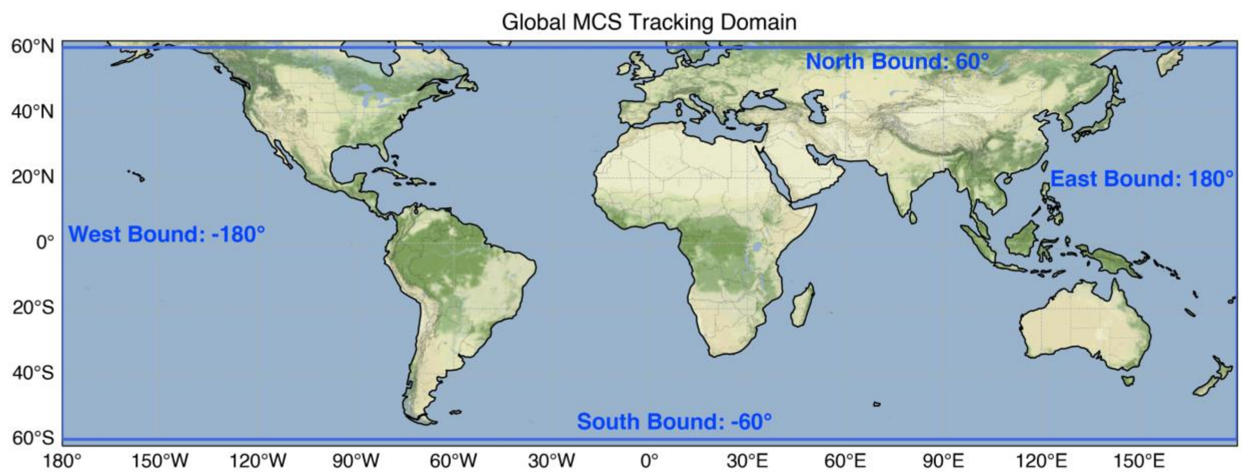
# Description of the PyFLEXTRKR MCS Tracking Dataset

**Prepared by Zhe Feng (zhe.feng@pnnl.gov)**
**Pacific Northwest National Laboratory**

## 1. Introduction

This documentation explains the Global MCS tracking dataset developed by Feng et al. (2021). The current v2 of the dataset is produced by the Python FLEXible object TRaKeR (PyFLEXTRKR) algorithm (Feng et al., 2023). The period is from June 2000 to December 2020. The geographic coverage is 180°W-180°E, 60°S-60°N.




## 2. Purpose

The purpose of PyFLETRKR is to provide automated identification and lifecycle evolution of individual MCS events, along with a suite of important MCS characteristics including lifetime, infrared brightness temperature ($T_b$) defined cloud shield (cold cloud cores and surrounding cold anvils) and surface precipitation feature characteristics, movement, and more. PyFLEXTRKR can be applied to observations and model simulations, facilitating their objective and statistical comparisons.

## 3. MCS Tracking Algorithm

The FLEXTRKR algorithm first identifies and tracks large cold cloud systems (CCSs) associated with deep convection using $T_b$ data, and subsequently identifies MCSs based on collocated precipitation feature (PF) characteristics. A CCS is defined using a detect-and-spread method, where contiguous regions of cold core are expanded to include surrounding cold anvils. An option to use PF to link CCS that share a large PF is provided. For more details of this method see Feng et al. (2018). An MCS is defined as a large CCS ($T_b < x$ K, area > $x$ km$^2$) containing a PF (contiguous area within a CCS with rain rate > $x$ mm h$^{-1}$) with major axis length > $x$ km, and PF area, PF mean rain rate, PF rain rate skewness, and heavy rain volume ratio exceeding user-defined lifetime-dependent thresholds and persists for at least $x$ hours. All user-defined parameters are set in the config file in yaml format. Most of the key parameters used to define MCS are also recorded in the global attribute of the MCS track statistics data.

## 4. MCS Data Description

### a) Hourly pixel-level data

The MCS tracking pixel-level data are produced on the native grid of the input dataset. The pixel-level data contains full field of $T_b$, precipitation, CCS cloud object identification, and MCS tracking number. The data file name follows this format "mcstrack_yyyymmdd_hhmm.nc".

**Figure 1** shows an example of the pixel-level MCS database over the Maritime Continent. IR brightness temperature (Tb) and precipitation are both full fields from the input dataset. The color shadings behind large clusters of PFs show the MCS masks for individual MCSs. The same color shadings between different times denote the same MCS being tracked.

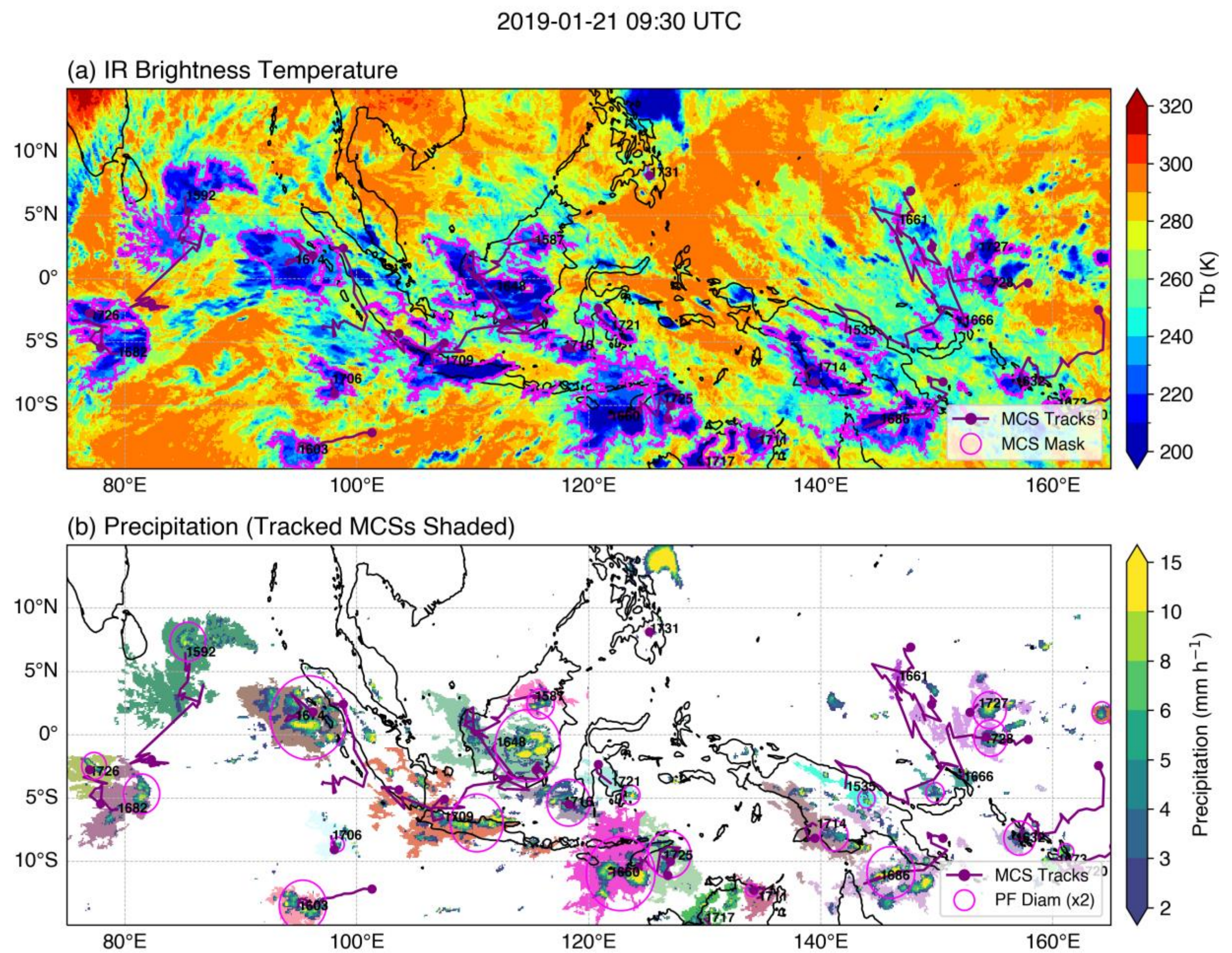



**Figure 1.** Example snapshot of pixel-level MCS database over the Maritime Continent during 21 January 2019. (a) IR brightness temperature, (b) precipitation. The magenta contour in (a) and the color shadings behind large clusters of PFs in (b) denote MCS masks ("*cloudtracknumber*"). The purple color lines are MCS tracks, the purple dots are MCS initiation locations, and the numbers are MCS track numbers. The

magenta circles in (b) show the largest PF equivalent diameter (multiplied by 2 for visibility) within an MCS. Animation for this day can be found in https://portal.nersc.gov/project/m1867/mcs_global_v2/MCS_pixeldata_example_animation.mp4.

**Table 1. MCS track pixel-level data variables and descriptions.**

| Variable Name | Dimension | Description |
|---|---|---|
| **time** | time | Epoch time |
| **base_time** | time | Epoch time (same as "time") |
| **lon** | lon | Longitude |
| **lat** | lat | Latitude |
| **longitude** | lat, lon | Longitude of each grid |
| **latitude** | lat, lon | Latitude of each grid |
| **tb** | time, lat, lon | Brightness temperature (full data) |
| **precipitation** | time, lat, lon | Precipitation (full data) |
| **cloudtracknumber** | time, lat, lon | MCS cloud track number mask |
| **pcptracknumber** | time, lat, lon | MCS PF track number mask. This track number is the same with "cloudtracknumber" except only applied to precipitation. |
| **cloudtracknumber_nomergesplit** | time, lat, lon | MCS cloud track number (exclude merge/split clouds) |
| **merge_tracknumbers** | time, lat, lon | Number of the MCS track that this cloud merges into |
| **split_tracknumbers** | time, lat, lon | Number of the MCS track that this cloud splits from |
| **cloudnumber** | time, lat, lon | Number associated with a CCS at a given pixel (not tracked) |
| **cloudtype** | time, lat, lon | Grid of cloud type classifications. 1 = core, 2 = cold anvil, 3 = warm anvil, 4 = other |

Explanation of each variable in the pixel-level data is provided in **Table 1.** Within each sub-directory of pixel-level data, denoted by the start and end date of a continuous tracking period (e.g., 20140101.0000_20140201.0000), the MCS track number ("*cloudtracknumber*", and "*pcptracknumber*") monotonically increases from 1 to X (the maximum number of MCSs tracked during a period). For example, *cloudtracknumber* = 1 within multiple pixel-level data files denotes the first MCS tracked during this period. The MCS track number in "*cloudtracknumber*" and "*pcptracknumber*" is the same, but they differ in areal coverage. "*cloudtracknumber*" is defined using $T_b$, so it has a larger coverage, usually containing most if not all of the precipitation in the vicinity of the MCS. "*pcptracknumber*" is defined using precipitation, so it has a smaller coverage, only containing the major precipitation features.

Note that the MCS track numbers all start from 1 for each individual tracking periods, i.e., they are not unique across different tracking periods. For example, there will be MCS track number = 1 in both tracking period 20140101.0000_20140201.000 and 20140201.0000_20140301.0000. Because each tracking period are tied to one track file (described below), which contains time and location of each MCSs, they can be distinguished across different tracking periods if one desired to do so. If there is not a single CCS identified in the entire domain at a given time, or there are significant areas of missing $T_b$ data (e.g., missing satellite scans), there will not be an MCS tracking pixel-level data for that time.

Sub-directory of pixel-level data: /mcstracking/yyyymmdd.hhmm_yyyymmdd.hhmm/

## b) Track statistics data

The track data contains many variables summarizing the key features (i.e., statistics) of MCSs. Each continuous tracking period correspond to one track file. The data file name follows this format "mcs_tracks_final_yyyymmdd.hhmm_yyyymmdd.hhmm.nc".

The track data describes the timing, location (e.g., "*meanlat*", "*meanlon*"), and evolution of each MCS during the tracking period. The variables are organized by [*tracks, times*]. The "*tracks*" dimension is the number of MCSs in the tracking period, and the tracks match the track number in the pixel-level data. For example, *tracks* = 0 is the first MCS in the pixel-level data (*cloudtracknumber* = 1, or *pcptracknumber* = 1), and *tracks* = 1 is the second MCS (*cloudtracknumber* = 2, or *pcptracknumber* = 2), …, etc. The "*times*" dimension is the relative time since an MCS is detected. For example, *times* = 0 is the convection initiation time (defined by the first detection of a CCS from $T_b$ data) of an MCS. A variable named "*base_time*" (Epoch time in seconds since 1970-01-01T00:00:00) with dimension [*tracks, times*] designates the time for each MCS track at each time. Combining "*base_time*", which locates the time of an MCS, and the track index, which corresponds to the "*cloudtracknumber*" mask in the pixel-level data described above, one can link the MCS track data with the exact pixel-level location of the MCS at a given time.

For MCS characteristics, there are generally two types of variables, one derived from IR $T_b$ data, which has two dimensions [*tracks, times*]; and another type derived from precipitation data, which has three dimensions [*tracks, times, nmaxpf*].

Explanation of each variable in the MCS track statistics data is provided in **Table 2**.

**Table 2. MCS track statistics hourly data variables and descriptions.**

| Variable Name | Dimension | Description |
|---|---|---|
| **track_duration** | tracks | Duration of each tracked MCS defined by CCS. This is the longest lifetime of MCS, potentially including times before and after precipitation. Multiply "track_duration" by "time_resolution_hour" (in global attribute) to convert to physical lifetime units. |
| **mcs_duration** | tracks | Duration of MCS stage (CCS > MCS area threshold). Multiply by "time_resolution_hour" (in global attribute) to convert to physical lifetime units. |
| **start_basetime** | tracks | Start Epoch time of each track |
| **end_basetime** | tracks | End Epoch time of each track |
| **start_split_cloudnumber** | tracks | Cloud number where this track splits from |
| **end_merge_cloudnumber** | tracks | Cloud number where this track merges with |

| | | |
|---|---|---|
| **start_status** | tracks | Flag indicating how the track starts |
| **end_status** | tracks | Flag indicating how the track ends |
| **track_status** | tracks, times | Flag indicating evolution / behavior a track. See **Table 3** for detailed descriptions. |
| **base_time** | tracks, times | Epoch time of each MCS in a track |
| **mcs_status** | tracks, times | Flag indicating the status of MCS based on Tb. 1 = Yes, 0 = No. |
| **meanlat** | tracks, times | Mean latitude of MCS defined using CCS area |
| **meanlon** | tracks, times | Mean longitude of MCS defined using CCS area |
| **core_area** | tracks, times | Area of cold cloud core |
| **cold_area** | tracks, times | Area of cold anvil |
| **ccs_area** | tracks, times | Area of cold cloud shield (CCS) |
| **corecold_mintb** | tracks, times | Minimum $T_b$ in core + cold anvil area |
| **corecold_meantb** | tracks, times | Mean $T_b$ in core + cold anvil area |
| **core_meantb** | tracks, times | Mean $T_b$ in core area |
| **cloudnumber** | tracks, times | Cloud number in the corresponding cloudid file |
| **merge_cloudnumber** | tracks, times, mergers | Cloud numbers that merge into MCS |
| **split_cloudnumber** | tracks, times, mergers | Cloud numbers that split from MCS |
| **merge_ccs_area** | tracks, times, mergers | Cold cloud shield area that merge into MCS |
| **split_ccs_area** | tracks, times, mergers | Cold cloud shield area that split from MCS |
| **Precipitation Feature (PF) Statistics** | | |
| **pf_lifetime** | tracks | MCS lifetime when a significant PF is present, which can be considered as the active period of the convection. |
| **pf_mcsstatus** | tracks, times | Flag indicating the status of MCS based on PF. 1 = Yes, 0 = No |
| **total_rain** | tracks, times | Total precipitation under CCS. No cosine latitude weighting is applied. |
| **total_heavyrain** | tracks, times | Total heavy precipitation under CCS. No cosine latitude weighting is applied. |
| **rainrate_heavyrain** | tracks, times | Mean heavy rain rate under CCS. No cosine latitude weighting is applied. |
| **pf_npf** | tracks, times | Number of PF under the CCS |
| **pf_landfrac** | tracks, times | Fraction of PF over land |
| **pf_area** | tracks, times, nmaxpf | PF area under CCS |
| **pf_lon** | tracks, times, nmaxpf | Mean longitude of each PF |
| **pf_lat** | tracks, times, nmaxpf | Mean latitude of each PF |
| **pf_lon_centroid** | tracks, times, nmaxpf | Centroid longitude of PF |
| **pf_lat_centroid** | tracks, times, nmaxpf | Centroid latitude of PF |
| **pf_lon_weightedcentroid** | tracks, times, nmaxpf | Rain rate weighted centroid longitude of PF |
| **pf_lat_weightedcentroid** | tracks, times, nmaxpf | Rain rate weighted centroid latitude of PF |
| **pf_rainrate** | tracks, times, nmaxpf | Mean precipitation rate of each PF |
| **pf_maxrainrate** | tracks, times, nmaxpf | Maximum precipitation rate of each PF |
| **pf_accumrain** | tracks, times, nmaxpf | Area total precipitation rate of each PF |
| **pf_accumrainheavy** | tracks, times, nmaxpf | Area total heavy precipitation of each PF |
| **pf_skewness** | tracks, times, nmaxpf | PF pixel-level rain rate skewness |
| **pf_majoraxis** | tracks, times, nmaxpf | PF major axis length |

| | | |
|---|---|---|
| **pf_minoraxis** | tracks, times, nmaxpf | PF minor axis length |
| **pf_aspecratio** | tracks, times, nmaxpf | PF aspect ratio (major axis length / minor axis length) |
| **pf_orientation** | tracks, times, nmaxpf | Orientation of major axis of PF |
| **pf_eccentricity** | tracks, times, nmaxpf | Eccentricity of PF |
| **Movement** | | |
| **movement_distance** | tracks, times | Movement distance along angle movement_theta |
| **movement_speed** | tracks, times | Movement speed along angle movement_theta |
| **movement_theta** | tracks, times | Movement direction<br>(Caution using this variable! Direction is not defined the same as traditional meteorology wind direction. Use *movement_distance_x* and *movement_distance_y* to compute direction instead.) |
| **movement_distance_x** | tracks, times | East-West component of movement distance |
| **movement_distance_y** | tracks, times | North-South component of movement distance |

**Table 3. Track status description.**

| Status | Description |
|---|---|
| 0 | Track stops |
| 1 | Simple track continuation |
| 2 | This is the bigger cloud in simple merger |
| 3 | This is the bigger cloud from a simple split that stops at this time |
| 4 | This is the bigger cloud from a split and this cloud continues to the next time |
| 5 | This is the bigger cloud from a split that subsequently is the big cloud in a merger |
| 13 | This cloud splits at the next time step |
| 15 | This cloud is the bigger cloud in a merge that then splits at the next time step |
| 16 | This is the bigger cloud in a split that then splits at the next time step |
| 18 | Merge-split at same time (big merge, splitter, and big split) |
| 21 | This is the smaller cloud in a simple merger. |
| 24 | This is the bigger cloud of a split that is then the small cloud in a merger |
| 31 | This is the smaller cloud in a simple split that stops |
| 32 | This is a small split that continues onto the next time step |
| 33 | This is a small split that then is the bigger cloud in a merger |
| 34 | This is the small cloud in a merger that then splits at the next time step |
| 37 | Merge-split at same time (small merge, splitter, big split) |
| 44 | This is the smaller cloud in a split that is smaller cloud in a merger at the next time step |
| 46 | Merge-split at same time (big merge, splitter, small split) |
| 52 | This is the smaller cloud in a split that is smaller cloud in a merger at the next time step |
| 65 | Merge-split at same time (smaller merge, splitter, small split) |

**Two methods to separate tracks that initiate *naturally* from *splitting*, or tracks that end *naturally* from merging:**

Initiate as split or “naturally”:

1. *start_split_cloudnumber* > 0 means track <u>start as a split</u>
2. *start_status* $\neq$ 1 or 2 means track start as a split

Possible values for *start_status*: 1, 2, 13, 15, 32, 33, 44, 46.

Possible *start_status* values when *start_split_cloudnumber* > 0 (split) are: 31, 32, 33, 44, 46, 52, 65.

End as merger or "naturally":

1. *end_merge_cloudnumber* > 0 means track end as a merger
2. *end_status* ≠ 0 or 3 means track end as a merger

Possible values for *end_status*: 0, 3, 21, 24, 34, 37.
Possible *end_status* values when *end_merge_cloudnumber* > 0 (merger) are: 21, 24, 34, 37, 52, 65.

The variables derived from precipitation data contains several (defined by "*nmaxpf*" in the config file) of the largest Precipitation Features (PFs) within each CCS. The "*nmaxpf*" dimension is organized by size, e.g., *nmaxpf* = 0 contains the largest PF at any given time associated with an MCS.

Movement of MCS PFs are estimated using 2-D cross-correlation map between user-defined time-steps ("*lag_for_speed*") of the MCS PFs. The detail description of the method is provided in Feng et al. (2018).

Sub-directory of track statistics data:
/stats/mcs_tracks_final_extc_yyyymmdd.hhmm_yyyymmdd.hhmm.nc

### c) Linking track statistics data with pixel-level data

Here is an example usage case to link the track statistics data with pixel-level data: where exactly is the MCS feature mask for track #2 at time = 2010-07-01 03:00 UTC? **Figure 2** shows an example on how to link the pixel-level MCS mask with the track statistics data. For a given pixel-level file, the MCS mask values in the variable "*cloudtracknumber*" correspond to the "tracks" dimension in the MCS track statistics file (offset by 1 for 0-based indexing). **Figure 2a** shows two MCSs (track # 2, 3) coexist at 2010-07-01 03:00 UTC. These two MCSs correspond to the "tracks" indices (1, 2) and "times" indices (0, 1) in the track statistics file, because the values in the track statistics "base_time" variable (Epoch time) matches the pixel-level file time 2010-07-01 03:00 UTC.

In practice, since both sets of pixel-level and track statistics files contain Epoch time ("base_time"), which contains the full date/time (e.g., 2010-07-01T03:00:00), one can simply search for the matching "base_time" between track statistics file and a specific pixel-level file to find all the indices of MCS tracks that exist in that hour.

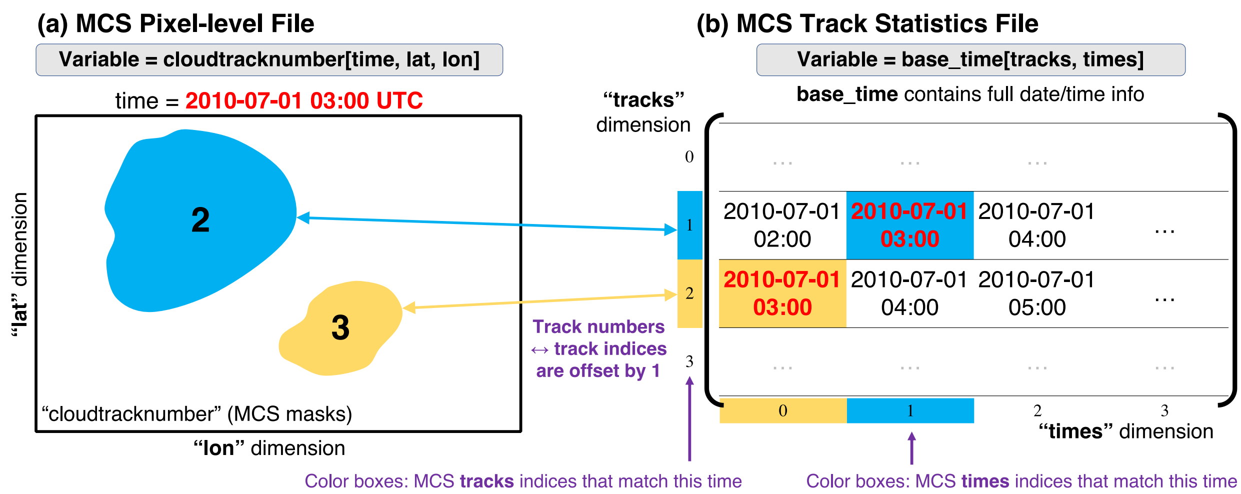



**Figure 2.** An example to link a track statistics variable to a pixel-level file. (a) MCS pixel-level file, showing variable "*cloudtracknumber*" containing MCS masks at 2010-07-01 03:00 UTC, (b) MCS track statistics file, showing variable "base_time" containing the Epoch time of each MCS. The MCS "*cloudtracknumber*" values correspond to the "tracks" indices (offset by 1 for 0-based indexing). For each matching MCS track, the matching time index can be located by matching the "base_time" value (Epoch time) with the pixel-level file "time" (Epoch time), shown in red colors (2010-07-01 03:00).

## 5. Analysis codes

Post-processing and analysis Python Notebooks are available on GitHub: [https://github.com/FlexTRKR/PyFLEXTRKR/tree/main/Notebooks](https://github.com/FlexTRKR/PyFLEXTRKR/tree/main/Notebooks)

**References**

Feng, Z., Hardin, J., Barnes, H. C., Li, J., Leung, L. R., Varble, A., & Zhang, Z. (2023). PyFLEXTRKR: a flexible feature tracking Python software for convective cloud analysis. *Geosci. Model Dev., 16*(10), 2753-2776. https://doi.org/10.5194/gmd-16-2753-2023

Feng, Z., Leung, L. R., Houze, R. A., Hagos, S., Hardin, J., Yang, Q., et al. (2018). Structure and Evolution of Mesoscale Convective Systems: Sensitivity to Cloud Microphysics in Convection-Permitting Simulations Over the United States. *Journal of Advances in Modeling Earth Systems, 10*(7), 1470-1494. https://doi.org/10.1029/2018MS001305

Feng, Z., Leung, L. R., Liu, N., Wang, J., Houze, R. A., Li, J., et al. (2021). A Global High‐Resolution Mesoscale Convective System Database Using Satellite‐Derived Cloud Tops, Surface Precipitation, and Tracking. *Journal of Geophysical Research: Atmospheres, 126*(8). https://doi.org/10.1029/2020JD034202