
PyCupid Documentation

Release 2

LIRAE Group, UTFSM

Oct 22, 2020

Contents:

| | | |
|----------|--|-----------|
| 1 | Contents | 3 |
| 1.1 | PyCupid Core Wrappers | 3 |
| 1.2 | Usage of PyCupid Interfaces of Cupid's Clumping Algorithms | 3 |
| 1.3 | Algorithms Parameters | 8 |
| 2 | Installation | 15 |
| 3 | Indices and tables | 17 |

Welcome to **PyCupid** project: Python wrappers for Starlink's CUPID package. CUPID is part of the Starlink Software Collection <<http://www.starlink.ac.uk/>>, it provides a set of commands that identify and analyse clumps of emission within 2- and 3-dimensional data arrays.

This module aims to be a clean interface between CUPID implementation (C and Fortran) and Python. As such, users can simply import the module and use CUPID functions as if they were implemented in Python.

For examples look into our [Ipython Notebook Repository](#).

1.1 PyCupid Core Wrappers

Note: For Astronomers and Advanced Users

Python Wrappers Functions to use the mean **clumping algorithms** in CUPID.

1.1.1 Clumping Functions

1.2 Usage of PyCupid Interfaces of Cupid's Clumping Algorithms

```
from pycupid import clumpfind, fellwalker, reinhold, gaussclumps
from astropy.io import fits
import numpy as np
import numpy.ma as ma
import matplotlib.pyplot as plt

def estimate_rms(data):
    """
    Computes RMS value of an N-dimensional numpy array
    """

    if isinstance(data, ma.MaskedArray):
        ret = np.sum(data*data) / (np.size(data) - np.sum(data.mask))
    else:
        ret = np.sum(data*data) / np.size(data)
    return np.sqrt(ret)

def load_data(fits_path):
    hdulist = fits.open(fits_path)
```

(continues on next page)

(continued from previous page)

```
data = hdulist[0].data
# dropping out the stokes dimension
data = np.ascontiguousarray(data[0])

# in case NaN values exist on cube
mask = np.isnan(data)
if np.any(mask): data = ma.masked_array(data, mask=mask)

# map to 0-1 intensity range
data -= data.min()
data /= data.max()

if data.shape[0]==1:
    data = np.ascontiguousarray(data[0])
    if np.any(mask):
        mask = np.ascontiguousarray(mask[0])
        data = ma.masked_array(data, mask=mask)
return data
```

1.2.1 Loading FITS data

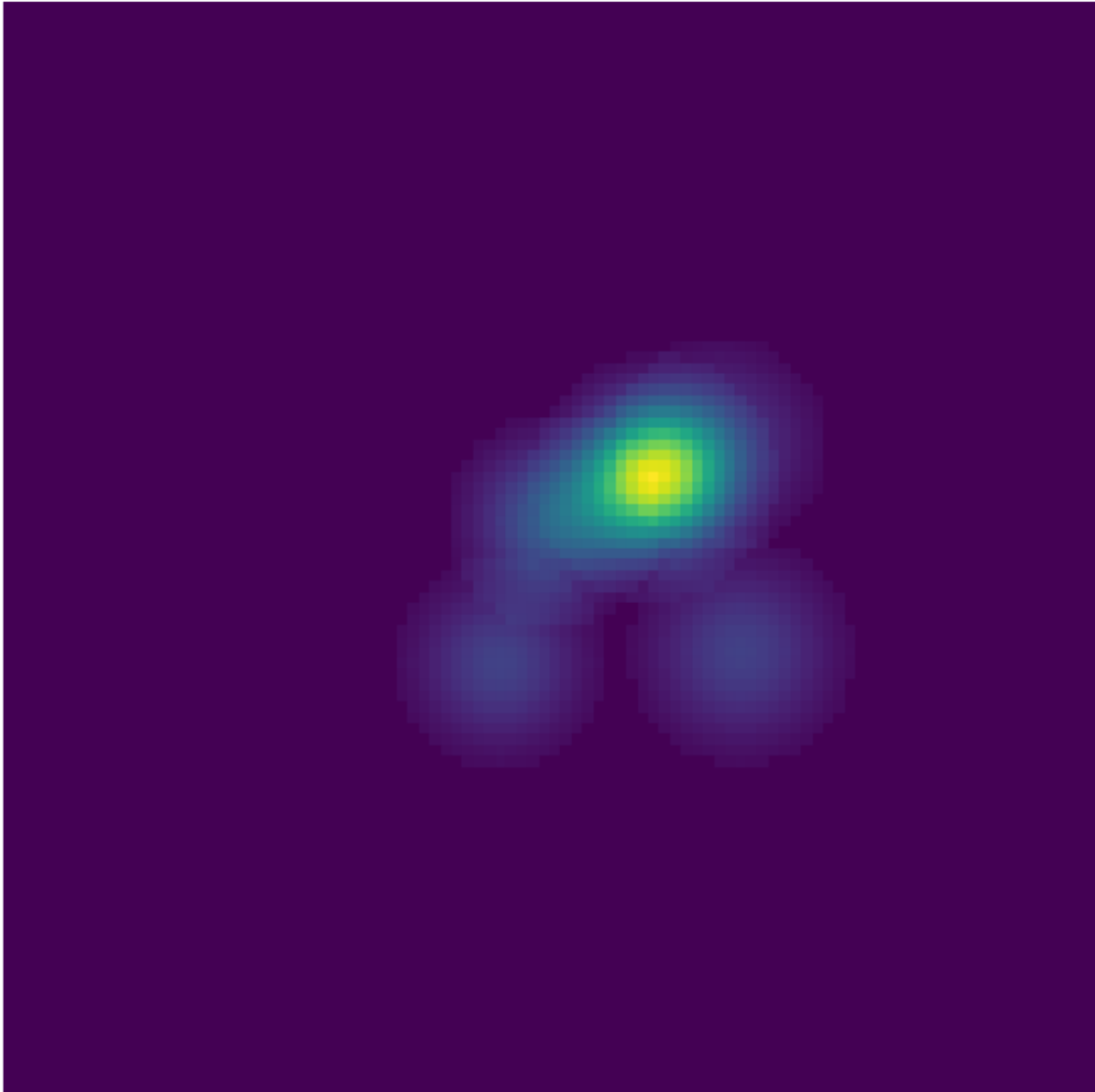
```
data = load_data('Orion.cont.image.fits')
rms = estimate_rms(data)
```

1.2.2 GaussClump

```
out_gc = gaussclumps(data, 0.5*rms)
```

```
out_gc[out_gc==out_gc.min()] = 0.
```

```
plt.figure(figsize=(10,10))
plt.imshow(out_gc)
plt.axis('off')
plt.show()
```

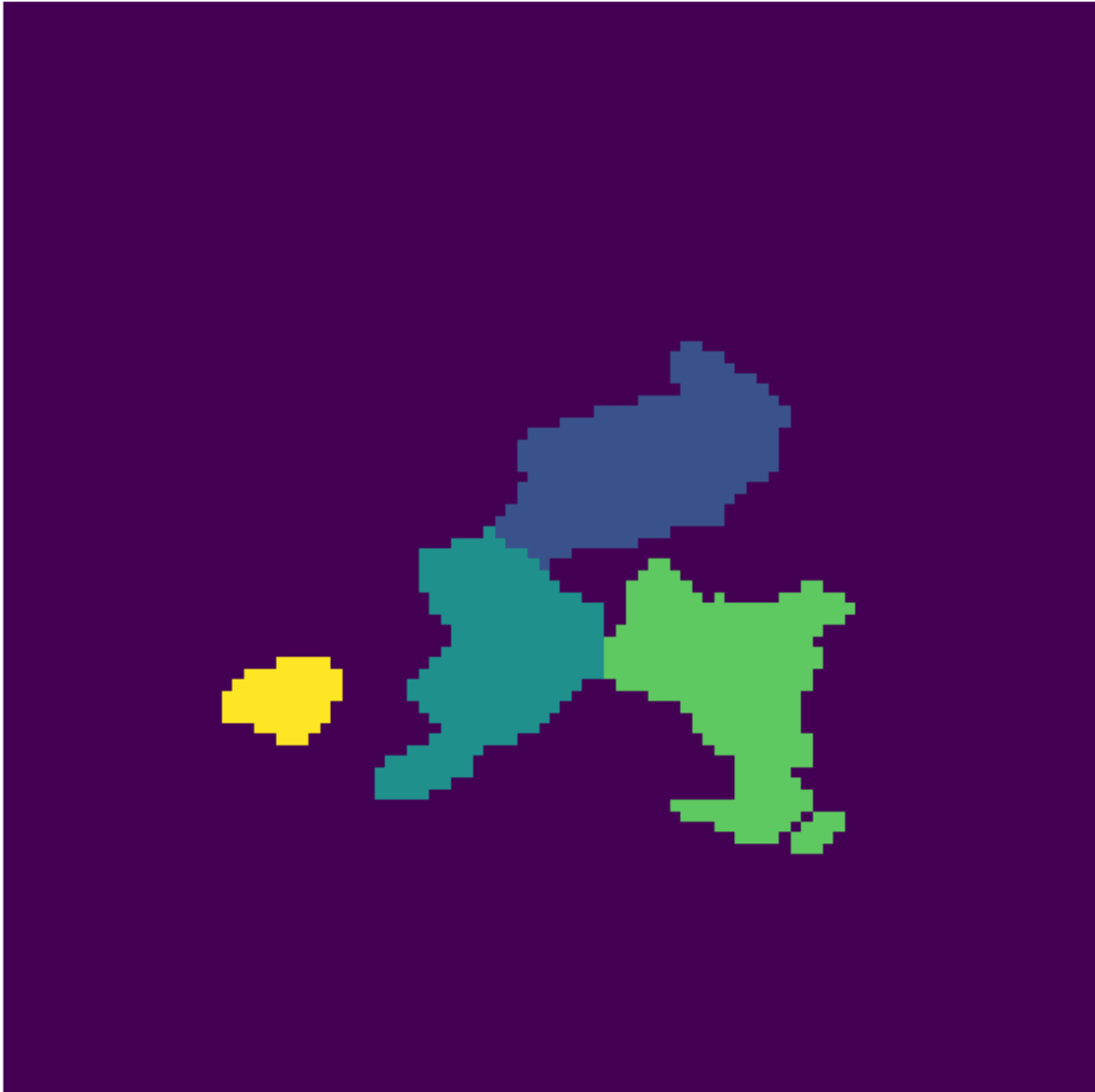



1.2.3 ClumpFind

```
out_cf = clumpfind(data, 0.5*rms)
```

```
out_cf[out_cf==out_cf.min()] = 0.
```

```
plt.figure(figsize=(10,10))  
plt.imshow(out_cf)  
plt.axis('off')  
plt.show()
```

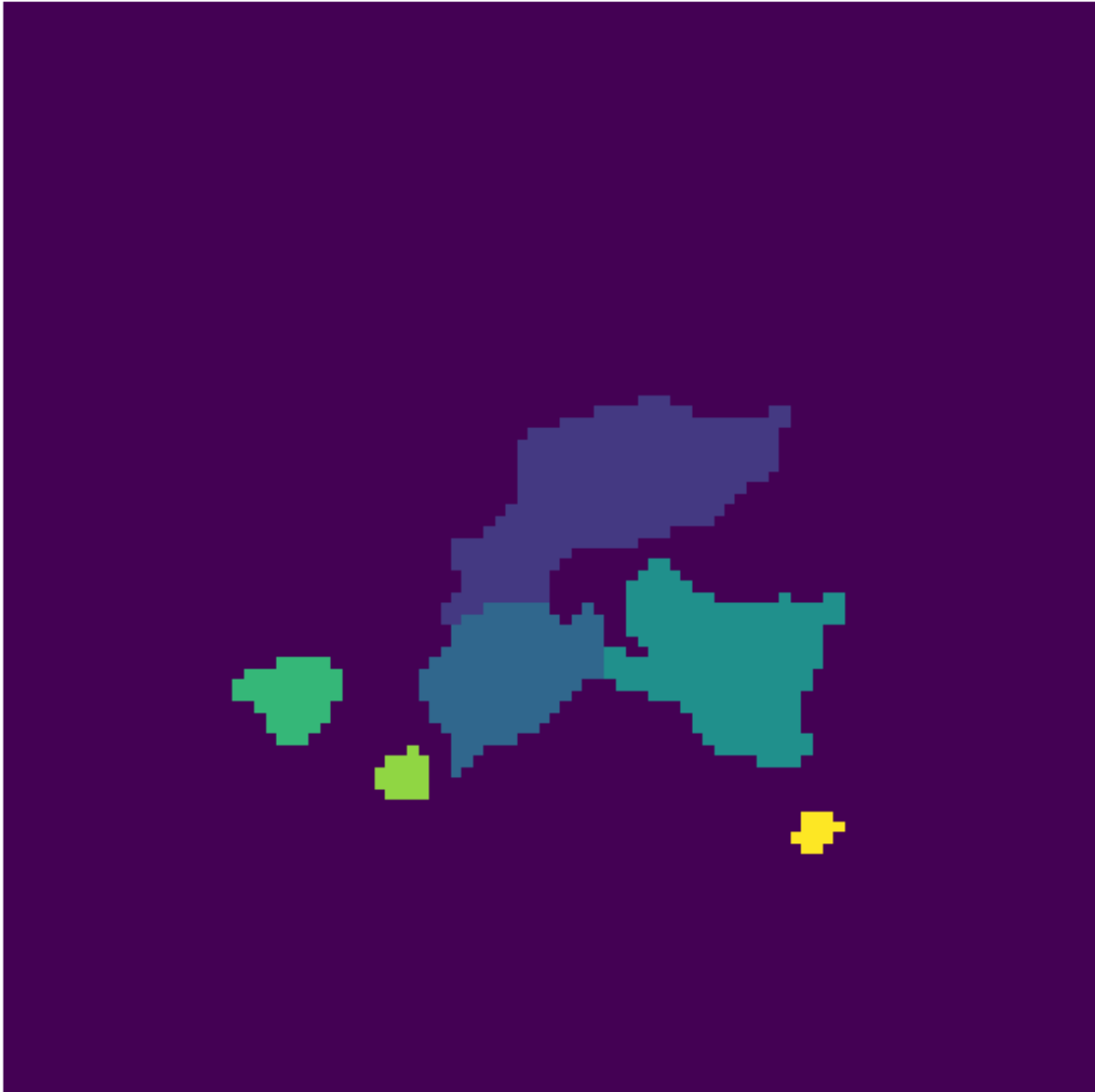


1.2.4 FellWalker

```
out_fw = fellwalker(data, 0.5*rms)
```

```
out_fw[out_fw==out_fw.min()] = 0.
```

```
plt.figure(figsize=(10,10))
plt.imshow(out_fw)
plt.axis('off')
plt.show()
```

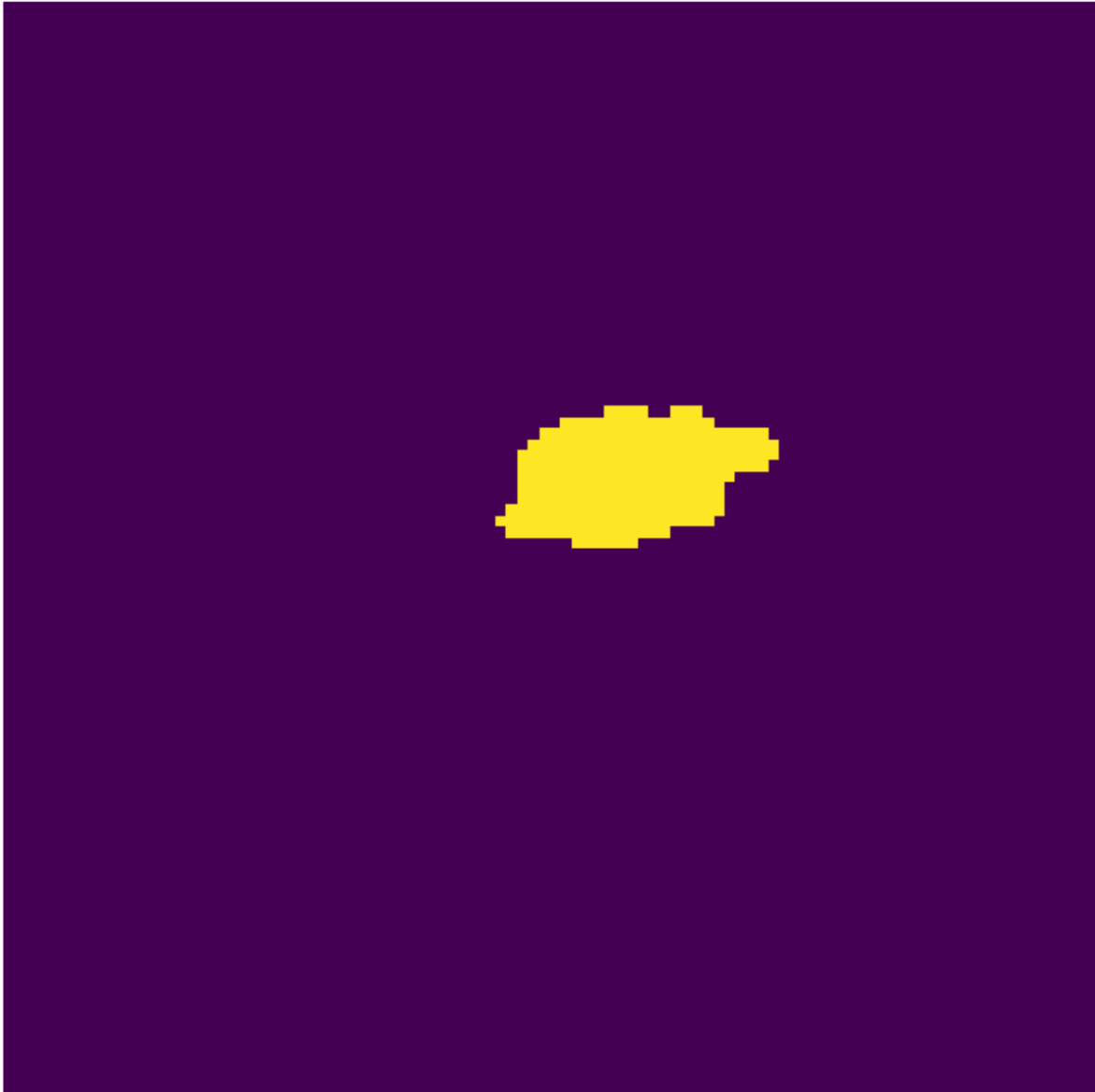


1.2.5 Reinhold

```
out_rh = reinhold(data, 0.5*rms)
```

```
out_rh[out_rh==out_rh.min()] = 0.
```

```
plt.figure(figsize=(10,10))  
plt.imshow(out_rh)  
plt.axis('off')  
plt.show()
```



1.3 Algorithms Parameters

In this section you will find a description of the allowed **keys names** and **values** in the `params` dictionary for each algorithm. The description of the parameters is the same as in the [official CUPID documentation](#).

Note: For each algorithm we list the parameter names (string) and its allowed value:
`params['PARAMETER_NAME'] = VALUE.`

1.3.1 GaussClump

1. "EXTRACOLS": (int) If set to a positive integer, then extra method-specific columns are added to the output catalogue. If set to 1, then the catalogue will include columns with names GCFWHM<i> (where <i> is 1, 2, or 3), holding the FWHM of the fitted Gaussian in units of pixels (these FWHM values have NOT been reduced to excluded the effect of the beam width), and "GCANGLE" - the spatial orientation angle (in degrees, positive from +ve GRID1 axis to +ve GRID2 axis). If set greater than 1, then additional columns will be included holding the initial estimates for the peak and background values, the number of fitting iterations used and the final ch-squared value for the fit.
2. "FWHMBEAM": (float) The FWHM of the instrument beam, in pixels. The fitted Gaussians are not allowed to be smaller than the instrument beam. This prevents noise spikes being fitted. In addition, if application parameter DECONV is set TRUE, the clump widths written to the output catalogue are reduced (in quadrature) by this amount. (default=2.0)
3. "FWHMSTART": (float) An initial guess at the ratio of the typical observed clump size to the instrument beam width. This is used to determine the starting point for the algorithm which finds the best fitting Gaussian for each clump. If no value is supplied (or if FwhmBeam is zero), the initial guess at the clump size is based on the local profile around the pixel with peak value.
4. "MAXBAD": (float) The maximum fraction of bad pixels which may be included in a clump. Clumps will be excluded if they contain more bad pixels than this value. (default=0.05)
5. "MAXCLUMPS": (int) Specifies a termination criterion for the GaussClumps algorithm. The algorithm will terminate when "MaxClumps" clumps have been identified, or when one of the other termination criteria is met. (default=float('inf'))
6. "MAXNF": (int) The maximum number of evaluations of the objective function allowed when fitting an individual clump. Here, the objective function evaluates the chi-squared between the current gaussian model and the data being fitted. (default=100)
7. "MAXSKIP": (int) The maximum number of consecutive failures which are allowed when fitting Gaussians. If more than "MaxSkip" consecutive clumps cannot be fitted, the iterative fitting process is terminated. (default=10)
8. "MODELLIM": (float) Determines the value at which each Gaussian model is truncated to zero. Model values below ModelLim times the RMS noise are treated as zero. (default=0.5)
9. "NPAD": (int) Specifies a termination criterion for the GaussClumps algorithm. The algorithm will terminate when "Npad" consecutive clumps have been fitted all of which have peak values less than the threshold value specified by the "Thresh" parameter, or when one of the other termination criteria is met. (default=10)
10. "RMS": (float) The global RMS noise level in the data. The default value is the value supplied for parameter RMS.
11. "S0": (float) The Chi-square stiffness parameter "S0" which encourages the fitted gaussian value to be below the corresponding value in the observed data at every point (see the Stutski & Gusten paper). (default=1.0)
12. "SA": (float) The Chi-square stiffness parameter "Sa" which encourages the peak amplitude of each fitted gaussian to be close to the corresponding maximum value in the observed data (see the Stutski & Gusten paper). (default=1.0)
13. "SB": (float) An additional Chi-square stiffness parameter which encourages the background value to stay close to its initial value. This stiffness is not present in the Stutzki & Gusten paper but is added because the background value is usually determined by data points which have very low weight and is thus poorly constrained. It would thus be possibly to get erroneous background values without this extra stiffness. (default=0.1)
14. "SC": (float) The Chi-square stiffness parameter "Sc" which encourages the peak position of each fitted gaussian to be close to the corresponding peak position in the observed data (see the Stutski & Gusten paper).

(default=1.0)

15. "THRESH": (float) Gives the minimum peak amplitude of clumps to be fitted by the GaussClumps algorithm (see also GaussClumps.NPad). The supplied value is multiplied by the RMS noise level before being used. (default=2.0)
16. "VELORES": (float) The velocity resolution of the instrument, in channels. The velocity FWHM of each clump is not allowed to be smaller than this value. Only used for 3D data. In addition, the velocity width of each clump written to the output catalogue is reduced (in quadrature) by this amount. (default=2.0)
17. "VELOSTART": (float) An initial guess at the ratio of the typical observed clump velocity width to the velocity resolution. This is used to determine the starting point for the algorithm which finds the best fitting Gaussian for each clump. If no value is supplied (or if VeloRes is zero), the initial guess at the clump velocity width is based on the local profile around the pixel with peak value. Only used for 3D data.
18. "WMIN": (float) This parameter, together with GaussClumps.Wwidth, determines which input data values are used when fitting a Gaussian to a given peak in the data array. It specifies the minimum normalised weight which is to be used. Pixels with normalised weight smaller than this value are not included in the fitting process. The absolute weight values are normalised by dividing them by a value equal to the mean of the absolute weights plus four standard deviations. An iterative sigma clipping algorithm is used when calculating this value in order to eliminate the effects of any pixel that have unusually low variance values, and thus unusually high absolute weights. (default=0.05)
19. "WWIDTH": (float) This parameter, together with GaussClumps.Wmin, determines which input data values are used when fitting a Gaussian to a given peak in the data array. It is the ratio of the width of the Gaussian weighting function (used to weight the data around each clump during the fitting process), to the width of the initial guess Gaussian used as the starting point for the Gaussian fitting process. The Gaussian weighting function has the same centre as the initial guess Gaussian. (default=2.0)

1.3.2 ClumpFind

1. "ALLOWEDGE": (int) If set to a zero value, then clumps are rejected if they touch any edge of the data array. If non-zero, then such clumps are retained. Note, other implementations of ClumpFind often include such clumps but flag them in some way. (default=0)
2. "DELTAT": (float) The gap between the contour levels. Only accessed if no value is supplied for "Level1", in which case the contour levels are linearly spaced, starting at a lowest level given by "Tlow" and spaced by "DeltaT". Note, small values of DeltaT can result in noise spikes being interpreted as real peaks, whilst large values can result in some real peaks being missed and merged in with neighbouring peaks. The default value of two times the RMS noise level is usually considered to be optimal, although this obviously depends on the RMS noise level being correct. The value can be supplied either as an absolute data value, or as a multiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=2*RMS)
3. "FWHMBEAM": (float) The FWHM of the instrument beam, in pixels. If application parameter DECONV is set TRUE, the clump widths written to the output catalogue are reduced (in quadrature) by this amount. If a direct comparison with other implementations of the ClumpFind algorithm is required, DECONV should be set to FALSE. (default=2.0)
4. "IDLALG": (int) If a non-zero value is supplied, then FINDCLUMPS emulates the ClumpFind algorithm as implemented by the IDL package available from Jonathan Williams WWW site on 28th April 2006. The default value of zero causes FINDCLUMPS to use the algorithm described in the Williams et al ApJ paper of 1994. These two algorithms differ in the way that pixels within merged clumps are allocated to individual clumps. Also the ApJ algorithm rejects clumps that do not extend above the second contour level, whereas the IDL algorithm accepts such clumps. See also parameter BACKOFF. (default=0)
5. "LEVEL": (float) The n-th data value at which to contour the data array (where <n> is an integer). Values should be given for "Level1", "Level2", "Level3", etc. Any number of contours can be supplied, but there must be no gaps in the progression of values for <n>. The values will be sorted into descending order before

being used. If "Levell" is not supplied (the default), then contour levels are instead determined automatically using parameters "Tlow" and "DeltaT". Note clumps found at higher contour levels are traced down to the lowest supplied contour level, but any new clumps which are initially found at the lowest contour level are ignored. That is, clumps must have peaks which exceed the second lowest contour level to be included in the returned catalogue. The values can be supplied either as absolute data values, or as multiples of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS").

6. "MAXBAD": (float) The maximum fraction of pixels in a clump that are allowed to be adjacent to a bad pixel. If the fraction of clump pixels adjacent to a bad pixel exceeds this value, the clump is excluded. If a direct comparison with other implementations of the ClumpFind algorithm is required, a value of 1.0 should be used. (default=0.05)
7. "MINPIX": (int) The lowest number of pixel which a clump can contain. If a candidate clump has fewer than this number of pixels, it will be ignored. This prevents noise spikes from being interpreted as real clumps. The default value is based on the supplied values for the other parameters that specify the minimum peak height, the background level and the instrumental beam widths, limited to be at least 16 pixels (for 3D data), 7 pixels (for 2D data) or 3 pixels (for 1D data, or if "PERSPECTRUM" is set TRUE). If a direct comparison with other implementations of the ClumpFind algorithm is required, a value of 5 should be used (for 3D data) or 20 (for 2D data).
8. "NAXIS": (int) Controls the way in which contiguous areas of pixels are located when contouring the data. When a pixel is found to be at or above a contour level, the adjacent pixels are also checked. "Naxis" defines what is meant by an "adjacent" pixel in this sense. The supplied value must be at least 1 and must not exceed the number of pixel axes in the data. The default value equals the number of pixel axes in the data. If the data is 3-dimensional, any given pixel can be considered to be at the centre of a cube of neighbouring pixels. If "Naxis" is 1 only those pixels which are at the centres of the cube faces are considered to be adjacent to the central pixel. If "Naxis" is 2, pixels which are at the centre of any edge of the cube are also considered to be adjacent to the central pixel. If "Naxis" is 3, pixels which are at the corners of the cube are also considered to be adjacent to the central pixel. If the data is 2-dimensional, any given pixel can be considered to be at the centre of a square of neighbouring pixels. If "Naxis" is 1 only those pixels which are at the centres of the square edges are considered to be adjacent to the central pixel. If "Naxis" is 2, pixels which are at square corners are also considered to be adjacent to the central pixel. For one dimensional data, a value of 1 is always used for "Naxis", and each pixel simply has 2 adjacent pixels, one on either side. Note, the supplied "naxis" value is ignored if the ADAM parameter "PERSPECTRUM" is set TRUE.
9. "RMS": (float) The global RMS noise level in the data. The default value is the value supplied for parameter RMS.
10. "TLOW": (float) The lowest level at which to contour the data array. Only accessed if no value is supplied for "Levell". See also "DeltaT". The value can be supplied either as an absolute data value, or as a multiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=2*RMS)
11. "VELORES": (float) The velocity resolution of the instrument, in channels. The velocity width of each clump written to the output catalogue is reduced (in quadrature) by this amount. If a direct comparison with other implementations of the ClumpFind algorithm is required, a value of zero should be used. (default=2.0)

1.3.3 FellWalker

1. "ALLOWEDGE": (int) If set to a zero value, then clumps are rejected if they touch any edge of the data array. If non-zero, then such clumps are retained. (default=1)
2. "CLEANITER": (int) This gives the number of times to apply the cellular automata which cleans up the filled clumps. This cellular automata replaces each clump index by the most commonly occurring value within a 3x3x3 cube (or 2x2 square for 2D data) of neighbours. The supplied value is ignored and a value of zero is assumed if "PERSPECTRUM" is set TRUE. (default=1)

3. "FLATSLOPE": (float) Any initial section to a walk which has an average gradient (measured over 4 steps) less than this value, and for which the data value is less than "NOISE+2RMS", will not be included in the clump. The value of this parameter is the data increment between pixels, and can be supplied either as an absolute data value, or as a mutiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=1.0*RMS)
4. "FWHMBEAM": (float) The FWHM of the instrument beam, in pixels. If application paremeter DECONV is set TRUE, the clump widths written to the output catalogue are reduced (in quadrature) by this amount. (default=2.0)
5. "MAXBAD": (float) The maximum fraction of pixels in a clump that are allowed to be adjacent to a bad pixel. If the fraction of clump pixels adjacent to a bad pixel exceeds this value, the clump is excluded. (default=0.05)
6. "MINDIP": (float) If the dip between two adjacent peaks is less than this value, then the peaks are considered to be part of the same clump. The value can be supplied either as an absolute data value, or as a mutiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=2.0*RMS)
7. "MINHEIGHT": (float) If the peak value in a clump is less than this value then the clump is not included in the returned list of clumps. The value can be supplied either as an absolute data value, or as a mutiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=Noise)
8. "MINPIX": (int) he lowest number of pixel which a clump can contain. If a candidate clump has fewer than this number of pixels, it will be ignored. This prevents noise spikes from being interpreted as real clumps. The default value is based on the supplied values for the other parameters that specify the minimum peak height, the background level and the instrumental beam widths, limited to be at least 16 pixels (for 3D data), 7 pixels (for 2D data) or 3 pixels (for 1D data).
9. "MAXJUMP": (int) Defines the extent of the neighbourhood about a local maximum which is checked for higher pixel values. The neighbourhood checked is square or cube with side equal to twice the supplied value, in pixels. (default=4)
10. "NOISE": (float) Defines the data value below which pixels are considered to be in the noise. No walk will start from a pixel with data value less than this value. The value can be supplied either as an absolute data value, or as a mutiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=2*RMS)
11. "RMS": (float) The global RMS noise level in the data. The default value is the value supplied for parameter RMS.
12. "VELORES": (float) The velocity resolution of the instrument, in channels. The velocity width of each clump written to the output catalogue is reduced (in quadrature) by this amount. (default=2.0)

1.3.4 Reinhold

1. "FWHMBEAM": (float) The FWHM of the instrument beam, in pixels. If application paremeter DECONV is set TRUE, the clump widths written to the output catalogue are reduced (in quadrature) by this amount. (default=2.0)
2. "MAXBAD": (float) The maximum fraction of pixels in a clump that are allowed to be adjacent to a bad pixel. If the fraction of clump pixels adjacent to a bad pixel exceeds this value, the clump is excluded. (default=0.05)
3. "MINLEN": (int) The minimum number of pixels spanned by a peak along any one dimension in order for the peak to be considered significant. If a peak is spanned by fewer than this number of pixels on any axis, then it is ignored. (default=4)

4. "MINPIX": (int) The lowest number of pixel which a clump can contain. If a candidate clump has fewer than this number of pixels, it will be ignored. This prevents noise spikes from being interpreted as real clumps. The default value is based on the supplied values for the other parameters that specify the minimum peak height, the background level and the instrumental beam widths, limited to be at least 16 pixels(for 3D data), 7 pixels (for 2D data) or 3 pixels (for 1D data).
5. "NOISE": (float) Defines the data value below which pixels are considered to be in the noise. A peak is considered to end when the peak value dips below the "noise" value. The value can be supplied either as an absolute data value, or as a mutiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=2*RMS)
6. "THRESH": (float) The smallest significant peak height. Peaks which have a maximum data value less than this value are ignored. The value can be supplied either as an absolute data value, or as a mutiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=Noise+2*RMS)
7. "FLATSLOPE": (float) A peak is considered to end when the slope of a profile through the peak drops below this value. The value should be given as a change in data value between adjacent pixels. The value can be supplied either as an absolute data value, or as a mutiple of the RMS noise using the syntax "[x]RMS", where "[x]" is a numerical value (e.g. "3.2RMS"). (default=1.0*RMS)
8. "CATHRESH": (int) Controls the operation of the cellular automata which is used to erode the (previously dilated) edges regions prior to filling them with clump indices. If the number of edge pixels in the 3x3x3 pixel cube (or 2x2 pixel square for 2D data) surrounding any pixel is greater than CAThresh, then the central pixel is considered to be an edge pixel. Otherwise it is not considered to be an edge pixel. The default value is one less than the total number of pixels in the square or cube (i.e. 8 for 2D data and 26 for 3D data).
9. "CAITERATION": (int) This gives the number of times to apply the cellular automata which is used to erode the edges regions prior to filling them with clump indices. (default=1)
10. "FIXCLUMPSITERATIONS": (int) This gives the number of times to apply the cellular automata which cleans up the filled clumps. This cellular automata replaces each output pixel by the most commonly occuring value within a 3x3x3 cube (or 2x2 square for 2D data) of input pixels centred on the output pixel. (default=1)
11. "RMS": (float) The global RMS noise level in the data. The default value is the value supplied for parameter RMS.
12. "VELORES": (float) The velocity resolution of the instrument, in channels. The velocity width of each clump written to the output catalogue is reduced (in quadrature) by this amount. (default=2.0)

CHAPTER 2

Installation

To use PyCupid Wrappers you can install it directly from **PyPI** (recommended):

```
pip install pycupid
```

or clone the [project repository](#) and run:

```
python setup.py build install
```

which is not recommended since it requires to clone the whole Starlink project and compile CUPID.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`