mheikenfeld-tobac Documentation

Release setup_v1.2

Sep 14, 2019

Contents

tobac is a Python package to identify, track and analyse clouds in different types of gridded datasets, such as 3D model output from cloud resolving model simulations or 2D data from satellite retrievals.

The software is set up in a modular way to include different algorithms for feature identification, tracking and analyses. In the current implementation, individual features are indentified as either maxima or minima in a two dimensional time varying field. The volume/are associated with the identified object can be determined based on a time-varying 2D or 3D field and a threshold value. In the tracking step, the identified objects are linked into consistent trajectories representing the cloud over its lifecycle. Analysis and visualisation methods provide a convenient way to use and display the tracking results.

Version 1.0 of tobac and some example applications are described in a paper that is currently in discussion for the journal “Geoscientific Model Development” as:

Heikenfeld, M., Marinescu, P. J., Christensen, M., Watson-Parris, D., Senf, F., van den Heever, S. C., and Stier, P.: tobac v1.0: towards a flexible framework for tracking and analysis of clouds in diverse datasets, Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-105 , in review, 2019.

The project is currently extended by several contributors to include additional workflows and algorithms using the same structure, synthax and data formats.

CHAPTER 1

Installation

tobac is now capable of working with both Python 2 and Python 3 (tested for 2.7,3.6 and 3.7) installations.

The easiest way is to install the most recent version of tobac via conda and the conda-forge channel:

` conda install -c conda-forge tobac `

This will take care of all necessary dependencies and should do the job for most users and also allows for an easy update of the installation by

` conda update -c conda-forge tobac `

You can also install conda via pip, which is mainly interesting for development purposed or to use specific development branches for the Github repository.

The follwoing python packages are required (including dependencies of these packages):

trackpy, *scipy*, *numpy*, *iris*, *scikit-learn*, *scikit-image*, *cartopy*, *pandas*, *pytables*

If you are using anaconda, the following command should make sure all dependencies are met and up to date:

```
conda install -c conda-forge -y trackpy scipy numpy iris scikit-learn
scikit-image cartopy pandas pytables
```

You can directly install the package directly from github with pip and either of the two following commands:

```
pip install --upgrade git+ssh://git@github.com/climate-processes/
tobac.git
```

```
pip install --upgrade git+https://github.com/climate-processes/tobac.
git
```

You can also clone the package with any of the two following commands:

```
git clone git@github.com:climate-processes/tobac.git
```

```
git clone https://github.com/climate-processes/tobac.git
```

and install the package from the locally cloned version (The trailing slash is actually necessary):

```
pip install --upgrade tobac/
```

CHAPTER 2

Data input and output

The output of the different analysis steps in tobac are output as either pandas DataFrames in the case of one-dimensional data, such a lists of identified features or cloud trajectoies or as Iris cubes in the case of 2D/3D/4D fields such as cloud masks.

Interoperability with xarray is provided by the convenient functions allowing for a transformation between the two data types. xarray DataArays can be easily converted into iris cubes using xarray's to__iris() method, while the Iris cubes produced as output of tobac can be turned into xarray DataArrays using the from__iris() method.

For the future development of the next major version of tobac, we are envisaging moving the basic data structures from Iris cubes to xarray DataArrays for improved computing performance and interoperability with other open-source sorftware packages.

CHAPTER 3

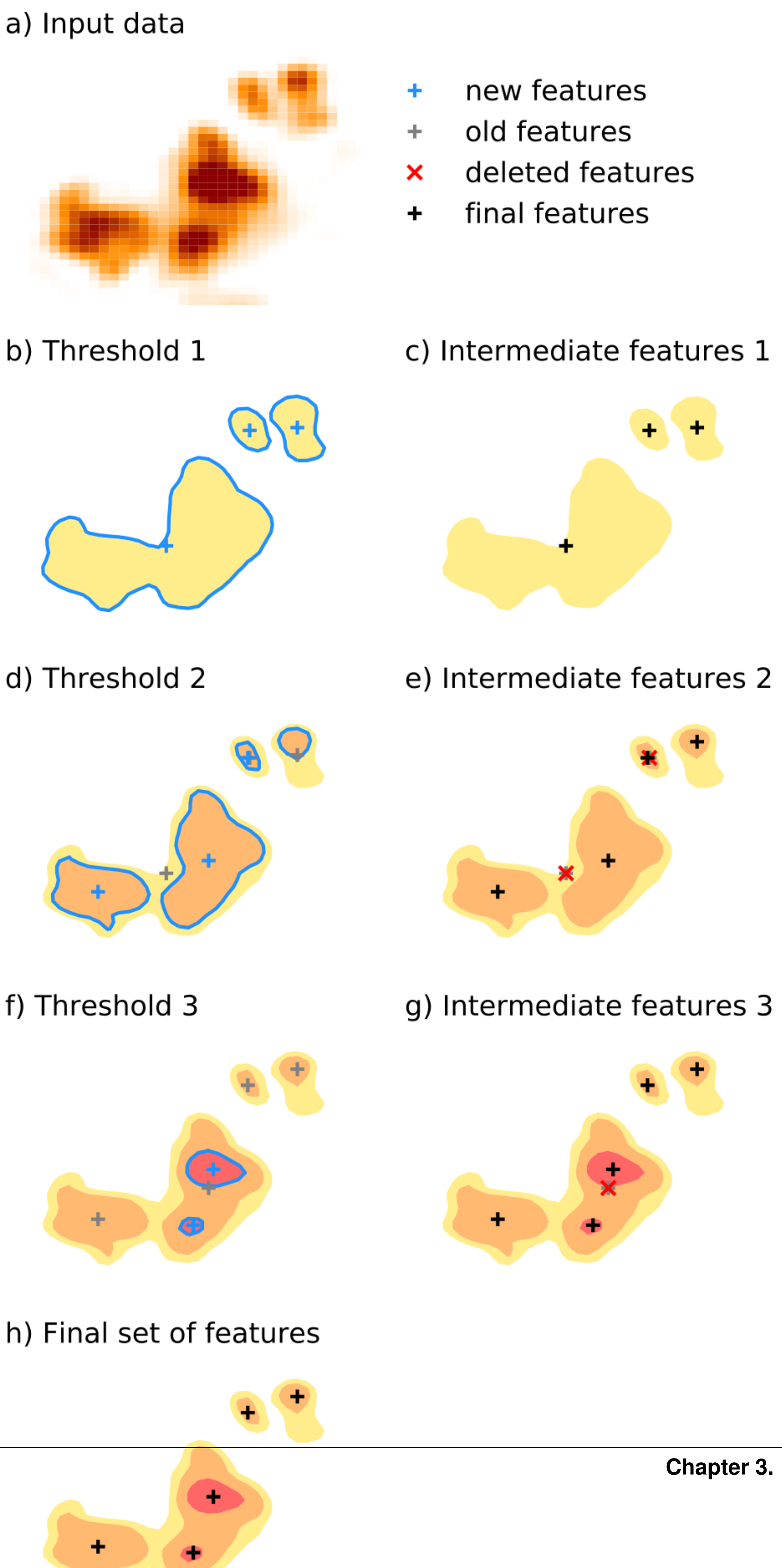
Feature detection

The feature detection form the first step of the analysis.

Currently implemented methods:

Multiple thresholds:

Features are identified as regions above or below a sequence of subsequent thresholds (if searching for eather maxima or minima in the data). Subsequently more restrictive threshold values are used to further refine the resulting features and allow for separation of features that are connected through a continuous region of less restrictive threshold values.


a) Input data
new features
old features
deleted features
final features
b) Threshold 1
c) Intermediate features 1
d) Threshold 2
e) Intermediate features 2
f) Threshold 3
g) Intermediate features 3
h) Final set of features

Current development: We are currently working on additional methods for the identification of cloud features in different types of datasets. Some of these methods are specific to the input data such a combination of different channels from specific satellite imagers. Some of these methods will combine the feature detection and segmentations step in one single algorithm.

CHAPTER 4

Segmentation

The segmentation step aims at associating cloud areas (2D data) or cloud volumes (3D data) with the identified and tracked features.

Currently implemented methods:

> **Watershedding in 2D:** Markers are set at the position of the individual feature positions identified in the detection step. Then watershedding with a fixed threshold is used to determine the area around each feature above/below that threshold value. This results in a mask with the feature id at all pixels identified as part of the clouds and zeros in all cloud free areas.
>
> **Watershedding in 3D:** Markers are set in the entire column above the individual feature positions identified in the detection step. Then watershedding with a fixed threshold is used to determine the volume around each feature above/below that threshold value. This results in a mask with the feature id at all voxels identified as part of the clouds and zeros in all cloud free areas.

Current development: We are currently working on providing additional approaches and algorithms for the segmentation step. Several of these approaches will combine the feature detection and segmentation into a single st pixe on

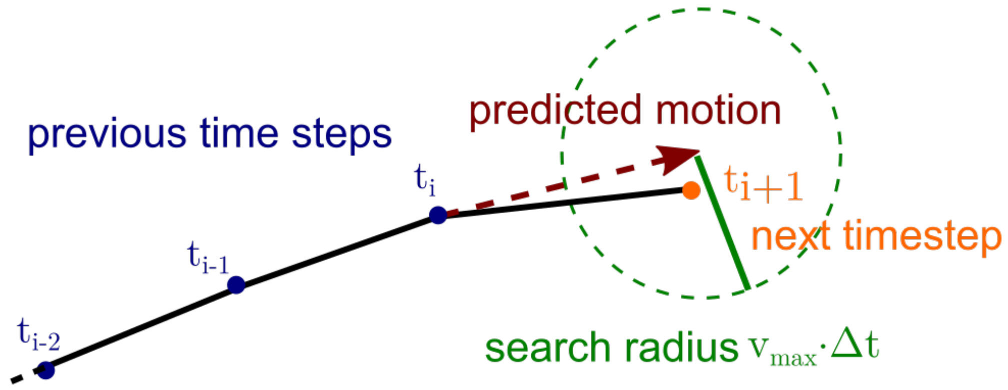
CHAPTER 5

Linking

Currently implemented methods for linking detected features into cloud tracks:

Trackpy:

This method uses the trackpy library (http://soft-matter.github.io/trackpy). This approach only takes the point-like position of the feature, e.g. determined as the weighted mean, into account and does not use any other information about the identified features into account. The linking makes use of the information from the linked features in the previous timesteps to predict the position and then searches for matching features in a search range determined by the v_max parameter.



Current development:

We are currently actively working on additional options for the tracking of the clouds that take into account the shape of the identified features by evaluating overlap between adjacent time steps, as well as the inclusion of cloud splitting and merging.

CHAPTER 6

Analysis

tobac provides several analysis functions that allow for the calculation of important quantities based on the tracking results. This includes the calculation of important properties of the tracked objects such as cloud lifetimes, cloud areas/volumes, but also allows for a convenient calculation of statistics for arbitratry fields of the same shape as as the input data used for the tracking analysis.

CHAPTER 7

Plotting

tobac provides functions to conveniently visualise the tracking results and analyses.

CHAPTER 8

Example notebooks

tobac is provided with a set of Jupyter notebooks that show examples of the application of tobac for different types of datasets.

The notebooks can be found in the **examples** folder in the the repository. The necessary input data for these examples is avaliable on zenodo: www.zenodo.org/. . . and can be downloaded automatically by the Jupyter notebooks.

The examples currently include four different applications of tobac: 1. Tracking of scattered convection based on vertical velocity and condensate mixing ratio for 3D cloud-resolving model output. 2. Tracking of scattered convection based on surface precipitation from the same cloud-resolving model output 3. Tracking of convective clouds based on outgoing longwave radiation (OLR) for convection-permitting model simulation output 4. Tracking of convective clouds based on OLR in geostationary satellite retrievals.

The examples are based on the analyses presented in an article describing tobac that has been submitted to the journal GMD (Geophysical model development).