AMDirT

Release 1.4

Maxime Borry, Jasmin Frangenberg, Nikolay Oskolov

Nov 25, 2022

CONTENTS:

Homepage: github.com/SPAAM-community/AMDirT

CHAPTER

ONE

AMDIRT

AMDirT: **A**ncient**M**etagenome**Dir** **T**oolkit

1.1 Install

Before we release AMDirt on (bio)Conda, please follow the instructions below.

1.1.1 1. With pip

. . . upon release of v 1.4

1.1.2 2. With conda

. . . upon release of v 1.4

1.1.3 The latest development version, directly from GitHub

```
pip install --upgrade --force-reinstall git+https://github.com/SPAAM-community/AMDirT.
↪git@dev
```

1.1.4 The latest development version, with local changes

- Fork AMDirT on GitHub
- Clone your fork `git clone [your-AMDirT-fork]`
- Checkout the `dev` branch `git switch dev`
- Create the conda environment `conda env create -f environment.yml`
- Activate the environemnt `conda activate amdirt`
- Install amdirt in development mode `pip install -e .`

1.2 Documentation

- Stable: amdirt.readthedocs.io/en/latest/
- Development version: amdirt.readthedocs.io/en/dev/

CHAPTER

TWO

PYTHON API

2.1 Dataset Validation

class AMDirT.validate.application.**AMDirValidator**(*schema: Union[AnyStr, BinaryIO, TextIO]*, *dataset: Union[AnyStr, BinaryIO, TextIO]*)

Bases: AMDirT.validate.domain.DatasetValidator

Validator Class for AncientMetagenomeDir datasets

__init__(*schema: Union[AnyStr, BinaryIO, TextIO]*, *dataset: Union[AnyStr, BinaryIO, TextIO]*)

Dataset validation class

errors

List of DFError objects
Type list

dataset_name

Dataset name
Type str

schema_name

Schema name
Type str

schema

JSON schema
Type dict

dataset

Dataset as pandas dataframe
Type pd.DataFrame

dataset_json

Dataset as dictionary
Type dict

Parameters

- **schema** (*Schema*) – Path to schema in json format
- **dataset** (*Dataset*) – Path to dataset in tsv format

__repr__()
Return repr(self).

__weakref__
list of weak references to the object (if defined)

check_columns() → bool
Checks if dataset has all required columns

Returns True if dataset has all required columns, False otherwise

Return type bool

check_duplicate_rows() → bool
Checks for duplicated rows in dataset

Returns True if dataset has no duplicated rows, False otherwise

Return type bool

check_multi_values(*column_names: Iterable[str] = ['archive_accession']*) → bool
Check for duplicates entries in multi values column :param column_names: List of multi values columns to check for duplications. Defaults to [“archive_accession”]. :type column_names: Iterable[str], optional

cleanup_errors(*error: jsonschema.exceptions.ValidationError*) → AMDirT.validate.domain.DFError
Cleans up JSON schema validation errors

Parameters **error** (*json_exceptions.ValidationError*) – JSON schema validation error

Returns Cleaned DataFrame error

Return type DFError

dataset_to_json() → dict
Convert dataset from Pandas DataFrame to JSON

Returns Dataset as dictionary

Return type dict

read_dataset(*dataset: Union[AnyStr, BinaryIO, TextIO]*, *schema: dict*) → pandas.core.frame.DataFrame
“Read dataset from file or string :param dataset: Path to dataset in tsv format :type dataset: str :param schema: Parsed schema as dictionary (from read_schema) :type schema: dict

Returns Dataset as pandas dataframe

Return type pd.DataFrame

read_schema(*schema: Union[AnyStr, BinaryIO, TextIO]*) → dict
Read JSON schema from file or string

Parameters **schema** (*str*) – Path to schema in json format

Returns JSON schema

Return type dict

to_markdown() → bool
Generate markdown output table for github display

Returns True if dataset is valid

Return type bool

Raises **SystemExit** – If dataset is invalid

to_rich()

Generate rich output table for console display

Returns True if dataset is valid

Return type bool

Raises **SystemExit** – If dataset is invalid

validate_schema() → bool

Validate dataset against JSON schema

Returns True if dataset is valid, False otherwise

Return type bool

2.2 Dataset conversion

AMDirT.convert.**run_convert**(*samples*, *tables*, *table_name*, *output='.'*, *verbose=False*)

Run the AMDirT conversion application to generate Eager and/or fetchNGS tables

Parameters

- **tables** (*str*) – Path to JSON file listing tables
- **samples** (*str*) – Path to AncientMetagenomeDir filtered samples tsv file
- **table_name** (*str*) – Name of the table of the table to convert
- **output** (*str*) – Path to output table. Defaults to "."

2.3 Dataset filtering

AMDirT.filter.**run_app**(*tables=None*, *verbose=False*)

Run the AMDirT interactive filtering application

Parameters **tables** (*str*) – path to JSON file listing AncientMetagenomeDir tables

2.4 ENA API

class AMDirT.core.ena.**ENAPortalAPI**

Bases: AMDirT.core.ena.ENA

Class to interact with the ENA Portal API

__get_json__(*url: str*) → List[Dict]

Get json content from URL

Parameters **url** (*str*) – URL to get json content from

Returns json content

Return type List[Dict]

__init__() → None

ENA Portal API class

base_url

base URL for ENA Portal API

Type str

__repr__() → str

Display URL of API documentation

__weakref__

list of weak references to the object (if defined)

doc(*dir: str = '.'*) → None

Get PDF documentation for API

Parameters **dir** (*str*) – path to output PDF directory

list_fields(*result_type: str*) → List

Get list of available fields

Parameters

- **result_type** (*str*) – A result is a set of data that can
- **returned** (*be searched against and*) –

Returns list of available fields

Return type List

list_results() → List[Dict]

Get list of available results

Returns list of available results

Return type List[Dict]

query(*accession: str*, *result_type: str = 'read_run'*, *fields: List = ['run_accession', 'sample_accession', 'fastq_ftp', 'fastq_md5', 'fastq_bytes']*) → dict

Generate list of runs metadata for a study accession

Parameters

- **accession** (*str*) – ENA accession
- **result_type** (*str*) – A result is a set of data that can
- **returned** (*be searched against and*) –
- **fields** (*List*) – list of fields to return

Returns run_accession as keys, and metadata as values

Return type dict

status() → bool

Check if API is up

Returns True if API is up, False otherwise

Return type bool

class AMDirT.core.ena.**ENABrowserAPI**

Bases: AMDirT.core.ena.ENA

Class to interact with the ENA Browser API

__get_json__(*url: str*) → List[Dict]

Get json content from URL

Parameters **url** (*str*) – URL to get json content from

Returns json content

Return type List[Dict]

__init__() → None

__repr__() → str

Display URL of API documentation

__weakref__

list of weak references to the object (if defined)

doc(*dir: str = '.'*) → None

Get PDF documentation for API

Parameters **dir** (*str*) – path to output PDF directory

status() → bool

Check if API is up

Returns True if API is up, False otherwise

Return type bool

CHAPTER

THREE

CLI

To access the help menu:

```
$ AMDirT --help
```

The list of arguments of options is detailed below

3.1 AMDirT

AMDirT: Performs validity check of ancientMetagenomeDir datasets
Authors: Maxime Borry, Jasmin Frangenberg, Nikolay Oskolov
Contact: <maxime_borry[at]eva.mpg.de>
Homepage & Documentation: https://github.com/SPAAM-community/AMDirT

```
AMDirT [OPTIONS] COMMAND [ARGS]...
```

Options

--version
Show the version and exit.

--verbose
Verbose mode

3.1.1 convert

Converts filtered samples and libraries tables to eager and fetchNGS input tables

SAMPLES: path to filtered ancientMetagenomeDir samples tsv file
TABLE_NAME: name of table to convert

```
AMDirT convert [OPTIONS] SAMPLES TABLE_NAME
```

Options

-t, --tables <tables>
(Optional) JSON file listing AncientMetagenomeDir tables

-o, --output <output>
conversion output directory

Default .

Arguments

SAMPLES
Required argument

TABLE_NAME
Required argument

3.1.2 filter

Launch interactive filtering tool

```
AMDirT filter [OPTIONS]
```

Options

-t, --tables <tables>
JSON file listing AncientMetagenomeDir tables

3.1.3 validate

Run validity check of ancientMetagenomeDir datasets

DATASET: path to tsv file of dataset to check
SCHEMA: path to JSON schema file

```
AMDirT validate [OPTIONS] DATASET SCHEMA
```

Options

-v, --validity
Turn on schema checking.

-d, --duplicate
Turn on line duplicate line checking.

-c, --columns
Turn on column presence/absence checking.

-i, --doi

Turn on DOI duplicate checking.

--multi_values <multi_values>

Check multi-values column for duplicate values

-m, --markdown

Output is in markdown format

Arguments

DATASET

Required argument

SCHEMA

Required argument

CHAPTER

FOUR

AMDIRT USAGE TUTORIAL

CHAPTER

FIVE

INDICES AND TABLES

INDEX

Symbols

A

B

C

D

E

L

Q

R

S

T

V