TAMSAT Data Download and Extraction API

https://github.com/TAMSAT/tamsat_download_extraction_api
Updated: 20th July 2021

Introduction

The TAMSAT download and extraction API is a Python-based utility that allows users to download TAMSAT rainfall estimates from 1983-present to their local computer and extract rainfall estimates for their own requirements.

The API is similar to the functionality provided TAMSAT's web-based data subsetting tool (http://www.tamsat.org.uk/data-subset/), but offers users the opportunity to perform these tasks themselves in a Python environment and to configure the API to perform multiple tasks without the need to submit multiple requests through the online data subsetting tool. As such, this API will be useful for those wishing to further their Python skills.

The API will be continued to be developed to provide additional functionality in the future. Information about updates to the API will be provided on the TAMSAT website. If you have any questions about the API, please contact tamsat@reading.ac.uk.

API Functionality

The TAMSAT Data Download and Extraction API allows users to download the full Africa-wide rainfall estimates from the TAMSAT Group's public server (http://gws-access.jasmin.ac.uk/public/tamsat/rfe/) and extract locally rainfall estimates for their own requirements.

The extraction options currently available are as follows:

Extraction option	Functionality	Output format
Point extraction	Extract rainfall estimates for a given latitude and longitude over a defined time-period. This is useful if you need a rainfall time-series for a given location.	CSV
Area-average extraction	Extract the area-average rainfall for a given rectangular domain over a defined time-period, specified by providing the north (N), south (S), west (W) and east (E) coordinates of the domain. This is useful if you need an area-average rainfall time-series for a given domain.	CSV
Domain extraction	Extract the gridded rainfall estimates for a given rectangular domain over a defined time-period, specified by providing the north (N), south (S), west (W) and east (E) coordinates of the domain.	netCDF

	The output array has dimensions of longitude, latitude and time. This is useful if you need gridded rainfall estimates for a given domain which can then be used in your own application (e.g. producing maps for bulletin).	

Getting started

Before using the API, you will need the following:

- Ensure an up-to-date installation of Python is working on your computer. If you do not have Python installed, we recommend installing the Python scientific development environment 'Spyder' which will enable you to develop and run Python code. A useful overview to install Spyder on your computer is given here: https://academy.vertabelo.com/blog/how-to-install-python-spyder-ide/
- Download the 'TAMSAT_Data_Download_Extraction_API' zip folder from the GitHub repository (https://github.com/TAMSAT/tamsat_download_extraction_api). Unzip it and save the entire folder in a suitable location on your computer. (If you are using an external hard drive to save the data, you should save the folder to the external hard drive). Do not move, edit or delete any of the files or folders.
- Installing the required Python packages. The TAMSAT Data Download and Extraction API uses a number of Python packages. These add functionality to your Python environment. To ensure you have all the required packages installed:
    1. Open 'Anaconda Prompt'. This can be found in the 'Anaconda' folder in your Start menu.
    2. Type each of the following commands in turn into the Anaconda Prompt window. Some of these may already be installed or may need updating, and some will take a little while to run.

```
conda install xarray
conda install numpy
conda install pandas
conda install wget
conda install datetime
conda install itertools
conda install calendar
```

Once you have successfully installed all the required packages, you are ready to use the API.

Using the API

This API itself is composed of the following scripts:

- `tamsat_download_extract_api.py` [This contains all API source code]
- `run_download_extract.py` [This is the script that users adapt and execute]

Upon download, the script `user_download_extract.py` is configured with some examples which you can adapt to your own requirements.

The API is based on two tools (Python functions):

- **download** – a simple interface to manage the downloading of TAMSAT rainfall estimates for different time steps and spatial resolutions.
- **extract** – a simple interface to manage the extraction of TAMSAT rainfall estimates for the different extraction options (outlined above).

Downloading TAMSAT rainfall estimates

You can download TAMSAT rainfall estimates for different time steps and spatial resolutions. Current available options are:

- time steps: *daily, pentadal (5-day), dekadal (10-day), monthly*
- spatial resolutions: *0.0375° (this is TAMSAT's nominal resolution), 0.25°, 0.50°, 1.00°*

The download tool will accept the following arguments:

Argument	Type	Definition
timestep	string	Time-step - options are: daily, pentadal (5-day), dekadal (10-day), monthly
resolution	float	Spatial resolution - 0.0375°, 0.25°, 0.50°, 1.00°
start_date	string	Start date of the period – format is 'YYYY-MM-DD' where YYYY is the year, MM is the month and DD is the day.
end_date	string	Same as "start_date" but for the end date of the period.
version	float	Version of the TAMSAT rainfall dataset. Current version is 3.1.
localdata_dir	string	Path where the TAMSAT rainfall estimates will be downloaded to and where the extracted data will be written to.

Example:

Here is an example script to download daily estimates for June 2015 at 0.25° resolution.

```
from tamsat_download-extract_api import download, extract

download({
    "timestep": 'daily'
    "resolution": 0.25,
    "start_date": '2015-06-01',
    "end_date": '2015-06-30',
    "version": 3.1,
    "localdata_dir": '/home/user/scripts/tamsat_api/data'
    })
```

Extracting TAMSAT rainfall estimates

You can extract TAMSAT rainfall estimates for different extraction options (point, area-average or domain).

The extract tool will accept the following arguments:

Argument	Type	Definition
extract_type	string	Type of extract – options are: **point**, **area-average** or **domain**.
longitude	float	Longitude of the location of interest (in decimal degrees) – needs to be supplied if 'extract_type = point'.
latitude	float	Latitude of the location of interest (in decimal degrees) – needs to be supplied if 'extract_type = point'.
N	float	North coordinate of the rectangular domain of interest (in decimal degrees) – needs to be supplied if 'extract_type = area_average' or 'extract_type = domain' .
S	float	Same as "N", but for the south coordinate.
W	float	Same as "N", but for the west coordinate.
E	float	Same as "N", but for the east coordinate.
timestep	string	Time-step - options are: daily, pentadal (5-day), dekadal (10-day), monthly
resolution	float	Spatial resolution - 0.0375°, 0.25°, 0.50°, 1.00° *Note: 0.25° is used unless a different resolution is specified.*
start_date	string	Start date of the period – format is 'YYYY-MM-DD' where YYYY is the year, MM is the month and DD is the day.
end_date	string	Same as "start_date" but for the end date of the period.
version	float	Version of the TAMSAT rainfall dataset. Current version is 3.1.
localdata_dir	string	Path where the TAMSAT rainfall estimates will be downloaded to and where the extracted data will be written to.

Example for extracting data at a point:

Here is an example script to extract daily estimates for June 2015 at 0.25° resolution for the location 13.51 N and 22.73 E. Note that 0.25° is used automatically. If a different resolution is required (e.g. 0.0375°), include ["resolution": 0.0375,] in the script below.

```
from tamsat_download-extract_api import download, extract

extract({
    "extract_type": 'point',
    "longitude": 22.73,
    "latitude": 13.51,
    "timestep": 'daily',
    "start_date": '2015-06-01',
    "end_date": '2015-06-30',
    "version": 3.1,
    "localdata_dir": '/home/user/scripts/tamsat_api/data'
    })
```

Executing the above snippet of code will create a CSV file with the following naming convention:

TAMSATv<version>_<timestep>_<resolution>_<longitude>_<latitude>_<start_date>_<end_date>.csv

For example:
TAMSATv3.1_daily_0.25_22.73_13.51_2021-06-01_2021-06-30.csv

Example for extracting area-average data:

Here is an example script to extract area-average daily estimates for June 2015 at 0.25° resolution for the domain 20.0 N, 10.0 S, 30.0 W and 35.0 E. Note that 0.25° is used automatically. If a different resolution is required (e.g. 0.0375°), include ["resolution": 0.0375,] in the script below.

```
from tamsat_download-extract_api import download, extract

extract({
    "extract_type": 'area_average',
    "N": 20.0,
    "S": 10.0,
    "W": 30.0,
    "E": 35.0,
    "timestep": 'daily',
    "start_date": '2015-06-01',
    "end_date": '2015-06-30',
    "version": 3.1,
    "localdata_dir": '/home/user/scripts/tamsat_api/data'
    })
```

Executing the above snippet of code will create a CSV file with the following naming convention:

TAMSATv<version>_<timestep>_<resolution>_<N>_<S>_<W>_<E>_<start_date>_<end_date>.csv

For example:
TAMSATv3.1_daily_0.25_20.0_10.0_30.0_35.0_2021-06-01_2021-06-30.csv

Outputs

All extracted outputs will be saved to a subdirectory called 'extracted_data' within your 'localdata_dir' path and placed into one of the following subdirectories as shown below:

```
extracted_data
├── point
│   ├──
├── area_average
│   ├──
├── domain
│   ├──
```

Tips for using the API

It is strongly recommended to not download the 0.0375° files unless you have sufficient local disk space for these files, adequate internet speeds and computing power to read and process these files. For guidance, one year of 0.0375° daily files is about 450 Mb. As such, we strongly recommend using the 0.25° files as these are considerably smaller in size (one year of daily files is about 15 Mb) but are still sufficiently high spatial resolution.

Extracting data over a long period (e.g. many years) may take a long time depending on the area size selected, the number of years included and the memory of your computer. If you find that the extracting is taking too long or you do not have enough memory, consider the following:

- breaking the period into smaller chunks (e.g. a 30-year period could be split into three 10-year chunks or six 5-year chunks)
- using a coarser resolution
- using a different time-step – will pentadal estimates suffice compared to daily estimates?

Future developments

Future developments will include:

- Extract data for multiple locations
- Download and extract rainfall anomalies and seasonal rainfall estimates
- Ability to mask TAMSAT rainfall estimates on country and administrative-level shape files
- Ability to handle selected periods (e.g. every August between 1983-present).
- Ability to derive climatologies for bespoke periods and years