.2.5 LIMS Query

`alto query`

Query project metadata from a LIMS (Laboratory Information Management System) via RESTful APIs.

Given one project ID, this subcommand tries to fetch and display a vareity of metadata from the project. This subcommand also provides an option to write the metadata into a CSV file if the query type is "ngs".

Using this subcommand requires `lims_query` Python package installed. Users who want to use this subcommand need to write their own lims_query package, which should at least contain one function:

```
query_ngs(project_id: str) -> pandas.DataFrame
```

- Arguments:

--type {ngs,project}
Specify query type. Choose from "ngs" for FASTQ info or "project" for project metadata.

-o CSV_FILE
Write metadata information to a CSV file `CSV_FILE`.

-h, --help
Show this help message and exit.

8.2.6 Logs

`alto parse_monitoring_log`

Output maximum CPU, memory, and disk from monitoring log file

Type:

```
alto parse_monitoring_log -h
```

to see the usage information:

```
Usage:
    alto parse_monitoring_log [-h] [--plot PLOT] path
```

- Arguments:

  path
  Path to monitoring log file.

- Options:

  --plot PLOT Optional filename to create a plot of utilization vs. time -h, --help show this help message and exit