

Authors' response to the discussion of “tobac v1.0: towards a flexible framework for tracking and analysis of clouds in diverse datasets” by Max Heikenfeld et al.

We thank the two anonymous reviewers for their feedback and detailed comments on the submitted manuscript. Their questions and detailed input have helped to improve the revised manuscript.

We give a short overview of the main points addressed by both reviewers here and some minor additional improvements we made during the revision of the manuscript. Detailed responses to the individual comments by the two reviewers can be found below, along with a version of the manuscript with tracked changes compared to the initial submission.

- We have added an additional figure (Fig. 3) depicting the trajectory linking step and improved the associated description in the software description sections.
- We have improved the description of the feature identification and segmentation steps to make it clearer how these two steps are different and how they can be used together in tobac.
- We have added references discussing additional aspects of the existing literature about cloud tracking approaches (see details in the responses to the individual reviews).
- We have amended the description of the simulation and satellite data used in the two examples (Sec. 3 and 4) for increased reproducibility of the study.
- We have included input from the reviewers as well as some minor enhancements of the package into a version tobac 1.2, which we mention in the code availability section and which leads to a change in the title of the manuscript.

Version 1.2 includes the following changes:

- We now allow for a use of tobac with both Python 2 and Python 3 which allows interoperability with existing software in Python 2.
- We have added additional automated tests to the package
- There have been substantial additions to the documentation of the package
- tobac can now be installed directly via the conda package manager as it has been included in the conda-forge channel. We have added information about that to the code availability section.
- The comment by Reviewer 1 about the performance of the code has lead us to more detailed profiling of the software performance. We found and removed some additional bottlenecks, especially in the feature detection code, that improved the performance of the software significantly. We have added the enhancements to the new version of the software. The changes only affect the processing time and not change the analysis results at all, as they are simply related to the code structure, i.e. improving the way variables are created and values assigned.

Interactive comment on “tobac v1.0: towards a flexible framework for tracking and analysis of clouds in diverse datasets”

by Max Heikenfeld et al.

Anonymous Referee #1

Received and published: 25 June 2019

We would like to thank the reviewer for their comments and suggestions. The review pointed out important aspects that we have clarified or addressed by minor changes in the manuscript. In the following, we respond to the reviewer’s comments in black, with our answers to the comments in blue and the adapted text from the revised manuscript in green. We have attached the revised version of the manuscript with tracked changes to the general authors' response. In the general authors’ response (AR), we have added a few additional comments regarding the revised manuscript and points raised by both reviewers.

This paper present a tracking code for clouds and other atmospheric objects, using modern programming techniques and languages. The topic is certainly not new, but the fact that the code is developed to be flexible and openly available warrants a code description paper. I find the paper clearly written, appreciate the examples, and recommend publication after the following minor questions are answered:

1) Was it a conscious decision to not tack on “. . . and COnvection” to the acronym?

We actually played with the idea of that rather obvious acronym. But apart from avoiding possible negative associations with the health impacts of the associated substance, we decided to stick to a unique name for the package to make it easier in terms of package managers, documentation, searchability, etc. and thus opted for the acronym *tobac*.

2) What is the performance of this algorithm? For example, for each of the test cases, what was the maximum memory footprint, runtime of the script, etc.

We want to thank the reviewer specifically for this comment on a very important aspect of the software package, that we had not addressed quantitatively in the initial version of the manuscript. In fact, creating the detailed performance analyses for the example cases and some puzzling results lead us to more detailed profiling that, especially in the code for the feature detection, allowed to reduce the processing time by more than an order of magnitude. These changes were related to time-consuming operations related to creating specific data structures or writing values into these data structures and do not change the analysis results at all. As part of the testing and the improvements, we have modularised the code for the feature detection and segmentation into several sub-functions, which makes the code more accessible and also more suitable for future adaptations and additions.

We have added information about the typical performance for the examples in the manuscript text, taking both calculations on a typical laptop and on a dedicated data analysis facility (JASMIN) into account:

“The data processing has been performed on the JASMIN data analysis facility (CEDA, 2019). The script including feature detection, segmentation, trajectory linking and saving of the analysis output for the 1-min data output had a processing time of around 17 minutes with a maximum memory footprint of 3.1 GB using a maximum of 3 processes and 27

threads. The segmentation step has been broken up into chunks of 10 minutes each to limit the total memory consumption of the analysis. The processing time is almost entirely taken up by the segmentation step using time-resolved 3-dimensional data of the total condensate. It is thus strongly affected by the time required to access the data on the disk and thus highly dependent on both the infrastructure and the structure of the data file, i.e. the data compression in the input files.”

(page 14, line 10)

“The data processing was performed on the JASMIN data analysis facility (CEDA, 2019). The total processing time of the script including feature detection, segmentation trajectory linking and saving the output data was around 1 minute with a maximum memory footprint of around 500 MB for the model data and 2.5 minutes with a memory footprint of around 400 MB for the satellite data, each using a maximum of 3 processes and 27 threads.”

(page 18, line 28)

3) It makes sense to separate the several stages of the workflow from a coding point of view; is it also possible to save data after the trajectory linking, and if so, in what format and what kind of design?

The individual stages of the workflow are separated as discussed in the “software description” section. Intermediate results can be saved after the initial steps in the format of *pandas* DataFrames for features and cloud tracks or as *iris* cubes in the case of the cloud masks resulting from the segmentation step. We have added that explicit information in the “software description” section of the manuscript:

“The intermediate results of each individual analysis step can be conveniently saved and examined in the form of *pandas* DataFrames or *iris* cubes.”

(page 6, line 23)

4) P6: Do I understand correctly that the feature detection only works in 2D? Dawe and Austin (and others) did 3D feature detection, which ends up being more precise, especially also in the tracking part.

The feature detection we have currently implemented only works in 2D. This was a reasonable choice, especially with regard to the deep convective clouds we mainly focused on here. As you mention, there will be situations such as the analyses of shallow convection you referred to, where 3D identification and 3D tracking will be beneficial. We have concrete plans to include 3D feature detection and tracking in the next major version of *tobac*.

5) P9, I20: A circular range parameter v_max suggests that the trajectory linking is performed isotropically. Is it possible to assign/automatically retrieve an advection velocity as well? Heiblum et al did this very effectively.

The advection velocity of the trajectories in previous time steps is actually used as part of the tracking using *trackpy*. Thus, v_max only describes the size of a circle around the predicted position based on the advection in the previous steps, i.e. the allowed deviation of from the predicted position using this information from the previous time steps. We have adapted the manuscript in this section to make this clearer and added a schematic depiction of a specific linking step as Figure 3.

“The linking determines which of the features detected in a specific time step (see Sect. 2.2) is identical to an existing feature in the previous time step and is illustrated in Fig. 3. For each existing feature, the movement within a time step is predicted based on the velocities in a number of previous time steps. The algorithm then breaks the search process down to candidate features by restricting the search to a circular search region centred around the predicted position of the feature in the next time step. For newly initialised trajectories, where no velocity from previous time steps is available, the algorithm resorts to the average velocity of the nearest tracked objects. The parameter v_{max} restricts how much the future position of a feature is allowed to deviate from a linear extrapolation of the trajectory over time. It thus has the units of a velocity and describes the dependency of the circular search range d on the output time step Δt in the data used for the tracking $d = v_{max} \Delta t$.”

(page 9, line 31)

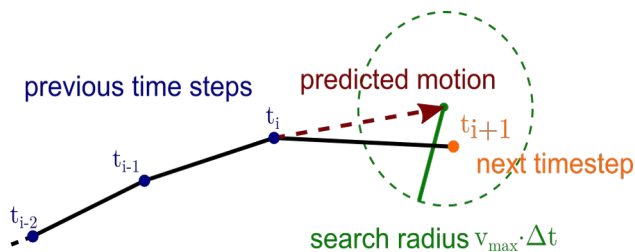


Figure 3. Schematic illustration of the trajectory linking with the predicted motion of the feature based on previous time steps and a search range around the predicted position.

(page 10)

Interactive comment on “tobac v1.0: towards a flexible framework for tracking and analysis of clouds in diverse datasets”

by Max Heikenfeld et al.

Authors Comments:

We would like to thank the reviewer for their comments and suggestions as well as the motivating positive feedback on the manuscript and the software. The review pointed out important aspects that required additional clarity or information that we have clarified or addressed by changes in the manuscript. In the following, we respond to the reviewer's comments in black, with our answers to the comments in blue and the adapted text from the revised manuscript in green. We have attached the revised version of the manuscript with tracked changes to the general authors' response. In the general authors' response (AR), we have added a few additional comments regarding the revised manuscript and points raised by both reviewers.

Anonymous Referee #2

Received and published: 4 July 2019

1 Reviewer's summary of the manuscript

In “tobac v1.0: towards a flexible framework for tracking and analysis of clouds in diverse datasets,” Heikenfeld et al. present a new, open source software framework for detecting and tracking clouds in datasets ranging from satellite observations to model output. They describe the details and justification for their Python-based implementation, which extensively leverages existing libraries that offer a number of benefits (e.g., lazy data loading from dask), and they describe in detail the various algorithmic steps of tobac. They further provide two illustrating examples of how tobac might be used: tracking of deep convective clouds in 3D CRM output and tracking/comparison of clouds in 2D satellite data and 2D model output. The authors also discuss some limitations of tobac—especially restrictions on the appropriate temporal resolution of the input data, and they describe how the modular design of tobac will facilitate the authors (and/or future users) to easily improve/upgrade various algorithmic components.

2 Summary of Review

Overall, Heikenfeld et al. present a well-written and thorough description of a new, open-source framework that other geoscientists will likely find useful. The paper is well within the scope of GMD. The author's algorithmic choices are both well-explained and justified, and the methods by which the authors evaluate tobac are sound: particularly with respect to evaluation of the impact of temporal resolution. I ordinarily have many paragraphs of feedback when reviewing papers, but in this case, the authors have put forth a solid manuscript; I do not have much feedback to offer, beyond a suggestions for clarifying in a few places. Based on this assessment, I recommend that the manuscript be accepted for publication in GMD pending some minor revisions. The authors should be commended on assembling an excellent manuscript and describing a useful open-source code.

3 General Comments

3.1 Some additional citations and discussion

In section 1, the authors give a reasonably thorough overview of other examples of cloud tracking in the literature. There are a few additional references that I would suggest the authors discuss:

- Wilcox, Eric M. “Spatial and Temporal Scales of Precipitating Tropical Cloud Systems in Satellite Imagery and the NCAR CCM3.” *Journal of Climate* 16, no. 22 (November 1, 2003): 3545–59. [https://doi.org/10.1175/1520-0442\(2003\)016<3545:SATSOP>2.0.CO;2](https://doi.org/10.1175/1520-0442(2003)016<3545:SATSOP>2.0.CO;2).

Note that the following papers focus on identifying clouds in time-slices, and not necessarily tracking them in time. But the concept is close enough—the papers are essentially steps 1–3 in the tobac workflow (Figure 1)—that these papers warrant discussion.

- Wilcox, Eric M, and V Ramanathan. “Scale Dependence of the Thermodynamic Forcing of Tropical Monsoon Clouds: Results from TRMM Observations.” *Journal of Climate* 14, no. 7 (April 1, 2001): 1511–24. [https://doi.org/10.1175/1520-0442\(2001\)014<1511:SDOTTF>2.0.CO;2](https://doi.org/10.1175/1520-0442(2001)014<1511:SDOTTF>2.0.CO;2).
- Wood, Robert, and Paul R. Field. “The Distribution of Cloud Horizontal Sizes.” *Journal of Climate* 24, no.18 (2011): 4800–4816. <https://doi.org/10.1175/2011JCLI4056.1>.
- O’Brien, Travis A, Fuyu Li, William D Collins, Sara A Rauscher, Todd D Ringler, Mark Taylor, Samson M Hagos, and L Ruby Leung. “Observed Scaling in Clouds and Precipitation and Scale Incognizance in Regional to Global Atmospheric Models.” *Journal of Climate* 26, no. 23 (December 2013): 9313–33. <https://doi.org/10.1175/JCLI-D-13-00005.1>.
- Igel, Matthew R., Aryeh J. Drager, and Susan C. van den Heever. “A Cloud-Sat Cloud Object Partitioning Technique and Assessment and Integration of Deep Convective Anvil Sensitivities to Sea Surface Temperature.” *Journal of Geophysical Research: Atmospheres* 119, no. 17 (2014): 10515–35. <https://doi.org/10.1002/2014JD021717>.
- Guillaume, A., B. H. Kahn, Q. Yue, E. J. Fetzer, S. Wong, G. J. Manion, H. Hua, and B. D. Wilson. “Horizontal and Vertical Scaling of Cloud Geometry Inferred from CloudSat Data.” *Journal of the Atmospheric Sciences* 75, no. 7 (July 2018): 2187–97. <https://doi.org/10.1175/JAS-D-17-0111.1>.

As a side note, I am a bit surprised that one of these references wasn’t already included, as it was written by one of the co-author’s former Ph.D. students.

We want to thank the reviewer for pointing us to these two aspects of the existing literature that had not been covered in the introduction section in the initial version of the manuscript. We have added a discussion of the investigation of cloud size statistics and the use of active sensors:

“The identification of individual cloud objects in satellite data is one aspect of cloud tracking and has been used to investigate the spatial scaling of clouds on a global scale (Wilcox and Ramanathan, 2001; Wood and Field, 2011), including studies on the representation of these distributions in global atmospheric models (Wilcox, 2003; O’Brien et al., 2013). The use of satellite data from active sensors (Nesbitt et al., 2000; Bacmeister and Stephens, 2011; Riley et al., 2011; Igel et al., 2014; Guillaume et al., 2018) allows to

include information about the vertical extent of identified cloud objects, which provides improved classification of cloud types and understanding of important physical processes.” (page 2, line 30)

This includes the following two additional references in addition to the ones suggested by the reviewer:

Nesbitt, Stephen W., Edward J. Zipser, and Daniel J. Cecil. 2000. ‘A Census of Precipitation Features in the Tropics Using TRMM: Radar, Ice Scattering, and Lightning Observations’. *Journal of Climate* 13 (23): 4087–4106. [https://doi.org/10.1175/1520-0442\(2000\)013<4087:ACOPFI>2.0.CO;2](https://doi.org/10.1175/1520-0442(2000)013<4087:ACOPFI>2.0.CO;2).

Bacmeister, J. T., and G. L. Stephens. 2011. ‘Spatial Statistics of Likely Convective Clouds in CloudSat Data’. *Journal of Geophysical Research: Atmospheres* 116 (D4). <https://doi.org/10.1029/2010JD014444>.

In addition to these additional references, it might be useful to add a paragraph or two that describes what—in terms of science—has been learned by using cloud tracking. For readers unfamiliar with cloud tracking, this might help justify the scientific motivation for a flexible and open-source software package for cloud tracking.

We have amended the paragraph to add some additional summary of the scientific understanding gained from the use of cloud tracking:

“This overview clearly shows the wide range of extensive efforts that went into the development of elaborate software and analysis tools to track clouds in different types of datasets. The application of cloud identification and tracking and related techniques has substantially increased our understanding of cloud size distributions, the time evolution of different types of clouds, and the underlying physical processes governing cloud formation, development, and propagation. However, the overview also highlights the problem of limited compatibility between the different existing approaches and implementations, especially regarding the intended use of tracking clouds based on different data sources using the same algorithms and analysis tools.” (page 3, line 27)

3.2 Feature ID vs Segmentation

Perhaps I’m being a bit dense about this, but I re-read both sections 2.2 and 2.3 several times and could not determine the functional difference between feature identification and segmentation. Based on what I’m reading, it sounds like the only output of the feature identification step is the set of feature positions (weighted mean centers), from which the segmentation starts. Is this understanding correct? If so, this point should be emphasized, and if not, the text would benefit from a revision to make the distinction between feature identification and segmentation more clear. In later parts of the manuscript, it seems that one functional difference might be that feature identification and segmentation might use different variables: e.g., max vertical velocity vs cloud condensate mixing ratio.

The reviewer has understood the procedure the right way regarding feature detection and segmentation. In the current implementation of `tobac`, the positions of the detected features are the only information passed on to the next steps segmentation and trajectory linking.

Separating the two steps allows for the use of different datasets as shown in Sect. 3. In addition, the separation of the two steps makes sense in the case of three-dimensional input data, as the tracking is currently restricted to two-dimensional input.

We have worked over the description to make this clearer in the revised manuscript, e.g. at the end of the description of the segmentation step:

“Similarities between the feature detection and segmentation steps mean that these steps could be combined in some implementations in future versions of `tobac`, e.g. for applications based on a single input dataset as the one used in Sect. 4. However, treating the two analysis steps separately allows for the combination of different datasets as shown in Sect. 3.”

(page 9, line 21)

Would the authors get the same result if they segmented the condensate field, based on any values above the segmentation threshold, and then filtered out objects with max vertical velocity below a certain threshold?

I'm not suggesting that this should be done, but rather I am illustrating that it might be useful to discuss the feature-ID/segmentation approach versus other plausible approaches; such a comparison might help readers grasp what appears (to me) to be a subtle distinction.

We have not tried this specific order of operations suggested by the reviewer. However, tests of using just the condensate field for detection of individual clouds were not really successful in the case of tracking deep convective clouds. This is related to the fact that the condensate field is substantially less confined compared to the comparatively well defined narrow updraft regions and also much more interconnected between neighbouring cloud features (e.g., via anvils or cold pool boundaries).

3.3 Style of package/software names

The manuscript contains a lot of references to software libraries, e.g.,: `tobac`, `pandas`, `xarray`, `scipy`, etc.. It is good, and useful, that the authors do this, but it might be useful to use a different text style for these software package names in order to visually distinguish package names from English words. This is important to make the manuscript accessible to readers who might not be familiar with the Python ecosystem of packages, and it is especially important for software packages that could easily be confused for English words. For example, on pg 10, line 8, the sentence “The trajectories are recorded in a `pandas DataFrames`” might trip-up a non-Python-initiated reader (or at least amuse them).

If the authors used LaTeX to compose the manuscript, and if GMD style rules allow it, I would strongly suggest that the authors use the `\verb:package-name:` macro.

We have opted for typewriter-style font for the names of Python packages such as `Iris`, `xarray` or `dask` to make these manuscript clearer and easier to read for readers not familiar with the specific Python packages names.

3.4 Nice code

This is a compliment rather than constructive feedback: I appreciate the pervasive use of comments in the code and the clear and consistent documentation of functions. It was easy to skim through the code and get a general understanding of how the code is functioning.

Minor issues

p 6, line 25-26: The use of the term ‘erosion’ here is a bit unclear: the use of this word, in this sense, is not common in the geophysical sciences. The term ‘erosion’ should be explained here.

We have extended the description for the use of the erosion techniques from image processing in more detail and explained how they are used in the context of the tobac feature detection:

“The detection of regions above a specific threshold can lead to large interconnected regions combining several features linked by narrow ridges. To prevent this and identify these interconnected features separately, the tobac feature detection allows for the use of "erosion" techniques based on the implementation in `skimage.morphology.binary_erosion`. These techniques shrink the identified regions from the edges by a specific length or number of pixels, thus removing the connecting ridges between interconnected features. This has been shown to lead to more robust detection of individual features, as described in detail in Senf et al. (2018).”
(page 6, line 31)

In a similar way we have also added a more detailed definition of ‘watershedding’, which could lead to misunderstandings with readers unfamiliar with image processing techniques:

“Watershedding segmentation treats the input field as a topographic map and separates the input into individual regions similar to individual watersheds or catchment basins along a dividing ridge in a geological context (Meyer, 1994). These techniques are widely used in several existing cloud tracking and analysis algorithms described in Sect. 1, such as Heiblum et al. (2016a), Fiolleau and Roca (2013) and Senf et al. (2018).”
(page 9, line 7)

p 7, lines 6-16: this multithreshold approach is interesting. I don’t think I’ve seen this in the literature before - do the authors have a reference for this, or is this an innovation of this study? It should be stated either way. Also, it might be useful to state whether the framework is flexible enough to permit a single threshold value, which might be appropriate in cases where a person wants to track any contiguous feature with non-zero cloud condensate.

We have not based our approach on any existing literature in the field of cloud tracking. The multi-threshold methodology was developed as an extension of working with a single threshold, which is still perfectly possible, so we have stated this explicitly in the revised manuscript. There are however other approaches using an iterative set of threshold values for cloud detection in a different way along with examples from another field of science with similar challenges, such as astronomy. We have thus added references to these other types of multi-threshold feature detection methods:

“While using multiple thresholds is usually beneficial, feature detection using a single threshold value is possible in tobac by only supplying a single threshold value and can be appropriate in certain applications. An iterative set of threshold values was used in other recent approaches to cloud detection (Liang et al., 2017; Fu et al., 2019), however with a specific focus on the application to certain types of satellite images. Multiple-threshold methods are also applied in other fields of science facing similar challenges in feature detection such as astronomy (Zheng et al., 2015).”

(page 7, line 27)

Liang, Kuai, Hanqing Shi, Pinglv Yang, and Xiaoran Zhao. 2017. ‘An Integrated Convective Cloud Detection Method Using FY-2 VISSR Data’. *Atmosphere* 8 (2): 42.
<https://doi.org/10.3390/atmos8020042>.

Fu, Hualian, Yuan Shen, Jun Liu, Guangjun He, Jinsong Chen, Ping Liu, Jing Qian, and Jun Li. 2019. ‘Cloud Detection for FY Meteorology Satellite Based on Ensemble Thresholds and Random Forests Approach’. *Remote Sensing* 11 (1): 44.
<https://doi.org/10.3390/rs11010044>.

Zheng, Caixia, Jesus Pulido, Paul Thorman, and Bernd Hamann. 2015. ‘An Improved Method for Object Detection in Astronomical Images’. *Monthly Notices of the Royal Astronomical Society* 451 (4): 4445–59.
<https://doi.org/10.1093/mnras/stv1237>.

p10 lines 2-5: “The trajectories..” ← it isn’t clear to me what this sentence means. It might need to be rephrased.

We have rewritten the sentence more clearly and added some additional motivation for the extrapolation of trajectories:

“The feature detection step can often omit the initial or final stages of the evolution of a cloud due to the choice of specific thresholds. Thus, trajectories can also be extrapolated to additional output time steps at the start and at the end of the tracked path. This allows for the inclusion of both the initiation of the cell and the decaying later stages in the analysis of the cloud life cycle.”

(page 10, line 21)

p 10, lines 12-13: “Instead, the algorithm. . .” it might be useful to produce an illustration of this. I’m finding that I’m unable to visualize how cloud trajectories from tobac would appear when there are cloud splits/mergers

We have extended the description of how the current version of tobac acts in situations of cloud splits/mergers:

“The current implementation of the linking step does not include an explicit treatment of the splitting and merging of clouds, as implemented in several of the cloud tracking algorithms reviewed earlier (Dawe and Austin, 2012; Heus and Seifert, 2013; Heiblum et al., 2016a). Instead, the current version of tobac will create a continuous track with only one of the two separate cloud objects that combine in a merger or evolve from splitting of a tracked object, mostly based on which of these has the more similar direction of travel to the joint object. However, we have structured the implementation of tobac in a way that allows for the future addition of more complex tracking methods that can record a more complex network of relationships between cloud objects at different points in time.”
(page 11, line 5)

tobac ~~v1.0~~1.2: towards a flexible framework for tracking and analysis of clouds in diverse datasets

Max Heikenfeld¹, Peter J. Marinescu², Matthew Christensen¹, Duncan Watson-Parris¹, Fabian Senf³, Susan C. van den Heever², and Philip Stier¹

¹Atmospheric, Oceanic & Planetary Physics, Department of Physics, University of Oxford, Oxford, United Kingdom

²Department of Atmospheric Sciences, Colorado State University, Fort Collins, USA

³Leibniz Institute for Tropospheric Research, Leipzig, Germany

Correspondence: Max Heikenfeld (max.heikenfeld@physics.ox.ac.uk)

Abstract. We introduce tobac (Tracking and Object-Based Analysis of Clouds), a newly developed framework for tracking and analysing individual clouds in different types of datasets, such as cloud-resolving model simulations and geostationary satellite retrievals. The software has been designed to be used flexibly with any two- or three-dimensional time-varying input. The application of high-level data formats, such as ~~iris-cubes-or-xarray~~Iris cubes or xarray arrays, for input and output allows for convenient use of metadata in the tracking analysis and visualisation. Comprehensive analysis routines are provided to derive properties like cloud lifetimes or statistics of cloud properties along with tools to visualise the results in a convenient way. The application of tobac is presented in two examples. We first track and analyse scattered deep convective cells based on maximum vertical velocity and the three-dimensional condensate mixing ratio field in cloud-resolving model simulations. We also investigate the performance of the tracking algorithm for different choices of time resolution of the model output. In the second application, we show how the framework can be used to effectively combine information from two different types of datasets by simultaneously tracking convective clouds in model simulations and in geostationary satellite images based on outgoing longwave radiation. tobac provides a flexible new way to include the evolution of the characteristics of individual clouds in a range of important analyses like model intercomparison studies or model assessment based on observational data.

1 Introduction

Clouds are a major feature of the Earth's atmosphere and control many critical processes in the Earth's energy and water budgets (Trenberth et al., 2009). Different types of convective clouds play important but distinct roles in many regions of the globe. Shallow cumulus clouds are widespread over the subtropical trade-wind latitudes and have a strong impact on the radiative balance of the atmosphere, including a potential for strong feedbacks from anthropogenic perturbations of the climate system (Stevens and Feingold, 2009). Deep convective clouds are a defining element of the atmosphere over most of the tropics (Nesbitt et al., 2006), driving both local weather dynamics and ~~larger-scale~~large-scale circulation patterns, which has impacts on the entire climate system (Emanuel, 1994). Furthermore, deep convective clouds play a major role in extreme weather events all over the globe (Doswell, 2001; Gensini and Mote, 2014). Therefore, clouds and their interactions with other aspects of the climate system are an essential aspect of many important challenges in our understanding of the Earth's atmosphere and

current changes due to anthropogenic influences (IPCC, 2013). The nature of convective clouds is highly localised. Individual convective cells undergo rapid dynamic development over relatively short timescales of minutes to hours (Orlanski, 1975), while organised convective features, such as mesoscale convective systems (MCS), can persist for many hours or even days (Orlanski, 1975; Laing and Fritsch, 1997; Fritsch and Forbes, 2001; Feng et al., 2018). Further advances in understanding the physical processes underlying the development of these clouds require analysis techniques that go beyond the usual approaches, which are often based on bulk statistical properties over larger regions in space and time, such as entire modelling or observational domains. Model intercomparison studies with cloud-resolving model (CRM) simulations have mostly relied on the comparison of domain and time-averaged quantities or similar statistics (Varble et al., 2011, 2014a, b; Fan et al., 2017). This generally limits the investigation of differences between the models on the scale of individual convective cells or analyses that take the temporal evolution of individual clouds into account.

Any analysis ~~that focuses~~ focused on the properties of individual clouds in larger databases containing numerous cloud elements ~~, and aims and aimed~~ at including the time evolution over their development cycle ~~, requires~~ some form of cloud tracking technique. A large body of work exists on tracking individual clouds in different types of data, ranging from ground-based radar and geostationary satellite retrievals to model simulations at a range of different resolutions. We now present a short but certainly not exhaustive overview of existing approaches. This will be used to show the capabilities of the existing software and to discuss drawbacks or limitations which motivated the development of the more flexible software framework *tobac* (Tracking and Object-Based Analysis of Clouds) presented here.

Tracking of individual convective clouds in radar data has been performed for decades (Crane, 1979; Rosenfeld, 1987). These efforts were often motivated by their use in nowcasting of severe weather warnings, e.g. for flooding due to convective precipitation, damage from hail or impacts of high wind speeds such as tornadoes (Dixon and Wiener, 1993; Lakshmanan and Smith, 2009). Satellite-based tracking of convective clouds has been performed both with a similar focus on nowcasting convection and for long-term analysis in climate research (Menzel, 2001; Sieglaff et al., 2012). Special tracking algorithms that combine information from different wavelength bands of imagers on geostationary satellites, such as Cb-TRAM (Zinner et al., 2008, 2013) and RTD (Autonès and Moisselin, 2013) have been developed as tools to identify and track deep convective clouds throughout their development cycle, including the initial stage of rising cumulus towers. However, both products have been developed for a specific application that strongly limits a more general adaption of the software by the user. Several other studies have used geostationary satellite data to investigate the growing phase and glaciation of deep convective clouds (Mecikalski and Bedka, 2006; Mecikalski et al., 2011; Senf et al., 2015; Senf and Deneke, 2017). Other applications have specifically focused on the analysis of long-lived MCSs over different regions of the globe (Machado et al., 1998; Feng et al., 2012; Hagos et al., 2013; Feng et al., 2018). The identification of individual cloud objects in satellite data is one aspect of cloud tracking and has been used to investigate the spatial scaling of clouds on a global scale (Wilcox and Ramanathan, 2001; Wood and Field, 2011), including studies on the representation of these distributions in global atmospheric models (Wilcox, 2003; O'Brien et al., 2013). The use of satellite data from active sensors (Nesbitt et al., 2000; Bacmeister and Stephens, 2011; Riley et al., 2011; Igel et al., 2014; Guillaume et al., 2018) allows to include information about the vertical extent of identified cloud objects, which provides improved classification of cloud types and understanding of important physical processes.

Tracking of individual cloud objects in high-resolution CRM simulations and large-eddy simulation models (LES) has been developed alongside the evolution of these simulations in recent decades. Earlier studies on tracking shallow convection in high-resolution model simulations (Zhao and Austin, 2005a, b; Heus et al., 2009) strongly relied on manual detection techniques. Subsequently, Dawe and Austin (2012) and Heus and Seifert (2013) presented automated methods of tracking shallow convection that rely on a continuous release of a decaying tracer from the model as described in Couvreur et al. (2010). However, the functionality of the tracer release and advection must be specifically implemented in each model and restricts the use of this technique to output of high-resolution models. Cloud tracking algorithms applied online during the actual model simulations (Plant, 2009) have the advantage of direct access to the relevant model fields at the model ~~timestep~~time step and thus the highest possible time resolution. However, these online algorithms must also be implemented separately in a specific model.

Moseley et al. (2013, 2016) tracked precipitation patterns for investigations of deep convective clouds and convective invigoration. Davis et al. (2006, 2009) presented an object-based analysis of rainfall patches, including tracking capabilities, which was applied to precipitation on a relatively large regional scale. Heiblum et al. (2016a, b) developed and applied a tracking algorithm for warm convective clouds that determines cloud volumes from the condensate mixing ratio field and then propagates the clouds based on the velocity of the cloud centre of mass. This algorithm allows for cloud splits and merges to form complex cloud entities possibly involving numerous individual clouds. Only a few studies have focused on tracking individual deep convective clouds in model simulations in a way that takes into account the actual cloud volumes (Chen et al., 2017). Terwey and Rozoff (2014) developed a tracking algorithm for individual convective updrafts and applied it to CRM simulations of hurricane cases with two different models. However, this effort has not been provided to the community as a generalised software package aimed at more widespread use cases. Several other approaches that included the tracking of individual updrafts in different types of cumulus clouds in a very detailed manner (Sherwood et al., 2013; Hernandez-Deckers and Sherwood, 2016) would not be easily transferable to data with a lower temporal and spatial resolution. Despite these advances in developing detailed cloud tracking approaches for use in highly resolved model simulations, most current studies are performed with model grid spacings of several hundreds of metres to a few kilometres, especially when using larger domains or simulations for longer time periods. Providing adequate ways of performing tracking and object-based analyses for different types of clouds, including deep convection, in these kinds of simulations provides a key pathway to better understanding the underlying physical processes.

This overview clearly shows the wide range of extensive efforts that went into the development of elaborate software and analysis tools to track clouds in different types of datasets. ~~However, it~~The application of cloud identification and tracking and related techniques has substantially increased our understanding of cloud size distributions, the time evolution of different types of clouds, and the underlying physical processes governing cloud formation, development, and propagation. However, the overview also highlights the problem of limited compatibility between the different existing approaches and implementations, especially regarding the intended use of tracking clouds based on different data sources using the same algorithms and analysis tools.

To address some of these limitations of existing approaches and provide a more functional tool with increased flexibility for

different applications, we have developed `tobac` as a new flexible software tool for the identification, tracking and analysis of clouds. This approach certainly does not intend to replace the existing tools in their specific applications, but is rather aims to provide a flexible framework that can be used for a wide range of different datasets and also allows the future integration of some of the existing approaches discussed here.

`tobac` is designed in a modular way that includes the following basic steps, which are described in detail in the following sections 2.1–2.5:

1. Data input and output
2. Feature detection
3. Segmentation of cloud areas/volumes
4. Trajectory linking
5. Object-based analysis and visualisation

`tobac` provides a framework that allows for a convenient application to output from a wide range of model simulations and observational products, as long as it is provided with sufficient temporal and spatial resolution and contains output variables that can be used to identify individual clouds. Therefore, the software package can be used for a range of important applications like model intercomparison studies, which generally rely on simpler analysis methods that do not capture the evolution of individual clouds. These capabilities also allow for comparative studies between model simulations and observational datasets, e.g. from satellite retrievals, using the same underlying statistical methods. Due to the modular structure, `tobac` is set up for the integration of existing or newly developed algorithms for the different steps in the analysis chain. The implementation in Python provides `tobac` with access to numerous more specialised existing software libraries for different aspects of the software, such as data input/output, memory usage and the existing functionality from the field of image processing. We also show how we can leverage an existing Python library from an entirely different field of the physical sciences to perform integral parts of the linking step in our application. Furthermore, the choice of Python for `tobac` makes the package more easily accessible to users as it allows for easier integration into existing analysis workflows and also stimulates the integration of additional components in the modular workflow of the package.

To show the advantages of `tobac` in practical applications, we present two different examples of using the framework in tracking and analysing deep convective clouds. In the first application, the detection of features is performed on the column-maximum vertical velocity at each output time interval from a CRM simulation. A three-dimensional watershedding algorithm is applied to the updraft field and to the total condensed water content field (mass mixing ratio of all hydrometeors) at each step in time to infer both the volume of the individual updrafts and the clouds associated with the tracked updrafts. These features are then filtered and linked into consistent trajectories. We use the tracking results to assess the distribution of cloud lifetimes and the requirements for the model output temporal resolution. In the second application, we perform a simultaneous analysis for model and satellite data. Similar vertically resolved data as used in the other example are usually not available from satellite

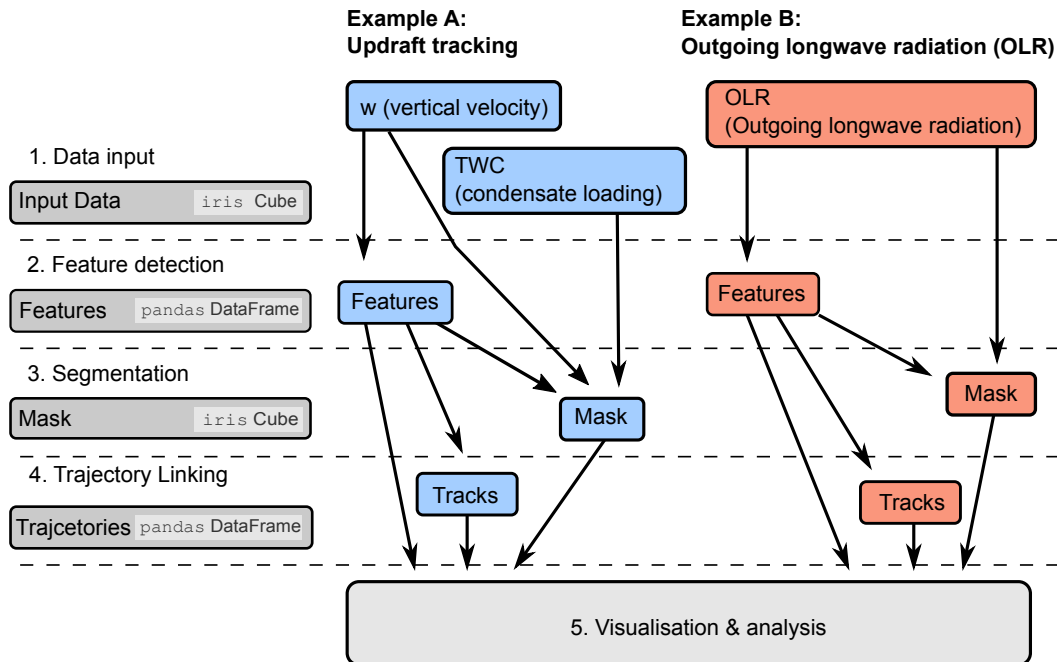


Figure 1. Schematic overview of the general workflow of the tobac tracking analysis framework and of the two examples presented in this manuscript.

imagers. The information in most satellite retrievals of cloud properties is limited to two-dimensions. With a multi-spectral selection of channels from the satellite instrument, cloud-top height and radiative properties can be retrieved (McGarraugh et al., 2018). An analysis of model-simulated and satellite-retrieved fields of outgoing longwave radiation is presented to demonstrate the flexibility of the tobac approach. By making use of the framework consistently across different datasets like this, we can compare the tracked clouds in both data sources by examining the statistical properties of the resulting population of convective clouds, thereby facilitating model-observation comparisons.

2 Software description

In this section, we describe the general design and workflow of the software package as illustrated in Fig. 1 for the two example applications presented in Sect. 3 and Sect. 4. The implementation of the individual analysis steps described here reflects an example combination of analysis steps currently implemented in tobac. Due to the modular setup of the package, different parts of the workflow can be combined in a different way or replaced by future additions to the framework.

2.1 Data input and output

The input data for the framework are provided in a high-level format of either ~~Iris-Iris~~ cubes (Met Office, 2018) or ~~xarray~~ xarray DataArrays (Hoyer and Hamman, 2017), which include detailed metadata for each data variable, such as units and coordinates. The algorithm can thus automatically use these metadata, and the tracking setup can be controlled independently of the temporal resolution, spatial resolution or dimensions of the input data. Tracking parameters representing physical properties like distances or time periods can be set in physical units and are automatically converted to pixel-based values needed for the underlying calculations. Scientific data are provided in a vast variety of different file formats and data structures. Implementing a way of loading the data into the right format for an application often proves to be a significant limitation to the use of new datasets and generally consumes an unjustifiable amount of time and effort, apart from providing ~~a significant an~~ important source of implementation errors. The Python library Community intercomparison Suite (CIS) (Watson-Parris et al., 2016) overcomes this challenge and provides a convenient way to automatically load a vast array of observational datasets into ~~Iris-compatible-Iris-compatible~~ objects for direct use in tobac.

Both ~~Iris-and-xarray-Iris and xarray~~ make use of so-called "lazy loading" based on the ~~dask-dask~~ package (Rocklin, 2015; Dask Development Team, 2016) for efficient memory usage. The initial loading of data from a file only creates a place-holder. Then, individual operations on the data are combined and evaluated once the final result is to be saved, printed or plotted. Only at this stage are data actually loaded from disk into the physical memory of the computing machine and individual calculations performed. Based on these capabilities, the entire tobac framework is written with a focus on limiting instantaneous memory usage by splitting up calculations into chunks, e.g. along the time dimension. Hence, even large datasets with individual fields much larger than the memory available on the computing system can be conveniently processed without special adaptation by the user.

The output of the tracking analysis is given in commonly used high-level Python data format as ~~pandas-pandas~~ DataFrames (McKinney, 2010) for a table containing the tracked cell centres and trajectories and as ~~Iris-cubes-or-xarray-Iris cubes or xarray~~ DataArrays for the masks of cloud volumes or areas. This output is automatically amended with the same metadata as the input data like coordinates (e.g. time, longitude, latitude), along with additional information from the tracking process, e.g. a time coordinate relative to the initiation of an individual convective cell. This allows for the convenient and direct use of the output for visualisation and further analyses. The intermediate results of each individual analysis step can be conveniently saved and examined in the form of pandas DataFrames or Iris cubes.

2.2 Feature detection

The feature detection can work with any two-dimensional field either present or derived from the input data. In the first example, we use maxima in the maximum vertical velocity in each column of the three-dimensional high-resolution model output to identify individual updrafts (see Sect. 3). In the second example, minima in outgoing longwave radiation from satellite retrievals and model output are used in the feature detection (see Sect. 4).

To identify the features, contiguous regions above or below a threshold are determined and labelled individually. Smoothing the input data, e.g. with a Gaussian filter, ~~makes this step much more reliable.~~ ~~Erosion of the labelled regions by a specified~~ The detection of regions above a specific threshold can lead to large interconnected regions combining several features
 5 linked by narrow ridges. To prevent this and identify these interconnected features separately, the tobac feature detection allows for the use of "erosion" techniques based on the implementation in `skimage.morphology.binary_erosion`. These techniques shrink the identified regions from the edges by a specific length or number of pixels~~leads-~~, thus removing the connecting ridges between interconnected features. This has been shown to lead to more robust detection of individual features, as described in detail in Senf et al. (2018).

- 10 To describe the specific location of the feature at a specific point in time, we have investigated the use of different spatial properties describing the identified region. The geometric centre can be strongly affected by changes in the shape of the boundary, which is determined based on the selected threshold value. Instead, we have found that a weighted mean

$$\mathbf{x}_{\text{meanmean}} = \sum_i \mathbf{x}_i w_i \quad (1)$$

with weights w_i given by the difference between the values of the chosen field at the individual points V_i and the threshold value V_{feature}

$$w_i = \frac{V_i - V_{\text{feature}}}{\sum_i V_i - V_{\text{feature}}} \frac{V_i - V_{\text{feature}}}{\sum_i V_i - V_{\text{feature}}} \quad (2)$$

has proven to perform best in determining a robust feature position. We can interpret this position as the centre of mass of the component of the field exceeding the chosen threshold value.

- Using a single threshold to identify features can lead to problematic results in two different ways. A very restrictive threshold
 20 ~~means that can result in omitting~~ clouds with weak vertical velocities or clouds during their initial and decaying stage will not be captured. On the other hand, a weakly restrictive threshold can lead to spurious results as it might lead to large unconfined regions around deep convection being selected, or to an unwanted merging of several distinct cloud features into one. To resolve these conflicting requirements arising in the case of a single threshold value, we have developed a step-wise approach with a range of threshold values (Fig. 2). These threshold values have to be chosen specifically for each application of tobac.
- 25 The choice can be based on a detailed analysis of the data used for tracking to determine where the features separate from the background, e.g. based on histograms as shown in Sect. 4, or using empirical values from previous studies of the specific phenomena. The feature identification starts with labelling the regions for the least restrictive threshold. For each threshold value, features are identified in the same way (Fig. 2b,d,f) and replace existing features that were found based on a less restrictive threshold value in the surrounding region (Fig. 2e,g).
- 30 This combination of different thresholds allows tobac to detect lower intensity features representing weaker convective clouds or clouds in their initial or decay stage but identify localised features with stronger updrafts or colder cloud tops within the weaker-threshold areas. In the first example application (Sect. 3), consecutive maximum updraft threshold values of 3, 5 and 10 m s^{-1} were used, while tracking based on OLR in the second example (Sect. 4) was performed with consecutively smaller threshold values (250, 225, 200, 175 and 150 W m^{-2}). While using multiple thresholds is usually beneficial, feature detection

using a single threshold value is possible in tobac by only supplying a single threshold value and can be appropriate in certain applications.

An iterative set of threshold values was used in other recent approaches to cloud detection (Liang et al., 2017; Fu et al., 2019), however with a specific focus on the application to certain types of satellite images. Multiple-threshold methods are also applied in other fields of science facing similar challenges in feature detection such as astronomy (Zheng et al., 2015).

2.3 Segmentation

Once features and feature centres are identified, segmentation techniques are used to associate areas or volumes to each identified feature. In the current version of the tobac framework, we have implemented segmentation using watershedding techniques from the field of image processing (~~skimage.morphology.watershed from the scikit-image~~ `skimage.morphology.watershed from the scikit-image` library (Soille and Ansault, 1990; van der Walt et al., 2014)) with a fixed threshold value V_{seg} . This value has to be set specifically for every type of input data and application, as explained in more detail for the two example applications in Sect. 3 and Sect. 4. Watershedding ~~is~~ `segmentation treats the input field as a topographic map and separates the input into individual regions similar to individual watersheds or catchment basins along a dividing ridge in a geological context` (Meyer, 1994). These techniques are widely used in several existing cloud tracking and analysis algorithms described in Sect. 1, such as Heiblum et al. (2016a), Fiolleau and Roca (2013) and Senf et al. (2018).

This segmentation routine can be performed for both two-dimensional and three-dimensional data. At each ~~timestep~~ `time step`, a marker is set at the position (weighted mean centre) of each feature identified in the detection step in an array otherwise filled with zeros. In the case of the three-dimensional watershedding, all cells in the column above the weighted mean centre position of the identified features fulfilling the threshold condition $V_{segmentation}$ are set to the respective marker. The algorithm then fills the area (2D) or volume (3D) based on the input field starting from these markers until reaching the threshold $V_{segmentation}$. If two or more cloud objects are directly connected, the border runs along the watershed line between the two regions. This procedure creates a mask of the same shape as the input data, with zeros at all grid points where there is no cloud or updraft and the integer number of the associated feature at all gridpoints belonging to that specific cloud/updraft. This mask can be conveniently and efficiently used to select the volume of each cloud object at a specific ~~timestep~~ `time step` for further analysis or visualisation.

The structure of tobac allows for the future implementation of other algorithms for the segmentation step, e.g. replacing the watershedding approach by random walk techniques (Grady, 2006; Wang et al., 2019) or other image processing tools. `Similarities between the feature detection and segmentation steps mean that these steps could be combined in some implementations in future versions of tobac, e.g. for applications based on a single input dataset (OLR), as used in Sect. 4. However, treating the two analysis steps separately allows for the combination of different datasets (vertical velocity and condensed water content), as shown in Sect. 3.`

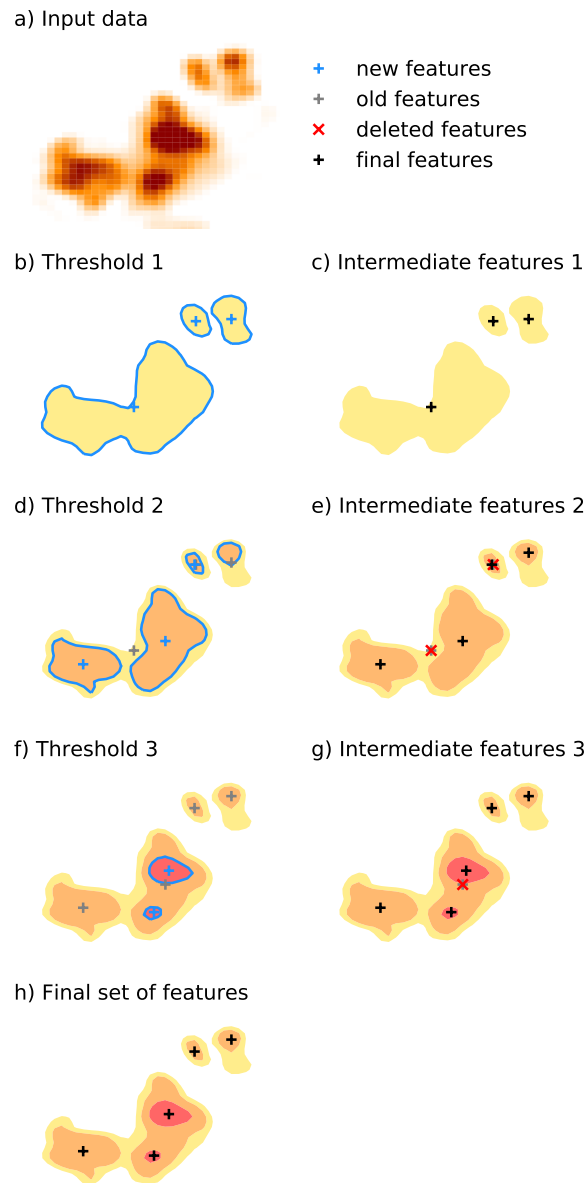


Figure 2. Schematic illustration of the multi-threshold feature detection approach using three different threshold values.

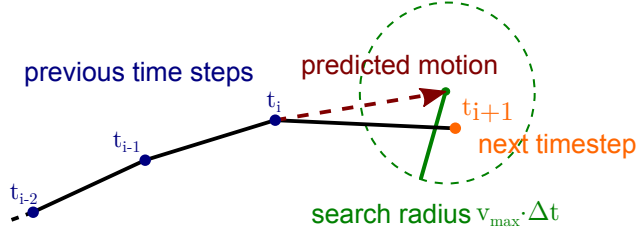


Figure 3. Schematic illustration of the trajectory linking with the predicted motion of the feature based on previous time steps and a search range around the predicted position.

2.4 Trajectory linking

- 15 The individual features and associated area/volumes identified in each **timestep** have to be linked into cloud trajectories to analyse the time evolution of cloud properties for a better understanding of the underlying physical processes. For this step, we have implemented a linking method that makes use of **trackpy-trackpy** (Allan et al., 2016), a Python library originally developed for tracking particles and cells in microscopic images. The linking determines which of the features detected in a specific **timestep** (see Sect. 2.2) is identical to an existing feature in the previous **timestep** and is illustrated
- 20 in Fig. 3. For each existing feature, the movement within a time step is **extrapolated-predicted** based on the velocities in a number of previous time steps. The algorithm then breaks the search process down to **a-few**-candidate features by restricting the search to a circular search region centred around the predicted position of the feature in the next time step. For newly initialised trajectories, where no velocity from previous **timesteps** is available, the algorithm resorts to the average velocity of the nearest tracked objects. The parameter v_{\max} restricts how much the future position of a feature is allowed to
- 25 deviate from a linear extrapolation of the trajectory over time. It thus has the units of a velocity and describes the dependency of the circular search range d on the output **timestep** Δt in the data used for the tracking

$$d = v_{\max} \Delta t. \quad (3)$$

- In the applications (**presented in** Sect. 3 and 4), we set this value to $v_{\max} = 10 \text{ m s}^{-1}$, which results in a search range of 600 m around the projected position for 1 minute data input and 3 km for 5 minute data input. Variations in the shape of the regions
- 30 used to determine the positions of the features can lead to quasi-instantaneous shifts of the position of the feature by one or two grid cells even for a very high temporal resolution of the input data, potentially jeopardising the tracking procedure. To prevent this, **tobac** uses an additional minimum radius of the search range d_{\min} (2 km, equivalent to four times the grid spacing in Sect. 3) that specifies a lower limit for the size of the search region. Both these parameters are given as physical quantities and then converted into pixel-based values used in **trackpy-trackpy**. This allows for cloud tracking that is controlled by physically-based parameters that are independent of the temporal and spatial resolution of the input data. We make use of this for cloud tracking with different model output frequencies for the same simulations in the example application in Sect. 3.

Features can be allowed to be missed for a certain number of ~~timesteps~~time steps (*memory*) and still get linked into a trajectory. However, this option should be used with caution, as it can lead to erroneous trajectory linking, especially for data with low time resolution. For example, convective clouds can produce outflow boundaries that initiate new convective clouds nearby, and the newly-formed clouds are more likely to be linked to the original clouds with this option.~~The-~~
The feature detection step can often omit the initial or final stages of the evolution of a cloud due to the choice of specific thresholds. Thus, trajectories can also be extrapolated to additional output ~~timesteps~~time steps at the start and at the end of the tracked path. This allows for the inclusion of both the initiation of the cell and the decaying later stages ~~of the cell development~~
~~that may have been unidentified based on the chosen thresholds~~in the analysis of the cloud life cycle. Furthermore, a threshold for the minimum lifetime of the tracked objects can be used to exclude the analysis of clouds that have only been tracked for a very short period and are likely to be spurious features. Such tracked objects can contaminate analyses focusing on the cloud lifetime and associated quantities.

The trajectories are recorded in a ~~pandas DataFrames~~pandas DataFrame. This enables filtering the resulting trajectories, e.g. to reject trajectories that are only partially captured at the boundaries of the input field both in space and time.

The current ~~version~~implementation of the linking step does not include an explicit treatment of the splitting and merging of clouds, as implemented in several of the cloud tracking algorithms reviewed earlier (Dawe and Austin, 2012; Heus and Seifert, 2013; Heiblum et al., 2016a). Instead, the ~~algorithm creates current version of tobac will create~~ a continuous track ~~for the cloud that most directly follows the~~ with only one of the two separate cloud objects that combine in a merger or evolve from splitting of a tracked object, mostly based on which of these has the more similar direction of travel ~~of the preceding or following cell path to the joint object~~. However, we have structured the implementation of tobac in a way that allows for the future addition of more complex tracking methods ~~recording that can record~~ a more complex network of relationships between cloud objects at different points in time.

2.5 Object-based analysis and visualisation

To make use of the results of the previous steps, we provide detailed tools to analyse and visualise the tracked objects. We provide a set of routines that enable performing analyses and deriving statistics for individual clouds, such as the time series of integrated properties and vertical profiles. We also provide routines to calculate summary statistics of the entire population of tracked clouds in the cloud field like histograms of cloud areas/volumes or cloud mass and a detailed cell lifetime analysis (see Fig. 6 and Fig. 10).

These analysis routines are all built in a modular manner. Thus, users can reuse the most basic methods for interacting with the data structure of the package in their own analysis procedures in Python. This includes functions performing simple tasks like looping over all identified objects or cloud trajectories and masking arrays for the analysis of individual cloud objects. Plotting routines include both visualisations of the entire cloud field and detailed visualisations for individual convective cells and their properties.

2.6 Advantages of the implementation in Python

While the majority of the existing tracking approaches reviewed in Sect. 1 are implemented either in Fortran, C and C++ or in proprietary programming languages like MATLAB, we have chosen to use Python for our tracking framework for several practical reasons. Python has become the go-to standard for data analysis in many fields of scientific research, including the atmospheric sciences in recent years (Lin, 2012; Perkel, 2015). This makes it possible to develop software that is accessible and modular, which allows for the successful addition of user-contributed algorithms or the adoption or application of the workflow for cases beyond those presented here. The use of libraries in the scientific Python ecosystem including ~~NumPy, SciPy, and matplotlib~~ NumPy, SciPy, and matplotlib (Hunter, 2007; van der Walt et al., 2011), along with a large stack of existing and optimised libraries providing image processing features (van der Walt et al., 2014), means that the package is based on actively developed open-source projects. This ensures an accurate, effective and tested implementation of the individual calculations as well as the continuous integration of new developments and improvements. Most of these Python libraries use Fortran or C for the actual underlying calculations, which means that many of the individual operations within `tobac` make use of the increased computational speed of these languages. The use of Python also means that even users without extensive programming experience will be able to easily adapt existing procedures into the workflow or contribute additional algorithms to the modular structure of the `tobac` tracking framework.

The implementation in Python also enables the use of Jupyter notebooks (Perez and Granger, 2007; Kluyver et al., 2016) as an innovative way of developing, visualising and sharing scientific data analyses. We provide examples of the analyses presented here as Jupyter notebooks provided in the software package.

Memory limitations have been cited as a significant challenge for the application of many of the presented algorithms (Dawe and Austin, 2012; Heus and Seifert, 2013). The use of modern memory management techniques such as "lazy data loading" in the underlying Python libraries ~~Iris-Iris~~ Iris-Iris (Met Office, 2018) or ~~xarray-xarray~~ xarray-xarray (Hoyer and Hamman, 2017), which both rely on the ~~dask-dask~~ dask-dask data types (Rocklin, 2015), allows for clear and concise source code while sparing the users of having to deal with most memory-related considerations themselves. Memory usage and algorithm run time for the two applications presented in this manuscript are included in the following sections.

3 Example A: tracking of convective cells in ~~high-resolution~~ high-resolution model simulations based on updraft velocities and condensate mixing ratios

In the first example, we apply the tracking framework to CRM simulations of scattered deep convection. Deep convective clouds are characterised by regions of strong vertical motions which are concentrated in relatively confined updraft cores that dominate the dynamics of the cloud evolution (Cotton et al., 2010). Hence, the updraft cores are well suited to be used for identifying and tracking individual convective cells. We use the total condensate mixing ratio, i.e. the total amount of liquid and frozen water per mass of dry air, to associate the identified updraft cores with the respective cloud volume at each time.

We make use of simulations that were performed as part of a larger model intercomparison case study in the deep convection working group of the Aerosol, Clouds and Precipitation (ACPC) initiative (van den Heever et al., 2017) aimed at understanding

the response of scattered convection over the region around Houston, Texas to changes in aerosol number concentrations. The tracking algorithm presented here will be used as part of the analysis for the model intercomparison study using several different three-dimensional CRMs. ~~In this manuscript, we show the identification and tracking of deep convective clouds on a model grid with horizontal grid spacing of 500~~
 5 ~~The simulations are performed in a nested setup with three domains using grid spacings of 4.5 and vertical grid spacing of 50–300 km, 1.5 km and 500 mm for simulations with the Weather Research and Forecasting (WRF) model. The simulations use the Morrison microphysics scheme and the Rapid Radiative Transfer Model (RRTMG) short and longwave radiation scheme. Initial~~ Initial conditions for all domains and boundary conditions for the ~~simulations in a nested setup with two additional grids of coarser grid spacing (1.5 and 4.5) around this target~~ outermost domain were
 10 provided by the GDAS-FNL reanalysis (NCEP, 2015). The simulations have been performed for 24 hours from 12:00 UTC on the 19 June 2013 to 12:00 UTC on the 20 June 2013. The simulations setup is described in more detail in van den Heever et al. (2017). The model time stepping is 3 s for the outmost domain and 1.5 s for the two inner domains. In this example, we use data from the innermost domain with a 500 m grid spacing and 500 grid cells in each horizontal direction. The simulation results are output at a frequency of 1 minute for an extended part of the simulation period (3 hours, 21:00 – 24:00 UTC) and at a frequency of 5 minutes for 12 hours of the simulations ~~-(16:00 – 04:00 UTC).~~ The outermost domain of the same nested
 5 simulation setup is used for the comparison with satellite data presented in Sec. 4.

For the two following examples, we use model results from simulations with the Weather Research and Forecasting (WRF) model (Skamarock et al., 2005). These simulations use the Morrison microphysics scheme (Morrison et al., 2005, 2009) and the Rapid Radiative Transfer Model (RRTMG) short and longwave radiation scheme (Iacono et al., 2008).

We use a combination of the three-dimensional fields of vertical velocity and total condensate mixing ratio in this application
 10 to track individual convective clouds.

~~Schematic overview of the individual steps of the tracking algorithm for an example subset of the domain used in example A including the input mid-tropospheric velocity field. The input data (a) are smoothed with a filter (b) before regions above or below a set of thresholds are determined (c) to identify the individual features (d). (e) shows the surface projection of the associated cloud volumes determined in the segmentation set and (f) shows the entire trajectories of the cells present at this~~
 15 ~~timestep, including the surface projection of the cell volume at the start (dashed) and at the end (solid) of the trajectory.~~ The individual steps of the analysis are visualised for a specific point in time and a subset of the model domain in Fig. 4. The three-dimensional vertical velocity field is reduced to the maximum updraft velocity in each model column over a mid-tropospheric range of geopotential height (3000 m to 8000 m) (Fig. 4a). This avoids the impact of strong vertical motions both in the lower troposphere, that may be associated with outflow boundaries, and also gravity waves in the upper troposphere. A Gaussian
 20 filter with a variance of $\sigma = 1$ km is used to filter the input in the feature detection step (Fig. 4b) to create a smoother field that assists in the feature detection. This two-dimensional field is then used to identify individual deep convective updrafts in the simulation. The feature identification following Sect. 2.2 is performed with a set of three updraft velocity thresholds of 3 m s^{-1} , 5 m s^{-1} and 10 m s^{-1} (Fig. 4c) and yields the individual features marked in Fig. 4d. Segmentation is performed on the condensate mixing ratio using the watershedding technique (see Sect. ~~??2.3~~) with a threshold of 0.5 g kg^{-1} to identify
 25 the cloud volumes corresponding to the individual identified updrafts. The cloud volumes derived with watershedding from

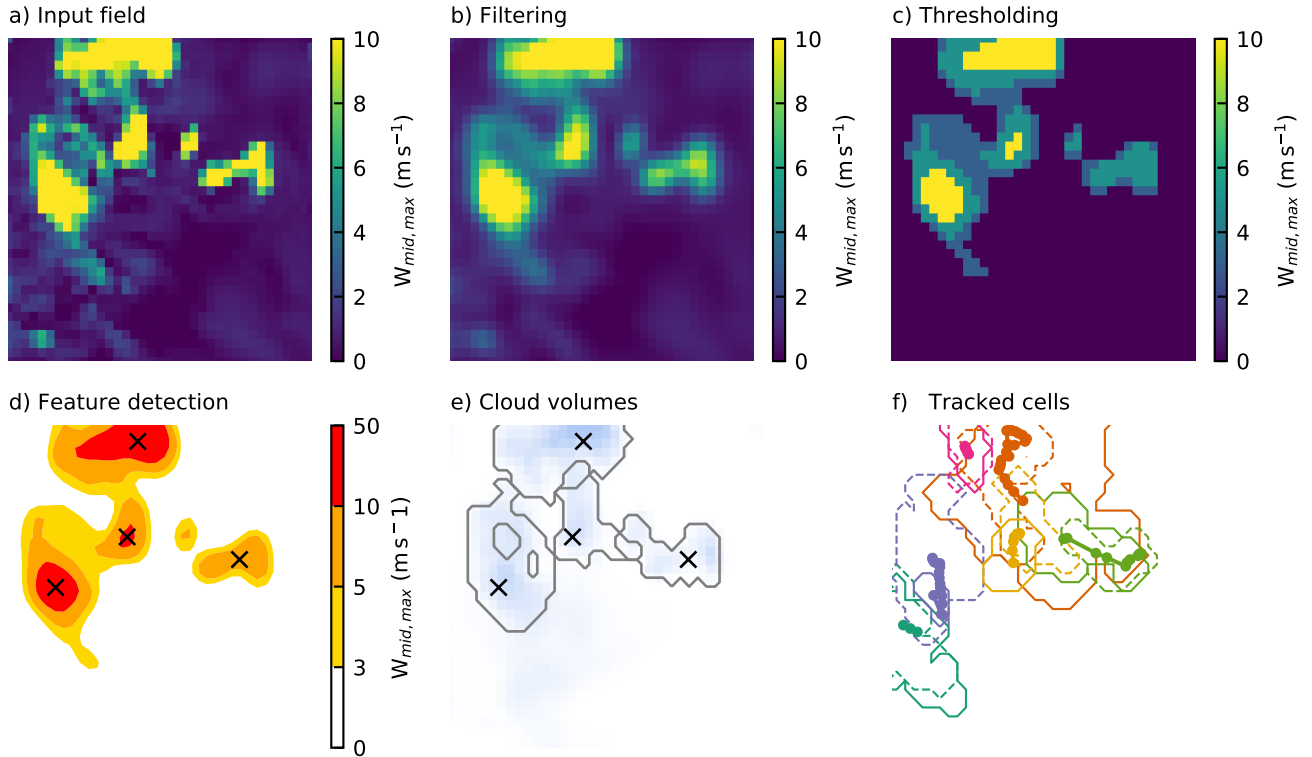


Figure 4. Schematic overview of the individual steps of the tracking algorithm for an example subset of the domain used in example A including the input mid-tropospheric velocity field. The input data (a) are smoothed with a filter (b) before regions above or below a set of thresholds are determined (c) to identify the individual features (d). (e) shows the surface projection of the associated cloud volumes determined in the segmentation set and (f) shows the entire trajectories of the cells present at this time step, including the surface projection of the cell volume at the start (dashed) and at the end (solid) of the trajectory.

the condensate mixing ratio field of each of the identified updrafts is represented by the surface projection of the 3D volumes (Fig. 4e). Note that the intersecting lines in Fig. 4e represent instances where cloud volumes associated with different updraft cores may be present in the same column but at different altitudes. Trajectories are formed by linking up the individual features and are shown including the surface projection of the cloud volumes at the initial and final ~~time step~~ time step of each tracked cell (Fig. 4f).

The data processing has been performed on the JASMIN data analysis facility (CEDA, 2019). The script including feature detection, segmentation, trajectory linking and saving of the analysis output for the 1-min data output had a processing time of around 17 minutes with a maximum memory footprint of 3.1 GB using a maximum of 3 processes and 27 threads. The segmentation step has been broken up into chunks of 10 minutes each to limit the total memory consumption of the analysis. The processing time is almost entirely taken up by the segmentation step using time-resolved 3-dimensional data of the total condensate. It is thus strongly affected by the time required to access the data on the disk and thus highly dependent on both

[the infrastructure and the structure of the data file, i.e. the data compression in the input files.](#) A smaller subset of the data and analysis for this example including the tracking analysis and visualisation is available as a Jupyter notebook as part of the package source code.

3.1 Time resolution requirements for cloud tracking

The cloud tracking framework presented here can be applied to model output from any atmospheric model simulation with sufficient resolution to resolve the features intended to be studied. However, successful tracking of individual clouds in the simulation output requires sufficiently high spatial and temporal resolution. However, writing output data at high frequency from numerical model simulations drastically increases the computational expense of the simulations and the size of the output datasets. For observational data, such as geostationary satellite data, the available time resolution might be limited by technical restrictions such as scanning time or data transmission. It is thus important to determine the necessary input frequency for the successful tracking of a specific type of dataset.

The tracking step (Sect. 2.4) uses `trackpy`, which is based on the tracking methods developed for in Crocker and Grier (1996). The algorithm has been originally developed for microscopic particles; however, all considerations apply equally to the tracked features we regard here in `tobac`. In their development of the algorithm, the authors state that successfully linking objects into trajectories is only feasible if the typical displacement of a particle during one time step is significantly smaller than the typical inter-particle spacing. To assess how valid these assumptions are for our application, we investigated the nearest-neighbour distances for individual cells and the typical displacement of the tracked objects within one `timestep`. Distances between cloudy updrafts (Fig. ??5a) were most frequently around 5 km with a significant-substantial tail of up to 30 km representing more isolated cells. This distribution is independent of the chosen output `timestep` as it represents an instantaneous relationship between cells at individual points in time. The updraft propagation velocities derived for tracking with a 1 minute output `timestep` (Fig. ??5b) were most frequently at around 4 m s^{-1} with more than 90 % of the velocities below 10 m s^{-1} .

Using the output `timestep` and these velocities, we can calculate the displacement of the clouds within one tracking `timestep` and compare that to the nearest-neighbour distances (Fig ??5c). In addition to the `timestep` of 1 minute, the displacements that would result from lower output frequencies of 5, 10, 15 and 30 minutes based on these velocities were calculated (Fig ??5c). While there is no significant-clear overlap between the nearest-neighbour distance distribution and the displacement distribution for an output `timestep` of 1 minute, the tails of the distributions start overlapping for 5-minute input data, although the peaks are still distinctly separate. For lower output frequencies of 15 minutes and 30 minutes, however, there is a clear overlap between the nearest-neighbour distance distribution and the distributions of displacement within one time step. Therefore, these frequencies would be outside the range postulated for the successful application of the tracking algorithm used by `trackpy`. Hence, when applying this tracking algorithm, it is important to understand both the spatial distribution of the desired tracked features and their propagation velocities to ensure that the output `timestep` is sufficiently frequent. For the simulations assessed here, both 1-minute and 5-minute output frequencies would be acceptable for tracking cloudy updrafts, with 1-minute output likely to provide more successful and accurate tracks.

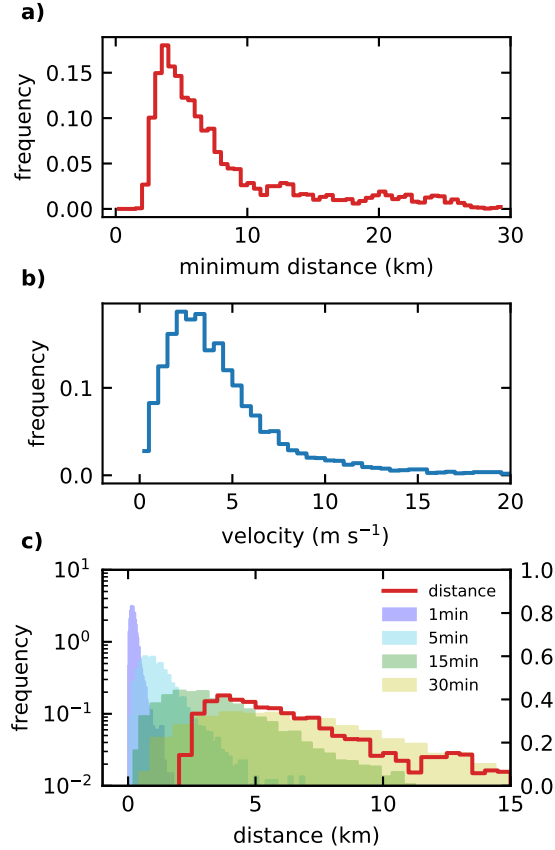


Figure 5. (a) Distributions of the distance to the next identified object for all identified objects and (b) velocities for tracked cloud objects at each timestep of the trajectories. (c) The distribution of derived travel distances of individual clouds during one output timestep (shaded colours) resulting from these velocities in (b) is shown together with the distribution of the minimal distance to the nearest-neighbour for individual objects as shown in (a).

The cloud lifetimes (Fig. 6) are analysed for the same 3-hour period using the two different time resolutions (1 minute and 5 minutes) and agree well for clouds with lifetimes larger than about 15 to 20 minutes. For shorter lifetimes, the 1-minute input data yield significantly substantially more tracked cells. It is obvious that we can only properly represent and analyse cloud lifetimes for clouds that exist over a certain number of output timesteps time steps in this framework. An individual cloud that is tracked for 5 to 10 minutes based on 1-minute output allows for robust conclusions about the evolution of the cloud in that period. The same time would merely lead to two or three individually identified objects for 5-minute data output, which would be the minimum to draw any useful conclusions about the lifetimes or time evolution of the clouds.

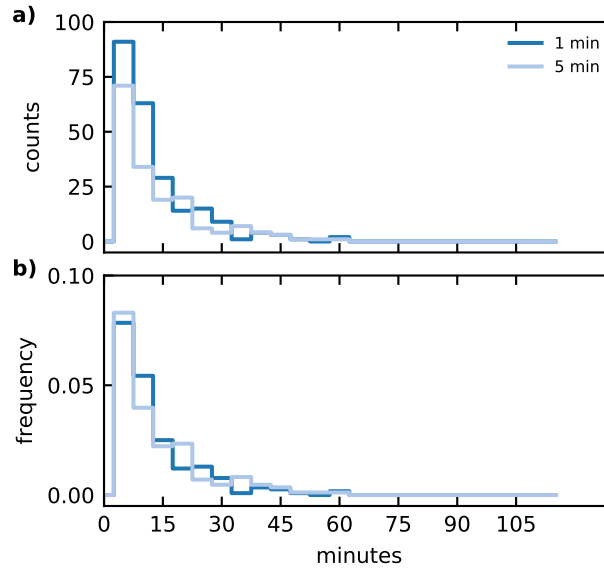


Figure 6. Cell lifetimes for tracking and analysis using two different output ~~timesteps~~time steps (1 minute and 5 minutes) showing both total counts (a) and the PDF (b).

4 Example B: tracking of deep convective clouds in model simulations and geostationary satellite data based on outgoing longwave radiation (OLR)

Satellite retrievals are an important tool in climate and weather research as they are an effective way of obtaining observation-based quantities over greater spatial scales in the atmosphere. Specifically, geostationary satellites offer continuous coverage in space and time for a specific region and can, therefore, be used for understanding the temporal evolution of atmospheric phenomena. Direct comparisons of model simulations with satellite retrievals for the same area and time period are an important means of assessing the models' capabilities to successfully represent atmospheric processes in the real world. Using a tracking framework for the analysis allows us to investigate the representation of clouds in the model in a way that takes the development of individual clouds within the population of clouds into account as opposed to relying on temporal and spatial statistics of the cloud field. Using the same tracking framework for both model and observation data allows for a more robust comparison between them.

Here, we use satellite data from the Geostationary Operational Environmental Satellite (GOES) system, specifically GOES-13 (Hillger and Schmit, 2007), and WRF model simulation results from the ACPC deep convection case study (see Sect. 3) (van den Heever et al., 2017). The satellite data were downloaded from the NOAA Comprehensive Large Array-data Stewardship System (CLASS) (NOAA, 2019a) for the Continental United States (CONUS) area in NetCDF format. The NOAA Weather and Climate Toolkit (WCT) (NOAA, 2019b) was used to convert pixel counts to radiances and brightness temperatures for the two channels used in the analysis here. The satellite data used in this example has an average horizontal spacing of about 4 km.

The model simulation comprises the outermost nested grid of the nested WRF ~~simulations used to drive the simulations in the inner domain that we used for the cloud tracking in the first example application (simulation setup described in Sect. 3).~~ This outer domain covers a much larger area, encompassing most of Texas and the surrounding states of the southern USA, as well as neighbouring areas of northeastern Mexico. ~~The simulations were performed with~~ It features a grid spacing of 4.5 km and ~~used explicit convection without a cumulus parametrisation. The simulations have been performed for 24 hours a width of 400 grid cells, equivalent to 1800 km in each horizontal direction. The simulation results were output at a time resolution of 15 minutes for the entire 24-hour simulation period from 12:00UTC on the UTC on 19 June 2013 to 12:00UTC on the UTC on 20 June 2013 and a square domain of 500 by 500 grid cells centred around Houston, Texas-2013.~~

~~To be able to perform~~ Although the temporal and spatial resolution of the input data can be arbitrary for the use in tobac, a meaningful comparison of the two datasets ~~, it has to be ensured requires~~ that the analysis covers the same region at a similar temporal and spatial resolution. The ~~satellite data were restricted to the temporal and spatial extent as~~ spatial resolution of the two datasets is reasonably similar (around 4 km for the satellite data and 4.5 km for the model output data. Furthermore, the) and both datasets use a regular 15-minute interval ~~available from the satellite data was used for both datasets,~~ with a difference of up to a minute due to the scan time of the satellite data. The satellite data were restricted to the same temporal and spatial extent as the model output.

Top of the atmosphere outgoing longwave radiation (OLR) is used to track individual deep convective clouds in both model simulations and satellite retrievals. OLR is a standard model output for most high-resolution simulations and is often used as a diagnostic for simulated deep convection (Pearson et al., 2010; Russo et al., 2011). OLR retrievals also have the benefit that they do not depend on other aspects of a complicated radiative transfer model, which require, amongst other assumptions, that pixels are assigned as either cloud or cloud-free for the radiative retrieval of several optical (effective radius and optical depth) and thermal (cloud top temperature and height) cloud properties (McGarraugh et al., 2018). For the satellite data, we use an empirical conversion derived in Singh et al. (2007) to convert the radiances L from two channels in the GOES-13 measurements, the water vapour channel (WV, 5.8 to 7.30 μm) and a channel in the infrared window (WIN, 10.2 to 11.2 μm), to OLR.

$$OLR = 11.44 L_{\text{WIN}} + 9.04 L_{\text{WV}} + \frac{9.11 L_{\text{WV}}}{L_{\text{WIN}}} - \frac{86.36}{L_{\text{WIN}}} - 0.14 L_{\text{WV}}^2 + 111.12. \quad (4)$$

Singh et al. (2007) report an uncertainty from these conversions within 2.5 W m^{-2}

The distribution of OLR for the model simulations and the satellite retrievals show a very similar shape (Fig. 7). The satellite-retrieved OLR features a larger number of pixels characterised by lower OLR values in the range between 100 and 250 W m^{-2} corresponding to deep cloud tops. The range covered and the peak position of OLR, corresponding to cloud-free and low cloud regions around 290 W m^{-2} , agree well between the model simulation and the satellite retrieval.

We use these histograms to choose the threshold values for the feature detection and the segmentation steps in the tobac routine. The threshold for the outline of the convective clouds in the segmentation step (250 W m^{-2}) reflects the lower tail of the

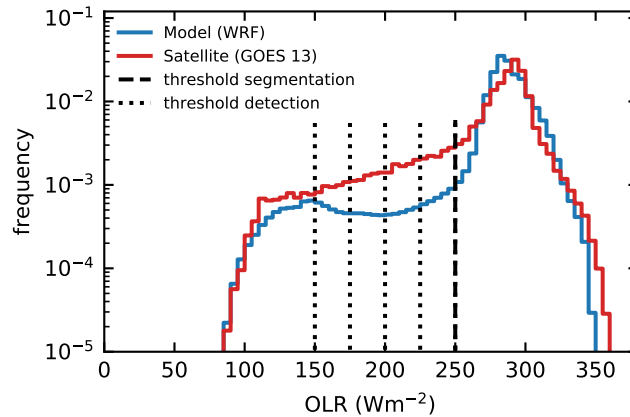


Figure 7. Probability density function of OLR for the model simulation and the satellite retrievals including the thresholds (vertical dashed and dotted lines) set for feature detection and segmentation.

peak of OLR in both the model simulations and the satellite retrievals. The additional thresholds used in the feature detection algorithm ($250, 225, 200, 175$ and 150 W m^{-2}) are distributed over the range of OLR values in the part of the distribution representing the deeper clouds.

- 15 The individual steps of the tracking analysis for the model data are shown in Fig. 8, but the same steps are applied equally to the satellite-retrieved data. The outgoing longwave radiation field (Fig. 8a) is filtered with a Gaussian filter with a standard deviation of $\sigma = 4.5 \text{ km}$, equivalent to the grid spacing of the model data (Fig. 8b). The feature identification following Sect. 2.2 is performed with the set of five OLR thresholds of $250, 225, 200, 175$ and 150 W m^{-2} (Fig. 8c,d). The segmentation is performed using the watershedding technique (Sect. 2.2.3) with an OLR threshold of 250 W m^{-2} to identify the area of the
- 20 individual clouds leading to the cloud areas shown in 8e. The complete linked trajectories of all clouds present at the specific [timestep](#) [time step](#), as illustrated in the other sub-figures, are shown in Fig. 8f with the cloud extent at the start (dashed) and end (solid) of the lifetime of the cloud.

[The data processing was performed on the JASMIN data analysis facility](#) (CEDA, 2019). [The total processing time of the script including feature detection, segmentation trajectory linking and saving the output data was around 1 minute with a maximum memory footprint of around 500 MB for the model data and 2.5 minutes with a memory footprint of around 400 MB for the](#)

- 5 [satellite data, each using a maximum of 3 processes and 27 threads. Additional tests of the processing on a typical laptop with 4 processing cores showed a similar processing time and memory footprint.](#) A smaller subset of the data and [the](#) analysis for this example including the tracking analysis and visualisation is available as a Jupyter notebook as part of the package source code.

The tracked clouds for both the model simulation and the satellite retrieval are visualised for two different times in Fig. 9.

- 10 Both the model simulations and the satellite retrieval show many individual convective clouds in a region north of the coastline, especially towards the east of the analysed domain around the Mississippi River Delta and further inland in Texas. In addition,

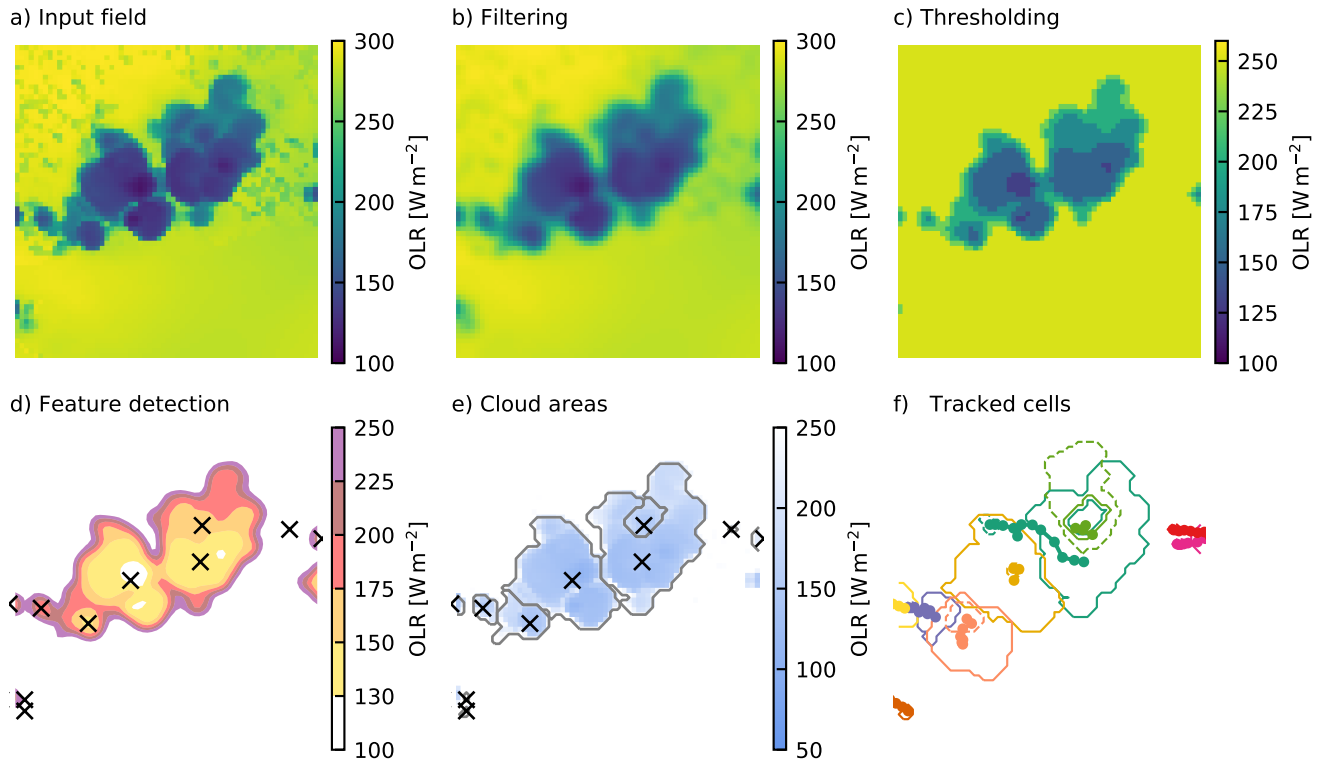


Figure 8. Schematic overview of the individual steps of the tracking algorithm for an example subset of the domain used in example B based on outgoing longwave radiation. The input data (a) are smoothed with a filter (b) before regions above or below a set of thresholds are determined (c) to identify the individual features (d). (e) shows the associated cloud areas determined in the segmentation step and (f) shows all individual clouds present at the time step over their entire life cycle, including outline the cloud area at the start (dashed) and at the end (solid) of the trajectory.

larger connected regions of clouds occur both towards the southern end of the analysed domain over the Gulf of Mexico and in the form of a large organised storm system entering the domain from the north-west. The propagation of this large system is not represented accurately in the model simulation, as it shows a lag of several hours and is smaller in magnitude than in the satellite retrievals. The lifetime distribution of the clouds identified and tracked from the model simulations and from the satellite retrievals show a similar distribution (see Fig. 10a). However, more clouds are identified in the satellite data than in the model data. When normalised for total number, the lifetime distributions agree better between the two different data inputs (Fig. 10b). Most cloud objects are tracked for periods of up to an hour, but in both the model simulations and the satellite retrievals there are numerous cloud objects tracked for up to several hours. The distributions of the cloud areas (Fig. 10c,d) show that the total cloud area for both model and satellite data is made up of two types of identified objects, smaller tracked clouds with a radius of up to 100 km and large tracked features with a radius of a few hundred kilometres. Due to the larger number of tracked clouds, there is more total cloud area in the tracked clouds in the satellite data. The distribution of cloud

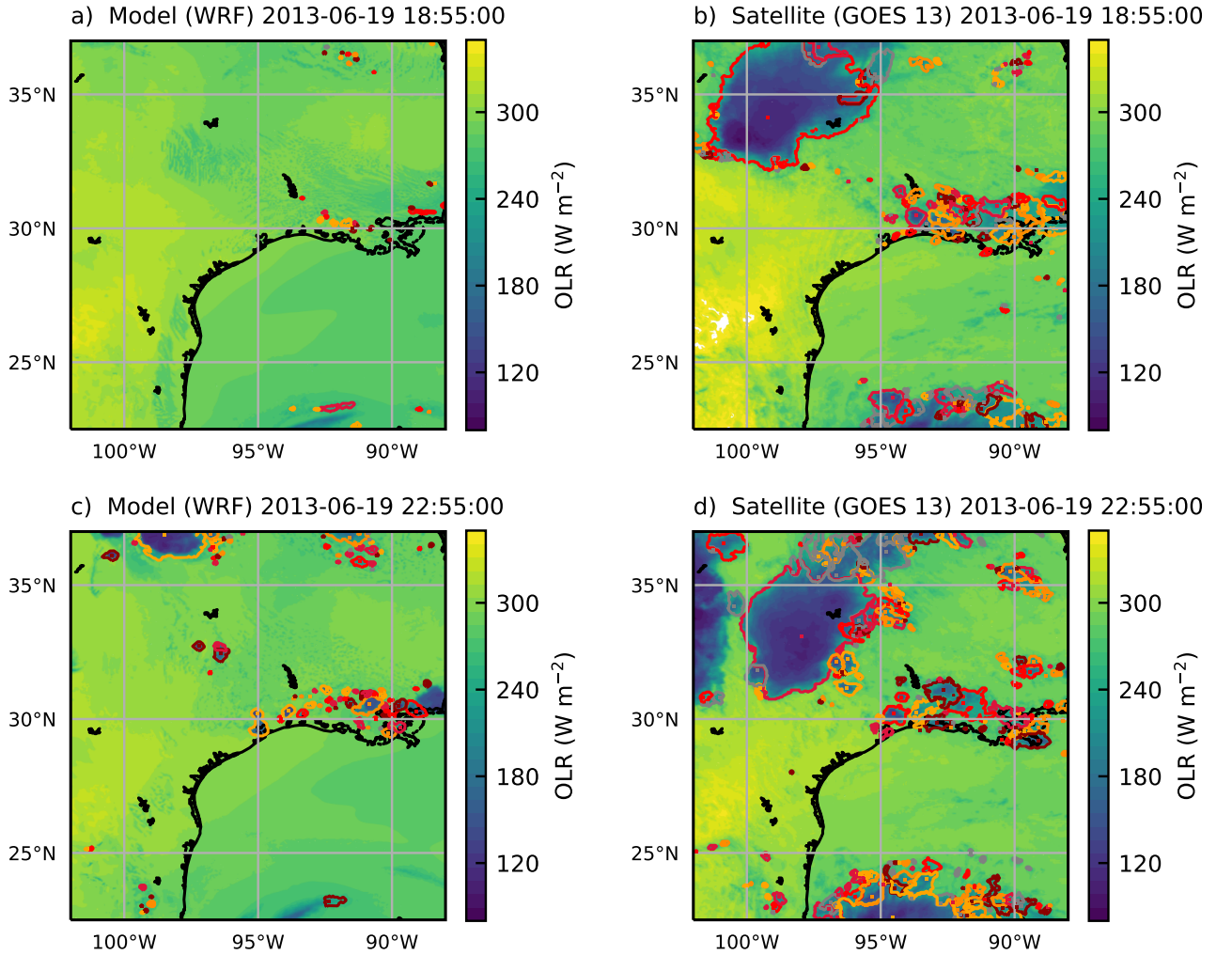


Figure 9. Identified and tracked objects at two specific points in time (19/06/2013 18:55 UTC and 22:55 UTC) based on outgoing long-wave radiation for the WRF simulations with 4.5 km grid spacing on the left (a, c) and the outgoing longwave radiation derived from the combination of two GOES-13 channels following Singh et al. (2007) on the right (b, d).

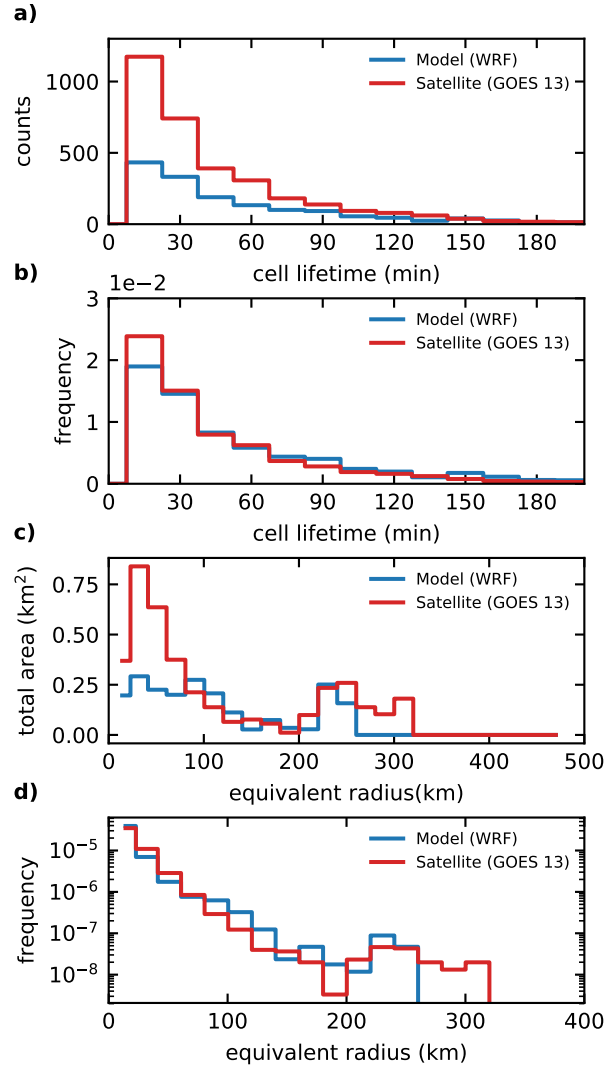


Figure 10. Distributions of cloud lifetimes obtained from the tracking of model data and satellite retrievals, shown as total counts (a) and frequency (b). The distribution of cloud areas is shown as the distribution of total area resulting from the sum in each area bin (d) and as a pdf of cloud area (c). Both these distributions are plotted against the equivalent radius of a circular cloud of the same area.

sizes is relatively similar between the two datasets. The satellite data show more small clouds below 100 km equivalent radius. Furthermore, the size of the largest tracked objects is larger in the satellite data than in the model data, which corresponds to the large MCS propagating through the domain of interest (Fig. 9), and which is not represented properly in the model simulations with respect to both timing and total size.

An analysis of the cloud velocities and nearest-neighbour distances as described in Sect. 3.1 is presented in Fig. 11. The distribution of both the nearest-neighbour distances (Fig. 11a) and the cloud displacement velocities (Fig. 11b) agree well between the model simulations and the satellite retrieval. The peak of the nearest-neighbour distances appears around 20 km. The propagation velocities peak at around 8 m s^{-1} , with most of the velocities below 20 m s^{-1} . A comparison of the nearest-neighbour distances and the displacements per input ~~timestep~~time step that would result for different temporal resolution (1, 5, 15 and 30 minutes) shows that the 15-minute ~~timestep~~time step used here already shows some overlap in the distributions. Longer ~~timesteps~~time steps of 30 minutes or more would probably lead to problems in the tracking, while shorter ~~timesteps~~time steps of a few minutes would be expected to improve the tracking further. However, output at similarly high temporal frequencies is not always feasible or simply not available for a lot of data sources, e.g. for the GOES-13 geostationary satellite retrievals used in this study. The newest generation of geostationary satellite imagers such as the GOES-R series (GOES 16/17) that has replaced the GOES-13 satellite used here, as well as Himawari-8 (Bessho et al., 2016) and the future Meteosat Third Generation (MTG) satellites (Stuhlmann et al., 2005) all feature ~~significantly~~substantially higher temporal and spatial resolution. The scattered convective cells of differing depths over the area of Houston that were the focus of the analysis in the first application example (Sect. 3) are not clearly resolved in these two datasets. The lower spatial resolution of the simulations and satellite retrieval (around 4 km compared to 500 m in the high-resolution simulations used in Sect. 3) limit the spatial scale of cloud features that can be resolved to more than a few tens of kilometres in radius. The use of outgoing longwave radiation as a variable for feature identification does not include as much information as the three-dimensional model output fields used in Sect. 3, however, it provides complementary information to compare model simulations with satellite retrievals.

5 Conclusions

We have presented tobac, a new framework for object-based analysis and tracking individual convective clouds in different types of input data. The workflow of the software package consists of the detection of suitable features, segmentation of the areas or volumes representative of an individual cloud object and subsequent linking of objects at individual time steps into trajectories. All individual steps are implemented in a modular way, thereby allowing for the implementation of different algorithms for each of the steps, should the need arise.

We have developed a feature detection algorithm based on identifying regions above/below a defined sequence of thresholds in two-dimensional input fields. Cloud volumes or cloud areas are associated based on a watershedding technique featuring a single specific threshold value on two- or three-dimensional input fields.

We have shown how we can leverage another open-source Python package ~~trackpy~~trackpy, initially developed for applica-

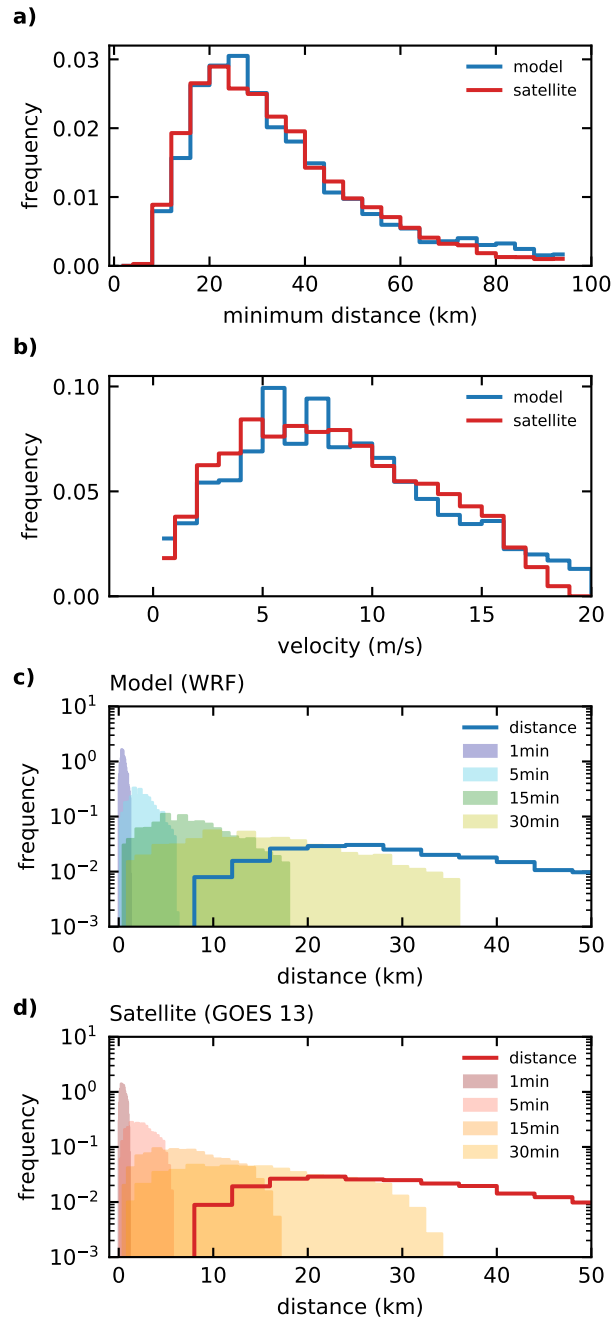


Figure 11. (a) Distributions of the distance to the next identified object for all identified objects and (b) velocities for tracked cloud objects at each timestep of the trajectories for both the model simulations and the satellite data. (d) The travel distance per input interval resulting from different time resolution of the input based on these velocities (b) is shown together with the distribution of the minimal distance to the nearest-neighbour in (a) for the model data in (c) and for the satellite data.

tion in microscopy, in the tobac framework to link up cloud objects at individual ~~timesteps~~time steps into consistent cloud trajectories. These cloud trajectories allow for an analysis of cloud lifetimes and the time evolution of cloud properties and physical processes in the clouds over the lifetime of the cloud. The analysis routines provided as part of the package can be
15 applied to derive cloud properties and statistics for individual clouds over their life-cycle as well as for the entire population of clouds in the analysed cloud field. The built-in visualisation routines allow for a convenient way to assess the performance of the analysis and evaluate the choice of parameters for the different steps of the analysis framework. The automatically created animated visualisations of individual tracked cells can guide users in the development of further detailed analyses based on the analysis tools provided in the framework.

20 The implementation of the tracking framework in Python enables the use of extensive and actively developed open source libraries for scientific computing. We have shown that this provides numerous advantages, e.g. for memory management, data structures and visualisation. The rapid development of the underlying libraries means that tobac can profit from future advances without any further development of tobac and any requirements on the side of the user. The modular structure of the framework allows for the inclusion of other existing or newly developed methods for the individual steps of feature detection,
25 object segmentation and tracking into the software package in the future. These capabilities enable the use of different tracking algorithms in parallel for evaluation and comparisons as well as tracking based on different types of input data in a single analysis framework.

We have presented two application examples of the use of tobac for the study of deep convective clouds. In the first application (example A), we have tracked scattered deep convective cells based on a combination of the vertical velocity and total
30 condensate mixing ratio fields from CRM simulations with WRF over the area around Houston, Texas. The simulations were performed with a grid spacing of 500 m, and thus represent a typical application of a CRM. The tracking framework is currently being applied to other CRMs for the same case study as part of the ACPC deep convection case study (van den Heever et al., 2017) to investigate the response of deep convective clouds in models to changes in aerosols. We have performed the tracking for different output frequencies to evaluate the dependency of the tracking performance on the time resolution of the input data.
35 The output resolutions of 1 minute and 5 minutes lead to comparable tracking results for scattered convective cells. This result can be confirmed using an analysis of typical displacement velocities of the clouds and nearest-neighbour distances between the individual identified cloud objects.

In a second application (example B), we have presented a simultaneous tracking of deep convective cloud features and larger convective systems based on outgoing longwave radiation output from model simulations with convection-permitting grid spacing
5 (4.5 km) and outgoing longwave radiation derived from geostationary satellite retrievals (GOES-13) in the same region. The 15-minute time resolution available from the satellite retrieval is shown to be sufficient for successful tracking performance. The analysis also demonstrated that the model simulations and the satellite retrieval feature clouds with a similar lifetime distribution. The distribution of cloud areas in model and satellite data shows a similar combination of smaller convective cells and larger systems. The main differences occur for the largest tracked systems, which are stronger in the satellite retrievals.
10 This can be explained by the limited representation of the propagation of two large organised storms within the model domain. This would have been more challenging to assess from a bulk analysis of the domain-wide averaged properties.

The newest generation of geostationary satellites, such as Himawari-8 or GOES-16/17, provide ~~significantly~~substantially higher spatial and temporal resolution (Bessho et al., 2016; Schmit et al., 2016). These advances will strongly improve the applicability of this type of satellite data for use in object-based tracking and analyses with tobac, and also allows for a wider

15 range of applications, e.g. by capturing smaller scattered cells such as the ones investigated in Sect. 3.

The ability of tobac to be used for both models and observations as shown in these examples helps to ~~more directly~~ compare models with observations more directly, and therefore, better understand the differences between the two types of data.

Although we have focused on tracking and analysing deep convection here, there are numerous other applications that tobac can be used for without much additional work. There are a large number of existing data products, such as high-resolution

20 radar data, e.g. from NEXRAD over the United States or similar networks in several other regions of the world (Reed et al., 2017), that would be most suited for the use with tobac. Furthermore, the application of tobac is not strictly limited to the analysis of clouds, and it can also be applied to study other features of the Earth system that can be identified as well-defined ~~time-evolving~~time-evolving regions, such as distinct aerosol plumes in the atmosphere or plankton in the surface layer of the ocean.

25 We are currently working on implementing additional algorithms for the modular steps of the framework, e.g. based on the analyses developed in Senf et al. (2018). Additionally, we are implementing a more flexible representation of the links between cloud objects at specific points in time, which will allow for ~~a~~ proper treatment of more complex splitting and merging of cells. We invite the community to contribute to the future development of tobac both through the implementation of existing algorithms into the common framework and by using the framework as a basis for new developments.

30 *Code and data availability.* The tobac source code publicly available in a GitHub repository distributed under a BSD 3-Clause license at <https://github.com/climate-processes/tobac> (Heikenfeld, 2019c, d). The version tobac 1.2 described here is available as a release (Heikenfeld, 2019a). The latest version of tobac can be installed using conda with the command `conda install -c conda-forge tobac`. The linking step makes use of `trackpy` (Allan et al., 2016), which is available on GitHub (<https://github.com/soft-matter/trackpy>). We use several standard Python packages for scientific computing and image processing that are all available through package managers such as pip or conda. The GOES-13 satellite imager data has been downloaded from the NOAA Comprehensive Large Array-data Stewardship System

5 (CLASS) (NOAA, 2019a) and processed with the NOAA Weather and Climate Toolkit (WCT) (NOAA, 2019b). Jupyter notebooks containing the tracking analysis and visualisations of the tracking results for a smaller subsample of the data used in the two example applications are provided as part of the tobac source code. The data used in these notebooks is downloaded automatically and is available as Heikenfeld (2019b) and on GitHub at https://github.com/climate-processes/tobac_example_data.

Author contributions. M. Heikenfeld lead the initial development of tobac with contributions by P.J. Marinescu, D. Watson-Parris and F. Senf. M. Christensen processed the geostationary satellite data and contributed to the analysis of the tracking based on outgoing longwave radiation. M. Heikenfeld performed the data analysis and wrote the manuscript with contributions and approval of the final version by P.J.

5 Marinescu., P. Stier., S.C. van den Heever, M. Christensen., D. Watson-Parris and F. Senf.

Acknowledgements. M. Heikenfeld acknowledges funding from the NERC Oxford DTP in Environmental Research (NE/L002612/1). The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) project BACCHUS under grant agreement n° 603445 and the European Research Council under the European Union's Horizon 2020 research and innovation program with Grant Agreement n° 724602 (RECAP). S.C. van den Heever and P.J. Marinescu acknowledge funding from the NASA CAMPEX project under grant 80NSSC18K0149, and P.J. Marinescu was ~~supported~~also partially by a National Science Foundation Graduate Research Fellowship Grant DGE-1321845. F. Senf acknowledges funding within the High Definition Clouds and Precipitation for advancing Climate Prediction (HD(CP)²) project funded by the BMBF (German Ministry for Education and Research) under grant 01LK1507C.

The research has been performed in the framework of the IGBP/WCRP initiative ACPC (Aerosols Clouds Precipitation and Climate) with a focus on developing analysis techniques for the deep convection case study over Houston, Texas. ~~Computations and data processing have been performed on the ARCHER and JASMIN computing facilities~~This work used the ARCHER UK National Supercomputing Service and JASMIN, the UK collaborative data analysis facility.

Thanks go to William Jones and other contributors to the tobac repository on GitHub for fruitful discussions and contributions to the development of tobac. The authors would like to thank two anonymous reviewers for their useful comments during the review stage of this paper.

References

- Allan, D., Caswell, T., Keim, N., and van der Wel, C.: Trackpy: Trackpy v0.3.2, Zenodo, <https://doi.org/10.5281/zenodo.60550>, <https://zenodo.org/record/60550#.XCpcrcbTVhF>, 2016.
- Autonès, F. and Moisselin, J. M.: Algorithm Theoretical Basis Document for “Rapid Development Thunderstorms”(RDT-PGE11 v3.0), Tech. rep., SAF/NWC/CDOP/MFT/SCI/ATBD/11, http://www.nwcsaf.org/AemetWebContents/ScientificDocumentation/Documentation/MSG/SAF-NWC-CDOP2-MFT-SCI-VR-11_v3.0.pdf, 2013.
- Bacmeister, J. T. and Stephens, G. L.: Spatial Statistics of Likely Convective Clouds in CloudSat Data, *Journal of Geophysical Research: Atmospheres*, 116, <https://doi.org/10.1029/2010JD014444>, 2011.
- Bessho, K., Date, K., Hayashi, M., Ikeda, A., Imai, T., Inoue, H., Kumagai, Y., Miyakawa, T., Murata, H., Ohno, T., Okuyama, A., Oyama, R., Sasaki, Y., Shimazu, Y., Shimoji, K., Sumida, Y., Suzuki, M., Taniguchi, H., Tsuchiyama, H., Uesawa, D., Yokota, H., and Yoshida, R.: An Introduction to Himawari-8/9— Japan’s New-Generation Geostationary Meteorological Satellites, *Journal of the Meteorological Society of Japan. Ser. II*, 94, 151–183, <https://doi.org/10.2151/jmsj.2016-009>, 2016.
- CEDA: JASMIN, the UK Collaborative Data Analysis Facility, <http://jasmin.ac.uk/>, 2019.
- Chen, Q., Koren, I., Altaratz, O., Heiblum, R. H., Dagan, G., and Pinto, L.: How Do Changes in Warm-Phase Microphysics Affect Deep Convective Clouds?, *Atmos. Chem. Phys.*, 17, 9585–9598, <https://doi.org/10.5194/acp-17-9585-2017>, 2017.
- Cotton, W. R., Bryan, G., and van den Heever, S. C.: *Storm and Cloud Dynamics*, Academic Press, 2010.
- Couvreur, F., Hourdin, F., and Rio, C.: Resolved Versus Parametrized Boundary-Layer Plumes. Part I: A Parametrization-Oriented Conditional Sampling in Large-Eddy Simulations, *Boundary-Layer Meteorology*, 134, 441–458, <https://doi.org/10.1007/s10546-009-9456-5>, 2010.
- Crane, R.: Automatic Cell Detection and Tracking, *IEEE Transactions on Geoscience Electronics*, 17, 250–262, <https://doi.org/10.1109/TGE.1979.294654>, 1979.
- Crocker, J. C. and Grier, D. G.: Methods of Digital Video Microscopy for Colloidal Studies, *Journal of Colloid and Interface Science*, 179, 298–310, <https://doi.org/10.1006/jcis.1996.0217>, 1996.
- Dask Development Team: Dask: Library for Dynamic Task Scheduling, <http://dask.pydata.org>, 2016.
- Davis, C., Brown, B., and Bullock, R.: Object-Based Verification of Precipitation Forecasts. Part II: Application to Convective Rain Systems, *Monthly Weather Review*, 134, 1785–1795, <https://doi.org/10.1175/MWR3146.1>, 2006.
- Davis, C. A., Brown, B. G., Bullock, R., and Halley-Gotway, J.: The Method for Object-Based Diagnostic Evaluation (MODE) Applied to Numerical Forecasts from the 2005 NSSL/SPC Spring Program, *Weather and Forecasting*, 24, 1252–1267, <https://doi.org/10.1175/2009WAF2222241.1>, 2009.
- Dawe, J. T. and Austin, P. H.: Statistical Analysis of an LES Shallow Cumulus Cloud Ensemble Using a Cloud Tracking Algorithm, *Atmos. Chem. Phys.*, 12, 1101–1119, <https://doi.org/10.5194/acp-12-1101-2012>, 2012.
- Dixon, M. and Wiener, G.: TITAN: Thunderstorm Identification, Tracking, Analysis, and Nowcasting—A Radar-Based Methodology, *Journal of Atmospheric and Oceanic Technology*, 10, 785–797, [https://doi.org/10.1175/1520-0426\(1993\)010<0785:TTITAA>2.0.CO;2](https://doi.org/10.1175/1520-0426(1993)010<0785:TTITAA>2.0.CO;2), 1993.
- Doswell, C. A.: Severe Convective Storms—An Overview, in: *Severe Convective Storms*, edited by Doswell, C. A., Meteorological Monographs, pp. 1–26, American Meteorological Society, Boston, MA, https://doi.org/10.1007/978-1-935704-06-5_1, https://doi.org/10.1007/978-1-935704-06-5_1, 2001.
- Emanuel, K. A.: *Atmospheric Convection*, Oxford University Press, New York, 1994.

- Fan, J., Han, B., Varble, A., Morrison, H., North, K., Kollias, P., Chen, B., Dong, X., Giangrande, S. E., Khain, A., Lin, Y., Mansell, E., Milbrandt, J. A., Stenz, R., Thompson, G., and Wang, Y.: Cloud-Resolving Model Intercomparison of an MC3E Squall Line Case: Part I—Convective Updrafts, *Journal of Geophysical Research: Atmospheres*, 122, 9351–9378, <https://doi.org/10.1002/2017JD026622>, 2017.
- 25 Feng, Z., Dong, X., Xi, B., McFarlane, S. A., Kennedy, A., Lin, B., and Minnis, P.: Life Cycle of Midlatitude Deep Convective Systems in a Lagrangian Framework, *Journal of Geophysical Research: Atmospheres*, 117, D23 201, <https://doi.org/10.1029/2012JD018362>, 2012.
- Feng, Z., Leung, L. R., Houze Jr., R. A., Hagos, S., Hardin, J., Yang, Q., Han, B., and Fan, J.: 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, 1470–1494, <https://doi.org/10.1029/2018MS001305>, 2018.
- 30 Fiolleau, T. and Roca, R.: An Algorithm for the Detection and Tracking of Tropical Mesoscale Convective Systems Using Infrared Images From Geostationary Satellite, *IEEE Transactions on Geoscience and Remote Sensing*, 51, 4302–4315, <https://doi.org/10.1109/TGRS.2012.2227762>, 2013.
- Fritsch, J. M. and Forbes, G. S.: Mesoscale Convective Systems, in: *Severe Convective Storms*, edited by Doswell, C. A., *Meteorological Monographs*, pp. 323–357, American Meteorological Society, Boston, MA, https://doi.org/10.1007/978-1-935704-06-5_9, https://doi.org/10.1007/978-1-935704-06-5_9, 2001.
- 35 Fu, H., Shen, Y., Liu, J., He, G., Chen, J., Liu, P., Qian, J., and Li, J.: Cloud Detection for FY Meteorology Satellite Based on Ensemble Thresholds and Random Forests Approach, *Remote Sensing*, 11, 44, <https://doi.org/10.3390/rs11010044>, 2019.
- Gensini, V. A. and Mote, T. L.: Estimations of Hazardous Convective Weather in the United States Using Dynamical Downscaling, *Journal of Climate*, 27, 6581–6589, <https://doi.org/10.1175/JCLI-D-13-00777.1>, 2014.
- Grady, L.: Random Walks for Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 1768–1783, <https://doi.org/10.1109/TPAMI.2006.233>, 2006.
- Guillaume, A., Kahn, B. H., Yue, Q., Fetzer, E. J., Wong, S., Manipon, G. J., Hua, H., and Wilson, B. D.: Horizontal and Vertical Scaling of Cloud Geometry Inferred from CloudSat Data, *Journal of the Atmospheric Sciences*, 75, 2187–2197, [https://doi.org/10.1175/JAS-D-17-](https://doi.org/10.1175/JAS-D-17-0111.1)
- 5 0111.1, 2018.
- Hagos, S., Feng, Z., McFarlane, S., and Leung, L. R.: Environment and the Lifetime of Tropical Deep Convection in a Cloud-Permitting Regional Model Simulation, *Journal of the Atmospheric Sciences*, 70, 2409–2425, <https://doi.org/10.1175/JAS-D-12-0260.1>, 2013.
- Heiblum, R. H., Altaratz, O., Koren, I., Feingold, G., Kostinski, A. B., Khain, A. P., Ovchinnikov, M., Fredj, E., Dagan, G., Pinto, L., Yaish, R., and Chen, Q.: Characterization of Cumulus Cloud Fields Using Trajectories in the Center of Gravity versus Water Mass
- 10 Phase Space: 2. Aerosol Effects on Warm Convective Clouds, *Journal of Geophysical Research: Atmospheres*, 121, 2015JD024 193, <https://doi.org/10.1002/2015JD024193>, 2016a.
- Heiblum, R. H., Altaratz, O., Koren, I., Feingold, G., Kostinski, A. B., Khain, A. P., Ovchinnikov, M., Fredj, E., Dagan, G., Pinto, L., Yaish, R., and Chen, Q.: Characterization of Cumulus Cloud Fields Using Trajectories in the Center of Gravity versus Water Mass
- 15 Phase Space: 1. Cloud Tracking and Phase Space Description, *Journal of Geophysical Research: Atmospheres*, 121, 2015JD024 186, <https://doi.org/10.1002/2015JD024186>, 2016b.
- Heikenfeld, M.: Tobac 1.2: Tracking and Object-Based Analysis of Clouds, *Zenodo*, <https://doi.org/10.5281/zenodo.3408268>, <https://zenodo.org/record/3408268>, 2019a.
- Heikenfeld, M.: Tobac Example Datasets, <https://doi.org/10.5281/zenodo.3195910>, <https://zenodo.org/record/3195910>, 2019b.
- Heikenfeld, M.: Tobac: Tracking and Object-Based Analysis of Clouds, *climate-processes*, <https://github.com/climate-processes/tobac>,
- 20 2019c.

- Heikenfeld, M.: Tobac: Tracking and Object-Based Analysis of Clouds, Zenodo, <https://doi.org/10.5281/zenodo.2577046>, 2019d.
- Hernandez-Deckers, D. and Sherwood, S. C.: A Numerical Investigation of Cumulus Thermals, *Journal of the Atmospheric Sciences*, 73, 4117–4136, <https://doi.org/10.1175/JAS-D-15-0385.1>, 2016.
- Heus, T. and Seifert, A.: Automated Tracking of Shallow Cumulus Clouds in Large Domain, Long Duration Large Eddy Simulations, *Geosci. Model Dev.*, 6, 1261–1273, <https://doi.org/10.5194/gmd-6-1261-2013>, 2013.
- Heus, T., Jonker, H. J. J., Van den Akker, H. E. A., Griffith, E. J., Koutek, M., and Post, F. H.: A Statistical Approach to the Life Cycle Analysis of Cumulus Clouds Selected in a Virtual Reality Environment, *Journal of Geophysical Research: Atmospheres*, 114, D06 208, <https://doi.org/10.1029/2008JD010917>, 2009.
- Hillger, D. W. and Schmit, T. J.: The GOES-13 Science Test: Imager and Sounder Radiance and Product Validations. NOAA, Environ. Satell. Data Inf. Serv., Silver Spring, MD, NOAA Tech. Rep, 141, 2007.
- Hoyer, S. and Hamman, J.: Xarray: N-D Labeled Arrays and Datasets in Python, *Journal of Open Research Software*, 5, 10, <https://doi.org/10.5334/jors.148>, 2017.
- Hunter, J. D.: Matplotlib: A 2D Graphics Environment, *Computing in Science Engineering*, 9, 90–95, <https://doi.org/10.1109/MCSE.2007.55>, 2007.
- Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., and Collins, W. D.: Radiative Forcing by Long-Lived Greenhouse Gases: Calculations with the AER Radiative Transfer Models, *Journal of Geophysical Research: Atmospheres*, 113, D13 103, <https://doi.org/10.1029/2008JD009944>, 2008.
- Igel, M. R., Drager, A. J., and van den Heever, S. C.: A CloudSat Cloud Object Partitioning Technique and Assessment and Integration of Deep Convective Anvil Sensitivities to Sea Surface Temperature, *Journal of Geophysical Research: Atmospheres*, 119, 10 515–10 535, <https://doi.org/10.1002/2014JD021717>, 2014.
- IPCC: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, <https://doi.org/10.1017/CBO9781107415324>, www.climatechange2013.org, 2013.
- Cluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., and Corlay, S.: Jupyter Notebooks-a Publishing Format for Reproducible Computational Workflows., in: *ELPUB*, pp. 87–90, 2016.
- Laing, A. G. and Fritsch, J. M.: The Global Population of Mesoscale Convective Complexes, *Quarterly Journal of the Royal Meteorological Society*, 123, 389–405, <https://doi.org/10.1002/qj.49712353807>, 1997.
- Lakshmanan, V. and Smith, T.: An Objective Method of Evaluating and Devising Storm-Tracking Algorithms, *Weather and Forecasting*, 25, 701–709, <https://doi.org/10.1175/2009WAF2222330.1>, 2009.
- Liang, K., Shi, H., Yang, P., and Zhao, X.: An Integrated Convective Cloud Detection Method Using FY-2 VISSR Data, *Atmosphere*, 8, 42, <https://doi.org/10.3390/atmos8020042>, 2017.
- Lin, J. W.-B.: Why Python Is the Next Wave in Earth Sciences Computing, *Bulletin of the American Meteorological Society*, 93, 1823–1824, <https://doi.org/10.1175/BAMS-D-12-00148.1>, 2012.
- Machado, L. a. T., Rossow, W. B., Guedes, R. L., and Walker, A. W.: Life Cycle Variations of Mesoscale Convective Systems over the Americas, *Monthly Weather Review*, 126, 1630–1654, [https://doi.org/10.1175/1520-0493\(1998\)126<1630:LCVOMC>2.0.CO;2](https://doi.org/10.1175/1520-0493(1998)126<1630:LCVOMC>2.0.CO;2), 1998.
- McGarraugh, G. R., Poulsen, C. A., Thomas, G. E., Povey, A. C., Sus, O., Stapelberg, S., Schlundt, C., Proud, S., Christensen, M. W., Stengel, M., Hollmann, R., and Grainger, R. G.: The Community Cloud Retrieval for CLimate (CC4CL); Part 2: The Optimal Estimation Approach, *Atmospheric Measurement Techniques*, 11, 3397–3431, <https://doi.org/https://doi.org/10.5194/amt-11-3397-2018>, 2018.

- McKinney, W.: Data Structures for Statistical Computing in Python, in: Proceedings of the 9th Python in Science Conference, pp. 51–56, <http://conference.scipy.org/proceedings/scipy2010/mckinney.html>, 2010.
- Mecikalski, J. R. and Bedka, K. M.: Forecasting Convective Initiation by Monitoring the Evolution of Moving Cumulus in Daytime GOES Imagery, *Monthly Weather Review*, 134, 49–78, <https://doi.org/10.1175/MWR3062.1>, 2006.
- 25 Mecikalski, J. R., Watts, P. D., and Koenig, M.: Use of Meteosat Second Generation Optimal Cloud Analysis Fields for Understanding Physical Attributes of Growing Cumulus Clouds, *Atmospheric Research*, 102, 175–190, <https://doi.org/10.1016/j.atmosres.2011.06.023>, 2011.
- Menzel, W. P.: Cloud Tracking with Satellite Imagery: From the Pioneering Work of Ted Fujita to the Present, *Bulletin of the American Meteorological Society*, 82, 33–48, [https://doi.org/10.1175/1520-0477\(2001\)082<0033:CTWSIF>2.3.CO;2](https://doi.org/10.1175/1520-0477(2001)082<0033:CTWSIF>2.3.CO;2), 2001.
- 30 Met Office: Iris: A Python Library for Analysing and Visualising Meteorological and Oceanographic Data Sets, Tech. rep., 2018.
- Meyer, F.: Topographic Distance and Watershed Lines, *Signal Processing*, 38, 113–125, [https://doi.org/10.1016/0165-1684\(94\)90060-4](https://doi.org/10.1016/0165-1684(94)90060-4), 1994.
- Morrison, H., Curry, J. A., and Khvorostyanov, V. I.: A New Double-Moment Microphysics Parameterization for Application in Cloud and Climate Models. Part I: Description, *Journal of the Atmospheric Sciences*, 62, 1665–1677, <https://doi.org/10.1175/JAS3446.1>, 2005.
- 35 Morrison, H., Thompson, G., and Tatarskii, V.: Impact of Cloud Microphysics on the Development of Trailing Stratiform Precipitation in a Simulated Squall Line: Comparison of One- and Two-Moment Schemes, *Monthly Weather Review*, 137, 991–1007, <https://doi.org/10.1175/2008MWR2556.1>, 2009.
- Moseley, C., Berg, P., and Haerter, J. O.: Probing the Precipitation Life Cycle by Iterative Rain Cell Tracking, *Journal of Geophysical Research: Atmospheres*, 118, 13,361–13,370, <https://doi.org/10.1002/2013JD020868>, 2013.
- Moseley, C., Hohenegger, C., Berg, P., and Haerter, J. O.: Intensification of Convective Extremes Driven by Cloud-Cloud Interaction, *Nature Geoscience*, 9, 748–752, <https://doi.org/10.1038/ngeo2789>, 2016.
- 5 NCEP: NCEP GDAS/FNL 0.25 Degree Global Tropospheric Analyses and Forecast Grids, Tech. rep., <https://doi.org/10.5065/D65Q4T4Z>, <https://rda.ucar.edu/datasets/ds083.3/>, 2015.
- Nesbitt, S. W., Zipser, E. J., and Cecil, D. J.: A Census of Precipitation Features in the Tropics Using TRMM: Radar, Ice Scattering, and Lightning Observations, *Journal of Climate*, 13, 4087–4106, [https://doi.org/10.1175/1520-0442\(2000\)013<4087:ACOPFI>2.0.CO;2](https://doi.org/10.1175/1520-0442(2000)013<4087:ACOPFI>2.0.CO;2), 2000.
- 10 Nesbitt, S. W., Cifelli, R., and Rutledge, S. A.: Storm Morphology and Rainfall Characteristics of TRMM Precipitation Features, *Monthly Weather Review*, 134, 2702–2721, 2006.
- NOAA: NOAA’s Comprehensive Large Array-Data Stewardship System - GOES Satellite Data - Imager (GVAR_IMG), https://www.avl.class.noaa.gov/saa/products/search?datatype_family=GVAR_IMG, 2019a.
- NOAA: NOAA Weather and Climate Toolkit (WCT), National Climatic Data Center, NESDIS, NOAA, <https://www.ncdc.noaa.gov/wct/>, 2019b.
- 15 O’Brien, T. A., Li, F., Collins, W. D., Rauscher, S. A., Ringler, T. D., Taylor, M., Hagos, S. M., and Leung, L. R.: Observed Scaling in Clouds and Precipitation and Scale Incognizance in Regional to Global Atmospheric Models, *Journal of Climate*, 26, 9313–9333, <https://doi.org/10.1175/JCLI-D-13-00005.1>, 2013.
- Orlanski, I.: A Rational Subdivision of Scales for Atmospheric Processes, *Bulletin of the American Meteorological Society*, 56, 527–530, <https://www.jstor.org/stable/26216020>, 1975.
- 20

- Pearson, K. J., Hogan, R. J., Allan, R. P., Lister, G. M. S., and Holloway, C. E.: Evaluation of the Model Representation of the Evolution of Convective Systems Using Satellite Observations of Outgoing Longwave Radiation, *Journal of Geophysical Research: Atmospheres*, 115, D20 206, <https://doi.org/10.1029/2010JD014265>, 2010.
- Perez, F. and Granger, B. E.: IPython: A System for Interactive Scientific Computing, *Computing in Science Engineering*, 9, 21–29, <https://doi.org/10.1109/MCSE.2007.53>, 2007.
- Perkel, J. M.: Programming: Pick up Python, *Nature News*, 518, 125, <https://doi.org/10.1038/518125a>, 2015.
- Plant, R. S.: Statistical Properties of Cloud Lifecycles in Cloud-Resolving Models, *Atmos. Chem. Phys.*, 9, 2195–2205, <https://doi.org/10.5194/acp-9-2195-2009>, 2009.
- Reed, J. L., Lanterman, A. D., and Trostel, J. M.: Weather Radar: Operation and Phenomenology, *IEEE Aerospace and Electronic Systems Magazine*, 32, 46–62, <https://doi.org/10.1109/MAES.2017.150178>, 2017.
- Riley, E. M., Mapes, B. E., and Tulich, S. N.: Clouds Associated with the Madden–Julian Oscillation: A New Perspective from CloudSat, *Journal of the Atmospheric Sciences*, 68, 3032–3051, <https://doi.org/10.1175/JAS-D-11-030.1>, 2011.
- Rocklin, M.: Dask: Parallel Computation with Blocked Algorithms and Task Scheduling, in: *Proceedings of the 14th Python in Science Conference*, edited by Huff, K. and Bergstra, J., pp. 130–136, 2015.
- Rosenfeld, D.: Objective Method for Analysis and Tracking of Convective Cells as Seen by Radar, *Journal of Atmospheric and Oceanic Technology*, 4, 422–434, [https://doi.org/10.1175/1520-0426\(1987\)004<0422:OMFAAT>2.0.CO;2](https://doi.org/10.1175/1520-0426(1987)004<0422:OMFAAT>2.0.CO;2), 1987.
- Russo, M. R., Marécal, V., Hoyle, C. R., Arteta, J., Chemel, C., Chipperfield, M. P., Dessens, O., Feng, W., Hosking, J. S., Telford, P. J., Wild, O., Yang, X., and Pyle, J. A.: Representation of Tropical Deep Convection in Atmospheric Models – Part 1: Meteorology and Comparison with Satellite Observations, *Atmospheric Chemistry and Physics*, 11, 2765–2786, <https://doi.org/10.5194/acp-11-2765-2011>, 2011.
- Schmit, T. J., Griffith, P., Gunshor, M. M., Daniels, J. M., Goodman, S. J., and Lehair, W. J.: A Closer Look at the ABI on the GOES-R Series, *Bulletin of the American Meteorological Society*, 98, 681–698, <https://doi.org/10.1175/BAMS-D-15-00230.1>, 2016.
- Senf, F. and Deneke, H.: Satellite-Based Characterization of Convective Growth and Glaciation and Its Relationship to Precipitation Formation over Central Europe, *Journal of Applied Meteorology and Climatology*, 56, 1827–1845, <https://doi.org/10.1175/JAMC-D-16-0293.1>, 2017.
- Senf, F., Dietzsch, F., Hünnerbein, A., and Deneke, H.: Characterization of Initiation and Growth of Selected Severe Convective Storms over Central Europe with MSG-SEVIRI, *Journal of Applied Meteorology and Climatology*, 54, 207–224, <https://doi.org/10.1175/JAMC-D-14-0144.1>, 2015.
- Senf, F., Klocke, D., and Brueck, M.: Size-Resolved Evaluation of Simulated Deep Tropical Convection, *Monthly Weather Review*, 146, 2161–2182, <https://doi.org/10.1175/MWR-D-17-0378.1>, 2018.
- Sherwood, S. C., Hernández-Deckers, D., Colin, M., and Robinson, F.: Slippery Thermals and the Cumulus Entrainment Paradox, *Journal of the Atmospheric Sciences*, 70, 2426–2442, <https://doi.org/10.1175/JAS-D-12-0220.1>, 2013.
- Sieglauff, J. M., Hartung, D. C., Feltz, W. F., Cronce, L. M., and Lakshmanan, V.: A Satellite-Based Convective Cloud Object Tracking and Multipurpose Data Fusion Tool with Application to Developing Convection, *Journal of Atmospheric and Oceanic Technology*, 30, 510–525, <https://doi.org/10.1175/JTECH-D-12-00114.1>, 2012.
- Singh, R., Thapliyal, P. K., Kishtawal, C. M., Pal, P. K., and Joshi, P. C.: A New Technique for Estimating Outgoing Longwave Radiation Using Infrared Window and Water Vapor Radiances from Kalpana Very High Resolution Radiometer, *Geophysical Research Letters*, 34, L23 815, <https://doi.org/10.1029/2007GL031715>, 2007.

- Skamarock, C., Klemp, B., Dudhia, J., Gill, O., Barker, M., Wang, W., and Powers, G.: A Description of the Advanced Research WRF Version 2, Tech. rep., <https://doi.org/10.5065/D6DZ069T>, <https://opensky.ucar.edu/islandora/object/technotes%3A479/>, 2005.
- Soille, P. J. and Anselmetti, M. M.: Automated Basin Delineation from Digital Elevation Models Using Mathematical Morphology, *Signal Processing*, 20, 171–182, [https://doi.org/10.1016/0165-1684\(90\)90127-K](https://doi.org/10.1016/0165-1684(90)90127-K), 1990.
- 25 Stevens, B. and Feingold, G.: Untangling Aerosol Effects on Clouds and Precipitation in a Buffered System, *Nature*, 461, 607–613, <https://doi.org/10.1038/nature08281>, 2009.
- Stuhlmann, R., Rodriguez, A., Tjemkes, S., Grandell, J., Arriaga, A., Bézy, J. L., Aminou, D., and Bensi, P.: Plans for EUMETSAT’s Third Generation Meteosat Geostationary Satellite Programme, *Advances in Space Research*, 36, 975–981, <https://doi.org/10.1016/j.asr.2005.03.091>, 2005.
- 30 Terwey, W. D. and Rozoff, C. M.: Objective Convective Updraft Identification and Tracking: Part 1. Structure and Thermodynamics of Convection in the Rainband Regions of Two Hurricane Simulations, *Journal of Geophysical Research: Atmospheres*, 119, 6470–6496, <https://doi.org/10.1002/2013JD020904>, 2014.
- Trenberth, K. E., Fasullo, J. T., and Kiehl, J.: Earth’s Global Energy Budget, *Bulletin of the American Meteorological Society*, 90, 311–324, <https://doi.org/10.1175/2008BAMS2634.1>, 2009.
- 35 van den Heever, S. C., Fridlind, A. M., Marinescu, P. J., Heikenfeld, M., White, B., and Stier, P.: Aerosol-Cloud-Precipitation-Climate (ACPC) Initiative: Deep Convective Cloud Group Roadmap, http://acpcinitiative.org/Docs/ACPC_DCC_Roadmap_171019.pdf, 2017.
- van der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy Array: A Structure for Efficient Numerical Computation, *Computing in Science Engineering*, 13, 22–30, <https://doi.org/10.1109/MCSE.2011.37>, 2011.
- van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T.: Scikit-Image: Image Processing in Python, *PeerJ*, 2, e453, <https://doi.org/10.7717/peerj.453>, 2014.
- Varble, A., Fridlind, A. M., Zipser, E. J., Ackerman, A. S., Chaboureaud, J.-P., Fan, J., Hill, A., McFarlane, S. A., Pinty, J.-P., and Shipway, B.: Evaluation of Cloud-Resolving Model Intercomparison Simulations Using TWP-ICE Observations: Precipitation and Cloud Structure, *Journal of Geophysical Research: Atmospheres*, 116, D12 206, <https://doi.org/10.1029/2010JD015180>, 2011.
- 5 Varble, A., Zipser, E. J., Fridlind, A. M., Zhu, P., Ackerman, A. S., Chaboureaud, J.-P., Collis, S., Fan, J., Hill, A., and Shipway, B.: Evaluation of Cloud-Resolving and Limited Area Model Intercomparison Simulations Using TWP-ICE Observations: 1. Deep Convective Updraft Properties, *Journal of Geophysical Research: Atmospheres*, 119, 13,891–13,918, <https://doi.org/10.1002/2013JD021371>, 2014a.
- Varble, A., Zipser, E. J., Fridlind, A. M., Zhu, P., Ackerman, A. S., Chaboureaud, J.-P., Fan, J., Hill, A., Shipway, B., and Williams, C.:
10 Evaluation of Cloud-Resolving and Limited Area Model Intercomparison Simulations Using TWP-ICE Observations: 2. Precipitation Microphysics, *Journal of Geophysical Research: Atmospheres*, 119, 13,919–13,945, <https://doi.org/10.1002/2013JD021372>, 2014b.
- Wang, Z., Guo, L., Wang, S., Chen, L., and Wang, H.: Review of Random Walk in Image Processing, *Archives of Computational Methods in Engineering*, 26, 17–34, <https://doi.org/10.1007/s11831-017-9225-4>, 2019.
- Watson-Parris, D., Schutgens, N., Cook, N., Kipling, Z., Kershaw, P., Gryspeerdt, E., Lawrence, B., and Stier, P.: Community Intercomparison Suite (CIS) v1.4.0: A Tool for Intercomparing Models and Observations, *Geosci. Model Dev.*, 9, 3093–3110, <https://doi.org/10.5194/gmd-9-3093-2016>, 2016.
- 830 Wilcox, E. M.: Spatial and Temporal Scales of Precipitating Tropical Cloud Systems in Satellite Imagery and the NCAR CCM3, *Journal of Climate*, 16, 3545–3559, [https://doi.org/10.1175/1520-0442\(2003\)016<3545:SATSOP>2.0.CO;2](https://doi.org/10.1175/1520-0442(2003)016<3545:SATSOP>2.0.CO;2), 2003.
- Wilcox, E. M. and Ramanathan, V.: Scale Dependence of the Thermodynamic Forcing of Tropical Monsoon Clouds: Results from TRMM
835 Observations, *Journal of Climate*, 14, 1511–1524, [https://doi.org/10.1175/1520-0442\(2001\)014<1511:SDOTTF>2.0.CO;2](https://doi.org/10.1175/1520-0442(2001)014<1511:SDOTTF>2.0.CO;2), 2001.

- Wood, R. and Field, P. R.: The Distribution of Cloud Horizontal Sizes, *Journal of Climate*, 24, 4800–4816, <https://doi.org/10.1175/2011JCLI4056.1>, 2011.
- Zhao, M. and Austin, P. H.: Life Cycle of Numerically Simulated Shallow Cumulus Clouds. Part II: Mixing Dynamics, *Journal of the Atmospheric Sciences*, 62, 1291–1310, <https://doi.org/10.1175/JAS3415.1>, 2005a.
- 840 Zhao, M. and Austin, P. H.: Life Cycle of Numerically Simulated Shallow Cumulus Clouds. Part I: Transport, *Journal of the Atmospheric Sciences*, 62, 1269–1290, <https://doi.org/10.1175/JAS3414.1>, 2005b.
- Zheng, C., Pulido, J., Thorman, P., and Hamann, B.: An Improved Method for Object Detection in Astronomical Images, *Monthly Notices of the Royal Astronomical Society*, 451, 4445–4459, <https://doi.org/10.1093/mnras/stv1237>, 2015.
- Zinner, T., Mannstein, H., and Tafferner, A.: Cb-TRAM: Tracking and Monitoring Severe Convection from Onset over Rapid De-
- 845 velopment to Mature Phase Using Multi-Channel Meteosat-8 SEVIRI Data, *Meteorology and Atmospheric Physics*, 101, 191–210, <https://doi.org/10.1007/s00703-008-0290-y>, 2008.
- Zinner, T., Forster, C., de Coning, E., and Betz, H.-D.: Validation of the Meteosat Storm Detection and Nowcasting System Cb-TRAM with Lightning Network Data & Europe and South Africa, *Atmospheric Measurement Techniques*, 6, 1567–1583, <https://doi.org/https://doi.org/10.5194/amt-6-1567-2013>, 2013.