Chasing rainfall: estimating event precipitation along tracks of tropical cyclones via reanalysis data and in-situ gauges

Jasmine B. D. **Jaffrés**[1], Jessie L. **Gray**[2]
[1]C&R Consulting, Townsville, Australia
[2]School of Health, Medical and Applied Sciences, Central Queensland University, Townsville, Australia

Last revised: 10 June 2023

Table of Contents

1 Introduction

The *rainfall_tracker* toolbox is utilised to extract rainfall within a defined radius from the path (individual track positions) of a low-pressure system – such as tropical cyclones (TCs) or weaker atmospheric lows. The circles are defined based on the haversine formula:

$A = \sin^2\left(\frac{lat_{rad2} - lat_{rad1}}{2}\right) + \cos(lat_{rad1}) \times \cos(lat_{rad2}) \times \sin^2\left(\frac{lon_{rad2} - lon_{rad1}}{2}\right)$, and

$\text{distance (km)} = 2 \times R \times \arcsin(\sqrt{A})$,

where lat_{rad1} and lat_{rad2} (lon_{rad1} and lon_{rad2}) are the radial latitude (longitude) of two points (track point vs rainfall location), and R = 6,371 km is the radius of Earth. The toolbox can be run in either MATLAB or open-source alternative GNU Octave.

To acknowledge the use of this toolbox, please cite:

Jaffrés, J.B.D., Gray, J.L. (2023) Chasing rainfall: estimating event precipitation along tracks of tropical cyclones via reanalysis data and in-situ gauges.

1.1 Software compatibility

The script has been written and tested in MATLAB (R2020b and R2017b) and GNU Octave (v5.1.0; Eaton et al., 2017).

GNU Octave requires several packages (*io*, *nan*, *netcdf* and *statistics*) to be installed. The *rainfall_tracker* toolbox automatically loads these packages when the script is run on GNU Octave.

For Global Historical Climatology Network (GHCN)-Daily data (rainfall_tracker_ghcnd.m), MATLAB requires the Statistics toolbox.

2 Data requirements

To run the *rainfall_tracker* toolbox, a gridded rainfall dataset (in netCDF format; section 2.3.22.3.1) or pre-processed station files from GHCN-Daily (section 2.3.2) are required – in addition to the track(s) of the low-pressure system(s) (section 2.2).

2.1 *rainfall_tracker* toolbox source

The functions and data samples are included in the *rainfall_tracker* .zip file. Download the file from GitHub (https://github.com/jjaffres/rainfall_tracker) or SourceForge (https://sourceforge.net/projects/rainfall_tracker) and unzip the package in your preferred location.

2.2 Tracks of atmospheric low-pressure systems (ALPSs)

Save your file containing the tracks of the atmospheric low(s) in your chosen input directory prior to running the *rainfall_tracker* toolbox. The default location is *'.\data\'*.

2.3 Rainfall dataset

The *rainfall_tracker* toolbox can access rainfall data from both gridded sources (in netCDF format; section 2.3.1) and station files (section 2.3.2). The latter have to be obtained from the GHCN-Daily database (Menne et al., 2012) and reformatted using the freely available *ghcnd_access* toolbox (Jaffrés, 2019) prior to running the *rainfall_tracker* code. Tested rainfall datasets are listed in Table 1.

Table 1: TC track and rainfall datasets accessed for this study.

Name		Type	Period	Website
Short	**Full**			
AWAP	Australian Water Availability Project	gridded, regional (Australia)	1900–present	https://dx.doi.org/10.25914/6009600b58196
ERA5	5th-generation ECMWF reanalysis	gridded, global	1979–present	https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=form
GHCN-Daily	Global Historical Climatology Network for daily station data	point/station data, global	1781–present	ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/
SILO	Scientific Information for Land Owners	gridded, regional (Australia)	1889–present	https://www.longpaddock.qld.gov.au/silo/gridded-data/

2.3.1 Gridded rainfall dataset (netCDF format)

Save your rainfall files in your chosen rainfall input directory prior to running the *rainfall_tracker* toolbox. The script rainfall_tracker.m can be applied on regional or global, gridded rainfall datasets.

Note: Ensure that each filename contains the rainfall data year – and that the filename contains no other 4-digit numbers (see also section 0 for potential error). For rainfall files that only include one year of data, example names are:

1) 2019.daily_rain.nc;
2) SILO_daily_rain_2020. nc;
3) ERA5_totPrecip_2018.nc;
4) totPrecip_ERA5_2021.nc.

If the file contains multiple years of rainfall data, ensure that the filename includes both the first and last year. However, ensure that you separate the two years in the filename (see section 3.4.5 for potential error). For example:

ERA5_totPrecip_2015_2019.nc (file contains rainfall data for 2015 to 2019).

Warning: If you intend to load several rainfall files during one run, you will need to apply a suitable wildcard (*). Especially if you store several rainfall types (e.g. SILO vs ERA5) in the same folder, ensure you specify the target rainfall correctly:

To load any type of netCDF file:

fileNames_rain = '*.nc'

To load any type of netCDF file with ERA5 in the filename:

fileNames_rain = '*ERA5*.nc'

Note: To check the order of the variables in the netCDF file, use the ncdisp() function.

Note: Zero (0) is the position of the first variable in netCDF files.

2.3.2 Station data (GHCN-Daily sourced)

To utilise station data, first download the GHCN-Daily weather station data from the NOAA website ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily. You can either download the complete database or a relevant subset. The files (in .dly format) will then have to be converted into a more accessible structure (.mat format) using the freely available *ghcnd_access* toolbox (Jaffrés, 2019) that can be accessed directly from https://github.com/jjaffres/ghcnd_access.

Once the original GHCN-Daily precipitation (PRCP) data have been reformatted, assign your chosen rainfall input directory prior to running the *rainfall_tracker* toolbox via rainfall_tracker_ghcnd.m.

2.4 List of files

The directory and files structure of the *rainfall_tracker* toolbox are described below.

rainfall_tracker/	Base directory containing main script, user’s guide, readme.txt and all subdirectories.
rainfall_tracker/data/	Directory for the sample track data of TC(s) / atmospheric low(s).
rainfall_tracker/output/	Default output location, including extracted rainfall for each atmospheric low.
rainfall_tracker/subs/	Directory containing all subroutines.

3 Running and modifying the script

3.1 User modifications (gridded rainfall data)

The rainfall_tracker.m script allows for several user modifications. Some code should be modified according to user preferences before the script is invoked. Additional selections are undertaken via the input() function while the script is running.

3.1.1 Pre-run modifications (gridded rainfall data)

in_dirTracks	Directory of track data (line 33).
in_dirRain	Directory of gridded rainfall data (line 34).
out_dir	Directory of output location (line 35).
fileNames_tracks	Identify the filename of the track data (line 38).
headerRows_tracks	Number of header rows in the track data (line 39). Only required for GNU Octave. Default is 1.
timeZone_tracks	Define the time zone of the track data with respect to UTC (line 40). Default is 0 (UTC+0).
trackID	Define the column with unique track identifiers (line 46).
trackID_lon	Define the column with track longitude (line 47).
trackID_lat	Define the column with track latitude (line 48).
trackID_year	Define the column with track year (line 49).
trackID_month	Define the column with track month (line 50).
trackID_day	Define the column with track day (line 51).
trackID_hour	Define the column with track hour (line 52), expressed as a fraction of a day (i.e. between 0 and 1). **Note:** Set to zero if there is no separate column (e.g. if "hour" is incorporated into the "day" variable via decimals).
rainSource	Define rainfall dataset (line 57). Tested options include SILO, ERA5 and AWAP.
fileNames_rain	Identify the (partial) filename of the rainfall data (line 61).
timeZone_rain	Define the time zone of the rainfall data with respect to UTC (line 66). Default is 0 (UTC+0).
unitConv	Define the unit conversion (line 73). Default is 1 (no conversion). **Note:** ERA5 data are in metres. For conversion to mm, apply unitConv = 1000.
rainRadius	Allocate the rain radius around individual track positions of the TC/low (line 78). The default is 500 km.

sideWin	Choose whether to apply a side window (in hours) to the date/time of each track point (line 79). The default is 0 (no side window). **Note:** The side window is applied to both sides, i.e. a side window of 24 (hours) represents a 2-day window. **Hint:** If an offset window (e.g. two days before, one day after) is wanted, this can be achieved by shifting the actual timezone (e.g. timeZone_tracks = 0 UTC) by the applicable amount (± shift). For this example, apply sideWin = 1.5×24 (half of three days) in conjunction with timeZone_tracks = 0 (UTC) – 12 (hours).
rain_options	Rain statistics to be exported (line 85). Options are: 'total' event rain (sum) over the period of the track. 'maximum' maximum rainfall intensity per grid point. 'raw' individual outputs for every rain time period.
rainID_Lon	ID (order) of longitude in rainfall netCDF file (line 93). **Note:** netCDF files are 0-indexed, i.e. ID (order) of first variable in any netCDF file is 0! (MATLAB is 1-indexed).
rainID_Lat	ID (order) of latitude in rainfall netCDF file (line 94).
rainID_Time	ID (order) of time in rainfall netCDF file (line 95).
rainID_Rain	ID (order) of rainfall in rainfall netCDF file (line 96).
outName_suffix	The suffix for the output rainfall file (.mat format; line 98).

3.2 User modifications (GHCN-Daily rainfall data)

The rainfall_tracker_ghcnd.m script allows for several user modifications. Some code should be modified according to user preferences before the script is invoked.

3.2.1 Pre-run modifications (GHCN-Daily rainfall data)

in_dirTracks	Directory of track data (line 37).
in_dirRain	Directory of gridded rainfall data (line 38).
out_dir	Directory of output location (line 39).
fileNames_tracks	Identify the filename of the track data (line 42).
headerRows_tracks	Number of header rows in the track data (line 43). Only required for GNU Octave. Default is 1.
timeZone_tracks	Define the time zone of the track data with respect to UTC (line 44). Default is 0 (UTC+0).
trackID	Define the column with unique track identifiers (line 50).
trackID_lon	Define the column with track longitude (line 51).
trackID_lat	Define the column with track latitude (line 52).

trackID_year	Define the column with track year (line 53).
trackID_month	Define the column with track month (line 54).
trackID_day	Define the column with track day (line 55).
trackID_hour	Define the column with track hour (line 56), expressed as a fraction of a day (i.e. between 0 and 1). **Note:** Set to zero if there is no separate column (e.g. if "hour" is incorporated into the "day" variable via decimals).
fileNames_rain	Identify the (partial) filename of the rainfall data (line 61).
timeZone_rain	Define the time zone of the rainfall data with respect to UTC (line 64). Default is 0 (UTC+0).
rainRadius	Allocate the rain radius around individual track positions of the TC/low (line 71). The default is 500 km.
sideWin	Choose whether to apply a side window (in hours) to the date/time of each track point (line 72). The default is 0 (no side window). **Note:** The side window is applied to both sides, i.e. a side window of 2 (hours) represents a 4-hour window. **Hint:** If an offset window (e.g. two days before, one day after) is wanted, this can be achieved by shifting the actual timezone (e.g. timeZone_tracks = UTC+0) by the applicable amount (± shift). For this example, apply sideWin = 1.5×24 (half of three days) in conjunction with timeZone_tracks = 0 (UTC) – 12(hours).
outName_suffix	The suffix for the output rainfall file (.mat format; line 80).

3.3 Warning messages and input() modifications

3.3.1 Unexpected track longitude or latitude

The code for the subsequent input (*scriptCheck*) is located in the track_check.m subroutine. The query will only be invoked if there appears to be an issue with the longitude or latitude of the track data. If longitude is outside its expected range (-180 to +360), the subsequent message will be followed by a warning:

> *Longitude ranges from* [minimum longitude] *to* [maximum longitude].
>
> *Are you sure you assigned the correct .csv column to longitude?!*

If latitude is outside its expected range (-90 to +90), the subsequent message will be followed by a warning:

> *Latitude ranges from* [minimum latitude] *to* [maximum latitude].
>
> *Are you sure you assigned the correct .csv column to latitude?!*

In both instances (longitude or latitude flag), an input() query requiring a user choice will then appear:

> *Do you want to continue (1) or stop the run (0)?* -->

If the user opts to ignore the warning (1), the following message will appear before the code continuous with the data aggregation despite the potential problems with the assigned longitude and/or latitude for the TC tracks:

> *You have chosen to continue the run.*

Otherwise, an error will be generated (see section 3.4.15):

> *Fix the* [longitude or latitude] *issues in the track file (or column assignment).*

Note: It is recommended that both the TC file and the assigned numbers for *trackID_lon* and/or *trackID_lat* (cf. section 3.1.1) are rechecked before re-running the script.

3.3.2 Unexpected track dates

The *rainfall_tracker* toolbox checks whether the dates of individual track data are monotonously increasing. If an unexpected date order is found (reverse order or identical dates) for any of the tracks, then a warning and a follow-up message will be generated:

> *Not all track data are monotonously increasing.*
>
> *An additional .mat file (with the prefix "Check") will be generated.*

This output file should be reviewed – with the relevant track data file amended as appropriate (see also section 4.3).

3.3.3 Unexpected time zone for track data

The code for the subsequent input is located in the track_load.m subroutine. The *rainfall_tracker* toolbox checks whether the assigned time zone for track data is, as expected, UTC+0. If a different time zone was assigned, then a warning message is generated to check if the user wants to modify the parameter:

> *The assigned time zone for your track data is unexpectedly UTC*[assigned time zone] *instead of UTC+0.*

After the above message, an input() query requiring a user choice will appear:

> *Do you want to stop (0) the run - or continue (1) with a track time zone of UTC*[assigned time zone]*?* -->

If the user opts to proceed with the unexpected time zone, the following message will appear before the code continuous with the data aggregation:

> *You have chosen to continue the run with the selected time zone (UTC*[assigned time zone]*) for track data.*

Otherwise, another message followed by an error will be generated (see section 3.4.3):

> *You have indicated that the time zone for the track data should be corrected.*

Fix "timeZone_tracks" in line [relevant line number] *in* [relevant file name].

3.3.4 Unexpected format of track hours

The code for the subsequent input (*hourCheck*) is located in the track_load.m subroutine. The *rainfall_tracker* toolbox checks whether the hour variable for the track data is in the expected range (0 to 1, i.e. the fraction of one day). If the hours are outside that range, then a warning and follow-up message are generated to check if the user wants to divide the hours by 24 (i.e. if the current format is in the 0 to 24 format):

Your track hours [minimum hour] *to* [maximum hour] *are not in the expected range (0 to 1).*

Are your hours in 24-hour format instead? If so, do you want to divide your hours by 24?

After the above messages, an input() query requiring a user choice will appear:

Do you want to continue (1) by applying hours/24 or stop the run (0) and fix your track hours? -->

If the user opts to divide the hours by 24 (1), the following message will appear before the code continuous with the data aggregation:

The track hours are now divided by 24 to obtain a range of 0 to 1.

Otherwise, an error will be generated (see section 3.4.4):

Fix your track hour format.

3.3.5 Default warning for SILO and AWAP data

Both SILO and AWAP are Australian, regional databases that variable time periods (relative to UTC) across their extent. Consequently, each time the rainfall_tracker.m with one of those types of rainfall datasets selected, the following warning is invoked:

[*rainSource*] *rainfall data have non-uniform dates - with state- and season-specific timezones.*

This warning message is followed by the following text:

If your TC tracks cover more than one timezone (of the rainfall data):

1) Split your TC dataset into:

a) Specific regions (e.g. Western Australia vs Queensland); and/or

b) Winter vs summer (if daylight saving is applied); or

2) Accept that your TC data will be misaligned with rainfall by up to three hours (for Australia-wide studies).

No action is required unless the user wants to terminate the run. The following message will then appear before the code continuous with the data aggregation:

You have chosen to continue the run with the selected time zone for rainfall.

3.3.6 Unexpected time zone for gridded, regional precipitation data

The code for the subsequent input (cont) is located in the rain_type.m subroutine. The query will only be invoked if an unexpected time zone for a gridded dataset is determined, with the message varying dependent on the accessed rainfall data type (e.g. SILO).

AWAP and SILO are two regional datasets that are limited to Australia. Consequently, the expected time zone would usually range from UTC+8 (Western Australia) to UTC+11 (south-eastern Australia during daylight saving, i.e. austral summer). When the chosen time zone (timeZone_rain) is outside its expected range (+8 to +11), the subsequent warning will be followed by a message:

> *Your selected time zone is* [time zone].
>
> *However, the time zone for* [rain data type] *ranges from +8 (AWST) in Western Australia to +9.5 (ACST) in the Northern Territory and +10 (AEST) in Queensland and +11 (AEDT) in New South Wales during summer.*

For gridded, global datasets, the following warning will be invoked:

> *Your selected time zone is* [time zone] instead of 0 UTC (the most common time zone for global datasets).

After the initial warning, an input() query requiring a user choice will then appear:

> *Do you want to stop (0) the run - or continue (1) with a time zone of* [time zone]*? -->*

If the user opts to ignore the warning (1), the following message will appear before the code continuous with the data aggregation:

> *You have chosen to continue the run with the selected time zone for rainfall.*

Otherwise, an error will be generated (see section 0):

> *You have indicated that "timeZone_rain" needs to be corrected. See line 66 in rainfall_tracker.m!*

3.3.7 Unexpected longitude or latitude for gridded rainfall

The code for the subsequent input is located in the rain_type.m subroutine. The query will only be invoked if there appears to be an issue with the longitude or latitude of the gridded rainfall data. If longitude is outside its expected range (-180 to +360), the subsequent message will be followed by a warning:

> *Longitude ranges from* [minimum longitude] *to* [maximum longitude].
>
> *Are you sure you assigned the correct netCDF ID (order) to longitude?!*

If latitude is outside its expected range (-90 to +90), the subsequent message will be followed by a warning:

> *Latitude ranges from* [minimum latitude] *to* [maximum latitude].
>
> *Are you sure you assigned the correct netCDF ID (order) to latitude?!*

In both instances (longitude or latitude flag), an input() query requiring a user choice will then appear:

> *Do you want to stop (0) the run - or continue (1) with* [longitudes or latitudes] *of* [minimum value] *to* [maximum value]*?* -->

If the user opts to ignore the warning (1), the code will continue with the data compilation despite the potential problems with the assigned longitude and/or latitude for the rainfall. Otherwise, an error will be generated (see section 3.4.9):

> *You have indicated that longitude or latitude needs be corrected. See lines 93-94 in rainfall_tracker.m!*

Adjust the relevant code lines in rainfall_tracker.m before re-running the script.

3.3.8 Multiple GHCN-daily input files

The *rainfall_tracker* toolbox expects to only find one rainfall input file when GHCN-daily data are wanted (see rainfall_tracker_ghcnd.m). If more than one relevant file is located, the following warning is displayed (see ghcnd_load.m subroutine):

> *Are you aware that there is more than one GHCN-daily file*

The warning is then followed by the subsequent message:

> *Only the first file will be accessed!*

3.4 Errors and caveats

The rainfall_tracker.m script was written in the assumption that the rainfall files are in yearly format (i.e. one or multiple years per file), with each filename containing the (first and last) year (and no other 4-digit number). Additionally, the code is only intended for daily and sub-daily rainfall data extraction.

Similarly, to successfully run rainfall_tracker_ghcnd.m, the original GHCN-Daily rainfall data – in daily format – first has to be downloaded and modified via the freely available *ghcnd_access* toolbox (Jaffrés, 2019).

The *rainfall_tracker* toolbox tests for various potential errors, outlined in sections 3.4.1 to 3.4.17.

3.4.1 Path and identification of rainfall data files

The pre-run user inputs (section 3.1.1) require the identification of the appropriate path to the rainfall data (*in_dirRain*) and applicable (partial) filename (*fileNames_rain*). If the script cannot identify any corresponding files (e.g. because the path is incorrect), the following error message will be generated (see ghcnd_load.m or rain_aggregate.m):

> *The rainfall file list is empty. Check the path (*[your stipulated path]*) and filename (*[your chosen filename]*)!*

Check both user inputs (section 3.1.1) before re-running the toolbox.

Note: For GHCN-Daily data, the *rainfall_tracker* toolbox expects that precipitation (PRCP) from the station files were first extracted via the *ghcnd_access* toolbox (section 2.3.2).

3.4.2 Unexpected GHCN-Daily data format

When the user is intending to run rainfall_tracker_ghcnd.m, the precipitation derived from GHCN-Daily (PRCP) will first have to be compiled separately via the freely available *ghcnd_access* toolbox (Jaffrés, 2019) (section 2.3.2). The resulting .mat should contain one or several *ghcnd_gauge_info** variables. If absent, a warning message will be displayed, followed by an error:

> *Have you first applied ghcnd_access to aggregate the GHCN-daily .dly data?*
>
> *Your file does not contain any variables in the ghcnd_gauge_info* format.*

Ensure you have obtained a PRCP output file via the *ghcnd_access* toolbox (section 2.3.2) before re-running *rainfall_tracker*.

3.4.3 Unexpected time zone for track data

The *rainfall_tracker* toolbox checks whether the assigned time zone for track data is, as expected, UTC+0. If the user accidentally assigned the incorrect time zone and opts to terminate the run, then the following error will be generated (see section 3.3.3):

> *Fix "timeZone_tracks" in line* [relevant line number] *in* [relevant file name].

Review the assigned value for timeZone_tracks in the relevant file.

3.4.4 Unexpected format of track hours

The *rainfall_tracker* toolbox checks whether the hour variable for the track data is in the expected range (0 to 1, i.e. the fraction of one day). If the hours are outside that range – and the user has not opted to divide the hours by 24 via the input() function (cf. section 3.3.1), an error will be generated:

> *Fix your track hour format.*

Check your hours format for your track data.

Note: If you do not have a separate column for hours (i.e. if already incorporated with the "day" variable), then ensure *trackID_hour* is set to zero (0).

3.4.5 Rainfall filename (gridded rainfall)

Rainfall data from filenames are extracted under the assumption that:

1) Each file comprises one or several complete (calendar) years of data; and
2) The filename includes the first and last year of contained data.

The 2nd assumption additionally requires that:

1) The filename contains the (first and last) year(s) and no other 4-digit number;

2) If the file contains data for multiple years, a gap between the two years is required in the filename (e.g. *ERA5_totPrecip_2015_2019.nc*, containing data for years 2015 to 2019).
    a. **Note:** The following notations are not allowed: *20152019* (no gap between years), *012015* (additional values, including month, attached to the year), etc.

If the first filename (in your directory) contains more than two 4-digit numbers, the following error will be displayed:

> *Your rainfall filenames contain more than two 4-digit numbers! See for example* [path and filename]

If the first filename contains no 4-digit number, the following error will be displayed:

> *Your rainfall filenames contain no 4-digit number (== year)! See for example* [path and filename]

Review the filename for the rainfall data (cf. *in_dirTracks* and fileNames_rain; section 3.1.1). Once relevant corrections have been applied, re-run the toolbox.

Note: Warning messages are generated if your rainfall dataset excludes years – or includes duplicate years – based on the filenames. If your dataset should be complete and without repetition, review the filenames and data content. For missing years, relevant messages include:

> *Are you aware that you do not have data for all years between* [oldest and newest year] *based on the filenames?!*
>
> *The missing years (based on filenames) are:*
>
> [list of missing years]

For duplicate years, relevant messages include:

> *Are you aware that you seem to have duplicate data based on the filenames?!')*
>
> *The duplicate years (based on filenames) are:*
>
> [list of duplicate years]
>
> *For each of these years, only the first file (in alpha-numerical order) will be accessed!*

If, later in the code, some of these files are relevant for individual tracks, the following messages will be displayed:

> *A total of* [number of files with the target year] *ERA5 files with rainfall data exist for the year* [target year]*!*
>
> *The following files will be ignored:*
>
> [file name (one per line)]
>
> *Instead, only the file "*[the first file with the target year]*" will be accessed!*

3.4.6 Time discrepancy between track and (gridded) rainfall data

The script was written under the assumption that rainfall data are available for all input tracks of the atmospheric lows. An error (along with two preceding messages) will be produced if the year of the track does not have a corresponding rainfall file:

> *Check whether you have rainfall data for the year* [year]*!*
>
> *If you do not have rainfall data for* [year]*, remove relevant tracks and then re-run the script!*
>
> *The year* [year] *for tracks of atmospheric lows does not seem to have corresponding rainfall data!*

3.4.7 (Gridded) netCDF data order and dimensionality

The user has to define the ID (order) of four parameters (longitude, latitude, time and rainfall) in the netCDF file(s) to be accessed. The toolbox will first check whether you have assigned a unique ID for each of these parameters. If a duplicate is determined, the following message and error will be displayed:

> *At least one rainfall parameter (rainID_Lon, rainID_Lat, rainID_Time or rainID_Rain) have been specified incorrectly!*
>
> *Check lines 93 to 96 in "rainfall_tracker.m" (duplicate extraction ID)!*

The code next tests whether the dimensionality of the four parameters is as expected (1D for longitude, latitude and time, 3D or 4D for rainfall).

Note: Only one file (the last relevant file contained in your “in_dirRain” folder) is assessed.

If the user-defined netCDF IDs (data order) lead to the extraction of data with unexpected dimensionality, then the following error will be displayed:

> *The dimensionality of "*[affected variable type]*" is unexpected (*=[extracted data dimension]*) - check* [affected parameter] *in rainfall_tracker.m!*

Review the affected parameter (see lines 93–96 in rainfall_tracker.m). Once the correction has been applied, re-run the toolbox.

Warning: Ensure that the ID (order) assignment is based on 0-indexing (for netCDF files), with the ID (order) of the first variable = 0!

3.4.8 (Gridded) rainfall data identification – time zone

If the user-defined time zone is unexpected, a user input will be requested to either continue or stop the script run. If the user does not elect to continue the run, the following error will be displayed:

> *You have indicated that "timeZone_rain" needs to be corrected. See line 66 in rainfall_tracker.m!*

Review the time zone assigned to rainfall data (cf. timeZone_rain; section 3.1.1). Once the correction has been applied, re-run the toolbox.

Note: For most global, gridded rainfall databases, a time zone of 0 (UTC) is expected, whereas the two regional rainfall datasets (AWAP and SILO) can range from +7 (AWST) to +11 (AEDT).

3.4.9 (Gridded) rainfall data identification – netCDF data order

If the longitudinal or latitudinal range is unexpected, a user input will be requested to either continue or stop the script run – to check whether the user-defined netCDF data position is correct. If the user does not elect to continue the run, the following error will be displayed:

> *You have indicated that longitude or latitude needs be corrected. See lines 93-94 in rainfall_tracker.m!*

Review the assigned data order for rainfall longitude and latitude (cf. *trackID_lon* and *trackID_lat*; section 3.1.1). Once the correction has been applied, re-run the toolbox.

3.4.10 Date for (gridded) rainfall data

If the incorrect netCDF variable is assigned to "time", the following warning may be obtained (depending on the allocated variable type):

> *The incorrect "time" variable was loaded for the first netCDF file!*

The following error will then be displayed:

> *Have you applied the incorrect time ID (order) in the netCDF file (e.g. for 4D ERA5 rain files)?!*

Check the correct netCDF dimensions for all variables (e.g. beware of differences between 3D and 4D ERA5 files) before rerunning the script.

3.4.11 Selected output type for gridded rainfall

Output for GHCN-Daily data automatically includes all types of rainfall options (i.e. event rainfall, maximum intensity and raw data for relevant stations and time periods). Conversely, because of more substantial data storage requirements, only one data type is extracted for gridded precipitation per run. Currently implemented options (*rain_options*) include *total* (event rainfall) and *maximum* (peak intensity; see section 3.1.1). If another term was assigned to *rain_options*, then the following error will be displayed:

> *You have not specified a valid rainfall variable (cf. rain_options on line 83).*

Review the chosen data type (cf. *rain_options*; section 3.1.1). Once the correction has been applied, re-run the toolbox.

3.4.12 Gridded multi-day rainfall

The toolbox was written to extract relevant precipitation from files with daily or sub-daily temporal resolution. If the code detects a timestep in excess of 24 hours, the following error will be displayed:

> *Your rainfall file is neither daily or subdaily (*[number of] *days)!*

Review the chosen data type (cf. *rainSource*; section 3.1.1) and/or netCDF data files. Once the correction has been applied, re-run the toolbox.

3.4.13 Rainfall data scaling – netCDF data unit

If the scaling factor (*unitConv*) is unexpected, a user input will be requested to either continue or stop the script run – to check whether the user-defined unit conversion is correct. If the user does not elect to continue the run, the following error will be displayed:

> *You have indicated that the rainfall conversion (unitConv) needs be corrected. See line 73 in rainfall_tracker.m!*

Review the scaling factor (cf. *unitConv*; section 3.1.1). Once the correction has been applied, re-run the toolbox.

3.4.14 Access to track file

Immediately prior the following error, the path and filename for the track data will also be displayed:

> *Check that you have defined the path and filename for the track data correctly!*

Review the displayed path and filename for TC data (cf. *in_dirTracks* and *fileNames_tracks*; section 3.1.1). Once relevant corrections have been applied, re-run the toolbox.

3.4.15 Discrepancy with longitude/latitude of tracks

If the user input for *scriptCheck* (section 3.3.1) is 0 (zero), then the script will stop after displaying an error message, either relating to longitude:

> *Fix the longitude issues in the track file (or column assignment).*

or latitude:

> *Fix the latitude issues in the track file (or column assignment).*

The recommendation is to recheck both the track file and the assigned numbers for *trackID_lon* and/or *trackID_lat* (cf. section 3.1.1). Once relevant corrections have been applied, re-run the toolbox.

3.4.16 Super-zombie versus track name duplicates

Names of TCs are not necessarily unique and can be repeatedly used if the TC (of that name) was not particularly deadly or costly. Consequently, in the absence of another, unique TC identifier, the toolbox would assume that all TCs of the same name correspond to one track. To test for such occurrences, the toolbox checks whether the first and last track position are more than a year apart. For any such instances, the following error is displayed:

> *Super-zombie (track lasts over a year) - check for potential database issues (e.g. duplicate name).*

Immediately prior to this error output, the relevant TC name will also be displayed. First check that you have identified the correct column for TC names (cf. *trackID*; section 3.1.1). If so, then review the TC name (i.e. check for duplicates) in your .csv file. Once relevant corrections have been applied, re-run the toolbox.

3.4.17 Discrepancies in rainfall data dimensionality

The toolbox was written under the assumption that all rainfall data (from an individual source – e.g. SILO, ERA5 or AWAP) are formatted consistently (i.e. identical data order and dimensionality). If this is not the case, the periods of consistent rain formatting should be identified – and track data split accordingly – before re-running the script for each period separately (using the subset of tracks).

The most likely scenario – when rainfall data inconsistencies may occur – is the addition of new precipitation data with different formatting. For example, the netCDF file for ERA5 year 2021 total precipitation (downloaded on 12 January 2022) included an additional variable (expver), increasing the dimensions from three (lat/long/time) to four (lon/lat/expver/time).

Specifically for ERA5, a try-catch subroutine was written to avoid an error message for 4D rainfall data (see rain_access.m subroutine, lines 65 to 83). However, the user still has to ensure that the loaded rainfall files all have otherwise the same formatting (e.g. rainfall data consistently listed in the 4th rather than 5th netCDF position; see lines 93–96 in rainfall_tracker.m) to prevent various potential warning message invoked by MATLAB / GNU Octave.

If the *rainfall_tracker* toolbox is run based on 3D ERA5 positions of netCDF variables but there are 4D files among the imported ERA5 data (and a track is included that coincides with rainfall in 4D format), then the following warning may appear:

> *Have you called rain files with mixed formatting (e.g. 3D vs 4D ERA5)?!*

The warning is then followed by the subsequent message:

> *If so, 1) either ensure you only call the relevant subset (if files are in the same folder), or... 2) move files with different formatting into another folder!*

MATLAB / GNU Octave will then produce an error message.

3.5 Script running

The following message should appear once the optional inputs have been set and processed:

> *User input is now complete and your rainfall data will now be collated*.

At completion, the following message will be displayed for gridded rainfall data:

> *The rainfall data were successfully collated for each low-pressure system*.

For GHCN-Daily data, the completion message will read:

> *The daily rainfall station data were successfully collated for each low-pressure system*.

The time required to complete the data extraction will depend on:

1) The rainfall dataset (gridded vs point/station data);

2) The number of (low-pressure system) tracks for which rainfall data have to be extracted; and
3) The applied software (MATLAB runs the code significantly faster than GNU Octave).

Further, the computer hardware and storage location of the rainfall data can also greatly affect the run time.

4 Output

Once the *rainfall_tracker* toolbox has been successfully run, a MAT-file (.mat) will be generated.

4.1 Gridded data file content

The default filename is [prefix based on *rain_options*]_[*rainSource*]_[*outName_suffix*].*mat* (in save_rain.m), where *rain_options* refers to the type of rainfall (e.g. *eventRain*), *rainSource* is the chosen rainfall datasets and *outName_suffix* is the user-specified name of the rainfall dataset.

readme	A descriptor of each variable saved in the output file.
rainRadius	The radius (from the centre of the low, at individual track positions) applied for the rainfall extraction.
sideWin	The applied side window (hours) for each track position.
timeZones	Assigned time zones for TC tracks (position 1) vs rainfall (position 2).
lonR	The longitude vector (based on the input rainfall data). Each row corresponds to the equivalent position in the 1st dimension of event_rain.
latR	The latitude vector (based on the input rainfall data). Each row corresponds to the equivalent position in the 2nd dimension of event_rain.
trackIDs	List of unique TCs/lows for which rainfall was tracked and aggregated. Each row corresponds to the equivalent position in the 3rd dimension of event_rain.
rain_options	Rain statistics to be exported (line 78). Options are: 'total' event rain (sum) over the period of the track 'maximum' maximum rainfall intensity per grid point
event_rain	The total rainfall along each track (within the area stipulated by the track and *rainRadius*) for each grid point. The 3D matrix dimensions correspond to lonR, latR and trackIDs. Only included if the selected *rain_options* is 'total'.
header_event_rain	Header for event_rain, describing each dimension of the 3D matrix. Only included if the selected *rain_options* is 'total'.
peakIntensity	The peak rainfall intensity along each track (within the area stipulated by the track and *rainRadius*) for each grid point. The 3D matrix dimensions correspond to lonR, latR and trackIDs. Only included if the selected *rain_options* is 'maximum'.

header_peakIntensity	Header for peakIntensity, describing each dimension of the 3D matrix. Only included if the selected *rain_options* is 'maximum'.

4.2 GHCN-Daily data file content

The default filename is *eventRain_ghcnd_*[*outName_suffix*].*mat* (line 100 in rain_aggregate_ghcnd.m), where *outName_suffix* is the user-specified suffix.

readme	A descriptor of each variable saved in the output file.
rainRadius	The radius (from the centre of the low, at individual track positions) applied for the rainfall extraction.
sideWin	The applied side window (hours) for each track position.
timeZones	Assigned time zones for TC tracks (position 1) vs rainfall (position 2).
lonlat	Vector of matrices that list relevant longitudes and latitudes (per track) for station rainfall data.
dims_lonlat	Header for lonlat_ghcnd.
pos_stationsIn[1]	Positions (of original GHCN-Daily list) of stations that are inside the circle at least once.
event_rain	Cell vector containing 1D vector of total rainfall per track and relevant weather station.
peakIntensity	Cell vector containing 1D vector of peak (24-hour) rainfall intensity per track and relevant weather station.
daily_rain	Cell vector of matrices that lists daily rainfall for all stations inside the circle.
daily_dates	Cell vector containing 1D vectors of relevant dates.
inCircle_YN	Cell vector of logical 2D matrices that shows which stations were inside the circle (regardless of data availability).
stationIDs	List of all relevant stations (per track).
trackIDs	List of unique TCs/lows for which rainfall was tracked and aggregated. Each row corresponds to the equivalent position in daily_dates, daily_rain, event_rain, inCircle_YN and pos_stationsIn.

[1] This output variable (pos_stationsIn) is generally redundant. However, if the *rainfall_tracker* result is to be compared to the original GHCN-Daily input data (e.g. to estimate proportional contributions by individual cyclones per station), this parameter permits cross-referencing to the input file.

4.3 Supplementary file: check track dates

The *rainfall_tracker* toolbox checks whether the dates of individual track data are monotonously increasing (see section 3.3.2). If an unexpected date order is found for any of the tracks, then an additional .mat files will be generated based on *fileNames_tracks* (see sections 3.1.1 and 3.2.1), with the prefix *Check_*. The .mat file will contain a *readme* and a *checkData* variable.

4.4 Matrix size limit in MAT-files

Note that MAT-files created with GNU Octave tend to be significantly larger than the equivalent MAT-files saved by MATLAB (e.g. 1 GB vs 21 MB, respectively). In addition, the code will take longer to run and compile the data.

5 Caveat

The script has several caveats that the user should be aware of.

5.1 Track dates not overlapping with rainfall data

The script assumes that rainfall data are available for all years coinciding with the tracks of the low-pressure systems. If any rainfall years are missing, the script will not run successfully. Hence, remove any TC tracks for which you have no corresponding rainfall data before running the code.

5.2 Track location outside the area of regional rainfall datasets

For regional rainfall datasets (such as the Australian datasets AWAP and SILO), the selected tracks may extend beyond the area covered by the rainfall data. This script will only extract rainfall for instances when the location of the atmospheric low is within the latitudinal range of the rainfall grid.

Conversely, rainfall is still extracted when the location is outside the longitudinal range of the regional rainfall dataset (as long as the rainfall is within the chosen radius from the low).

5.3 Radial area limited to track point positions

The chosen radius (e.g. 500 km) is only applied to individual track point positions. Thus, some zones within that selected distance from the actual/estimated track (but outside that distance from all track point positions) were potentially excluded from the rainfall extraction. However, except for fast-moving ALPS, the omitted areas should be small when track point positions are available for six-hourly or shorter intervals.

6 Future updates

Future updates of the *rainfall_tracker* toolbox will depend on user feedback. Especially in terms of applicable “rain day” for GHCN-Daily station data, users are requested to provide information about cut-off periods for daily rainfall datasets of individual countries (either in local time or in UTC), along with relevant websites/articles that support their feedback. This information would help to expand on and modify the relatively small list of countries for which the code automatically adjusts for 24-hour periods that are not midnight to midnight (e.g. 9 a.m. cut-off for Australian sites). Note that, for some countries, multiple, station-specific cut-off periods may apply (Jaffrés and Gray, 2023).

The list of countries for which a shift has been automated are listed in Table 2. A midnight-to-midnight rainfall period is assumed for any countries not listed in Table 2.

Table 2: Reporting time for 24-hour station rainfall for country-specific data in the GHCN-Daily database. The corresponding country codes associated with GHCN-Daily are also listed, along with the source of the timing information. The final column displays the toolbox-applied shifts to TC track time data to align with the rain date (of the reporting time).

Country (GHCN-Daily code)	24-hour cut-off time[#]	Source	Applied shift to tracks (hours)*
Australia (AS)	9 a.m. local time	Australian Bureau of Meteorology	+14.99/24
China (CH)	8 p.m. local time	Yatagai et al. (2012)	+3.99/24
Fiji (FJ)	9 a.m. local time	"	+14.99/24
India (IN)	8:30 a.m. local time	"	+15.49/24
Mongolia (MG)	8 p.m. local time	"	+3.99/24

[#]The assumption is made that the date corresponding to the cut-off time is the date assigned to the rainfall (reporting day).

***Note:** This shift is in addition to any differences in applicable time zones between TC tracks (usually UTC+0) and GHCN-Daily station data (e.g. UTC+10 in northeastern Australia). Ensure you separate apply the most appropriate time zone before running the code, considering 1) the main impact zone, 2) potential historical changes to country zones and 3) whether daylight saving is applicable (seasonal changes).

7 Publications related to toolbox testing

Testing of all toolbox capabilities, including total event rainfall, peak rainfall intensity and raw (individual timestep), and daily vs subdaily properties for cyclones throughout the world (datasets: ERA5, GHCN-Daily and SILO):

> Jaffrés, J.B.D. and Gray, J.L., 2023. Chasing rainfall: estimating event precipitation along tracks of tropical cyclones via reanalysis data and in-situ gauges. C&R Consulting, Townsville.

Total event rainfall for atmospheric lows in southeast Australia (dataset: SILO):

> Gray, J.L., Verdon-Kidd, D.C., Jaffrés, J.B., Hewson, M.G., Clarke, J.M., Sharma, K.K. and English, N.B., 2023. Characterizing Australia's east coast cyclones (1950–2019). International Journal of Climatology.

Total event rainfall and peak one-hourly intensity for atmospheric lows in southeast Australia (datasets: ERA5, SILO):

> Gray, J.L., 2023. Assessing the spatial and temporal characteristics of cyclone tracks along the east coast of Australia. Unpublished PhD thesis. Central Queensland University, Townsville, Australia.

8 References

Eaton, J.W., Bateman, D., Hauberg, S. and Wehbring, R., 2017. GNU Octave version 4.2.1 manual: a high-level interactive language for numerical computation.

Jaffrés, J.B.D., 2019. GHCN-Daily: a treasure trove of climate data awaiting discovery. Computers & Geosciences, 122: 35-44.

Jaffrés, J.B.D. and Gray, J.L., 2023. Chasing rainfall: estimating event precipitation along tracks of tropical cyclones via reanalysis data and in-situ gauges, C&R Consulting, Townsville.

Menne, M.J., Durre, I., Vose, R.S., Gleason, B.E. and Houston, T.G., 2012. An overview of the global historical climatology network-daily database. Journal of Atmospheric and Oceanic Technology, 29(7): 897-910.

Yatagai, A., Kamiguchi, K., Arakawa, O., Hamada, A., Yasutomi, N. and Kitoh, A., 2012. APHRODITE: Constructing a long-term daily gridded precipitation dataset for Asia based on a dense network of rain gauges. Bulletin of the American Meteorological Society, 93(9): 1401-1415.