

This repository uses a modified old version of the file `papaya2.hpp` from <https://github.com/morphometry/papaya2>

Banana is to be used for a variety of MT/MF related functions. By default, it creates Minkowski maps of the (absolute) value of  $\psi_s$  for a round window with diameter given by `smooth` for the brightness threshold given by `maxt`. Alternatively (`3d true`), it takes `numt` logarithmically spaced values from `mint` to `maxt` and saves a fits file with `numt` layers. Can also calculate the complex average (`avg true`, this is very useful for real data) or the average of the absolute value (additionally `abs_avg true`). Phases can be saved instead with `arg true`. Commands are discussed in detail in the list further below, here is a short overview.

`makeBanana true` leads to the calculation of various MF/MT for the areas of the image defined in the regions in `mask` (syntax see below) and the thresholds given by `mint`, `maxt` and `numt`. For every object in the `maskfile` a separate output table is created.

`makePeach true` is similar to `makeBanana`, but varies the window size instead of the brightness threshold at the given mask regions. Saves one table for every object.

`makeHist true` expects a file containing names of files containing point coordinates given by `stars` command (syntax see below) and creates a histogram of the input file brightness at the position of the points. Bins are linearly spaced with parameters `mint`, `maxt` and `numt`. File identifier given by `histapp` appended to each file.

`makePointspread true` takes a list of files containing points in the same format as for `makeHist` and creates an image with a gaussian of standard deviation given by `smooth` at the location of each star.

`makePNGs true` creates png files with a logarithmic brightness scale of the regions in `mask` with red given by `infile`, green by `greenfile` and blue by `bluefile`. `monochrome true` leads to creation of monochrome `infile`-based pngs. To remove single bright spots image can be eroded/dilated with diameter `erd` before.

`makeHedgehog true` creates linedensity maps at given `smooth` and calls `makeBubbles true`. Uses parameters `lineThresh` and `lineScale` for threshold of bubble detection and scale factor for length of line. `smooth` value of Minkmap to which image is compared is hardcoded for several relevant `smooth` values and set to `3*smooth` by default. Details see paper.

`makeBubbles true` searches for bubbles for existing linedensity map using `lineScale` and `lineThresh`. Also see `makeHedgehog`.

`makePattern [linedensity/minkmap/infile/infile.log]` creates a point pattern with intensity given by image according to command (`infile.log` means using the logarithm of the file in `infile`) and saves it as a table (can be read by R). Only takes image values larger than `lineThresh` into account (by default zero). `infile` can be smoothed with diameter given by `smooth` and eroded/dilated with diameter `erd`.

#### Recommendations:

Set a suitable file identifier (e.g. `wavelength`) with `filePrefix` and the correct folders for text files, minkmaps, line densities, pointspread, and PNGs (`out*` commands), objects to be read/written (`objectDIR`).

Many of these functions require calculating the shift in pixel coordinates introduced by smoothing relative to the original file to convert back and forth between corresponding locations. This is done by saving the header variable `CRPIX1` of the original file and comparing it to that of the smoothed file. While it is hard-coded for the example commands `file:halpha` and similar, it can be manually entered with the command `CRPIX1`. Especially important if `infile` is not an unshifted file.

Every command except the file:[name] commands demands either a number or true/false (false = anything except the word "true") as arguments. Give arguments separated by space after command

Syntax:

command Name of corresponding variable in source code, description

in infilename, FITS file to be read

mask maskfilename, File with names of objectfiles to be included in the original-Banana or makePNGs or makePeach call (everything else masked per object)

boxesToExclude boxesToExcludeName, text file containing boxes (format see below) to be set to zero in infile

filePrefix wavelength, prefix for file to be used in further filenames, usually wavelength

stars histfile, list with one ds9 region file located in ./objects/ per line containing pixel coordinates of stars in original image for histogram/other statistics (without .reg-ending, added automatically)

histapp histappendix, name appendix for histogram, enter anything besides histfile

mint min\_thresh, minimal threshold

maxt max\_thresh, maximal threshold or threshold for single threshold maps

numt num\_thresh number of (logarithmically spaced) thresholds to be considered

outminmap outminmap, saving directory of Minkowski-map FITS file. Settings will be appended to the name

outascii outfilename/settings.resultDIR, saving directory of of all output text files (ds9-bubbles, linemaps, banana tables, peach tables,...). Settings will be appended to the name. Default: ./results/

outlinedens outlinedens/settings.linedensDIR, saving directory of line density file

outpointspread outpointspread, saving directory of pointspread fits file

outPNGs settings.pngDIR, saving directory of PNGs

objectDIR settings.objectDIR, saving directory of object files (box lists)

s s, rank of Minkowski tensor to be used

smooth smooth, diameter of circle to be smoothed in marching-squares-map. Set either squaresize or smooth

squaresize squaresize, set size of squares for brute-force Minkowski-map. Set either squaresize or smooth. Smooth recommended over squaresize for speed reasons.

erd erd, set diameter of erosion/dilation smoothing of the infile(s) to be done before originalBanana/creating PNGs

- 3d threeD, if **true**: consider all different thresholds, else: consider only maximum. dDfault: false
- avg average, if **true**: average over all calculated thresholds, else: write 3D FITS file. Only relevant if 3d is set. Defalut: false
- abs\_avg absolute\_avg, if **true**: calculate absolute values of maps first, then average. Only relevant if **avg** is set
- arg Take phase instead of absolute value of tensors. Default: false
- makeBanana makeBanana, if **true**: sets make\_minkmap to false, such that the IMTs of the whole (masked) image are calculated, possible to erode/dilate before with erd command. Default: false
- makePeach make\_peach, if **true**: sets make\_minkmap to false, calculates averaged (over maskfile) IMTs of given regions depending on smooth diameter. Relies on pre-existing minkmaps, saves output in outascii. smooth diameters used: 10, 20, 30, 40, 50, 60, 100, 150, 200, 250, 300, 450. Default: false
- makeHist makepointHist, if **true**: sets make\_minkmap to false, calculates histogram of point sources given in stars command for given file. Also creates histogram of whole image, saves output in ./results/. Default: false
- makePointsread make\_pointsread, if **true**: sets make\_minkmap to false, calculates spread of point sources given in stars command for given point spread function (11.07.19: Only gauss ATM) and diameter given by smooth, specific width = smooth/2. Saves as FITS-file, saves output in outpointsread. Default: false
- makePattern make\_patternFromImage, linesorminkmap, Accepts "linedensity", "minkmap", "infile", or "infile.log" as follow-up commands, creates a R-readable table with around 4-5000 points with the number of points proportional to the value at that square, saves in ./results/. Input given by command, output name either name of input file or wavelength (in case of "infile[.log]"). Infile gets smoothed with given windowsize and eroded/dilated with diameter erd, with infile.log also logarithm taken before smoothing. Default: false/"linedensity"
- combineRegions combine\_regions, Takes all maskfiles given by combination and combines bubbles from smaller to larger size; set mask to name output. Outputs various formats: each combined bubble as objectfile containing its original boxes and maskfile listing them, ds9-readable files for each bubble containing all original boxes, R-readable plain table containing average centers of bubbles, ds9-regionfile containing average centers and sizes of all bubbles
- makePNGs makePNGs, if **true**: sets make\_minkmap to false, creates PNGs containing single objects either monochrome or with R = infile, G = greenfile, B = bluefile (default: G=SII, B=SII/H $\alpha$ ) based on the objects in maskfile, possible to erode/dilate images before with erd command, saves output in outPNGs/. Default: false
- monochrome If **true**: make monochrome images based on just infile (default), else read greenfile and bluefile. Default: true

greenfile Filename of green layer of PNG. Default: SII

bluefile Filename of blue layer of PNG. Default: SII/H $\alpha$

makeHedgehog make\_hedgehog, if **true**: reads minkmap with given smooth and standard parameters, writes ds9 region file in ./results/ containing lines at angles of regions with  $q_2 > 0.01$ , threshold given by lineScale sets length scale (length = smooth/2·| $q_2$ |/lineScale), creates fits file in outlinedens with number of lines crossing every block. Also makes bubbles (see below). Default: false

lineThresh line\_thresh, threshold of line density for bubble detection/threshold for point pattern generation (only fields above this considered)

lineScale line\_scale, factor by which length of lines is divided

makeBubbles make\_bubbles, if **true**: reads line density Fitsfile with given wavelength, threshold and smooth value, then takes every block with line number higher than threshold given by lineThresh and  $|q_2| > |q'_2|$  at about three times larger smooth (see switch smooth), creates ds9 region file in ./results/ with boxes centered around those blocks and width= 2·smooth. Writes every box into object file in ./objects/ and names of boxes into maskfile. Default: false

filewithoutWCS filewithoutWCS, to be used for FITS files that were created with gimp and do not contain any coordinate keyword. Simple placeholder settings are added and file is saved with a second .fits ending. Useful for the ds9 "lock: frame: WCS" tool when displaying several images in parallel

readkey WCSkeynames.push\_back([...]), read given key in addition to standard WCS keywords from infile

Syntax of files containing objects:

- combination-file: Contains name of one maskfile in every line listing bubbles of one size, starting at lowest size.
- maskfile: Contains name of one object in every line. For every such object an equally named file containing boxes in objectDIR is expected. Lines beginning with '#' are ignored.
- Files containing boxes to be in-/excluded (objects): One box per line with the following four space- or tab-separated numbers in pixels: RA/x center, DEC/y center, x width, y width (ds9 coordinates in pixels of original images, y axis flipped compared to internal coords)
- histfile: Contains name of one ds9 point-objectlist in every line. For every such line an equally named file containing point-regions in ds9 image coordinates in objectDIR is expected. Lines beginning with '#' are ignored.
- Files containing point-regions are expected to be of the form \*([ra],[dec])\* in the image coordinate system of the raw images. ds9 format is preferred.

Example commands:

Typical Minkowski map for one threshold:

```
./banana in myfile.fits filePrefix halpha outminmap ./minmapfolder/ s 2 smooth 10 maxt 5
```

Typical Minkowski map averaging several thresholds:

```
./banana in myfile.fits filePrefix halpha outminmap ./minmapfolder/ s 2 smooth 10 3d true  
avg true maxt 40 mint 0.1 numt 9
```

Typical banana (for eroded/dilated image):

```
./banana in myfile.fits filePrefix halpha outascii ./results/ makeBanana true s 2 3d true avg  
true maxt 20 mint 0.1 numt 17 mask allObjects erd 3
```

Typical peach(requires Minkmaps to exist):

```
./banana in myfile.fits filePrefix halpha outascii ./results/ s 2 3d true maxt 40 mint 0.1 numt  
9 avg true makePeach true mask ./objectlists/allObjects
```

Typical region-PNGs (for eroded/dilated image):

```
./banana in myfile.fits filePrefix halpha outPNGs ./PNGfolder/ objectDIR ./objects/ erd 3  
monochrome false greenfile mygreenfile.fits bluefile mybluefile.fits mask ./objectlists/usethe.txt
```

Typical hedgehog:

```
./banana in myfile.fits filePrefix halpha outlinedens ./linedens/ outminmap ./minmapfolder/  
makeHedgehog true s 2 smooth 40 mint 0.1 maxt 40 numt 9 3d true avg true lineScale 0.4  
lineThresh 20
```

Typical bubbles:

```
./banana in myfile.fits filePrefix halpha outlinedens ./linedens/ outminmap ./minmapfolder/  
makeBubbles true smooth 40 mint 0.1 maxt 40 numt 9 lineScale 0.4 lineThresh 21
```

Typical histogram:

```
./banana in myfile.fits filePrefix halpha outascii ./results/ stars ./files/allObjectLists maxt  
1. mint -1. numt 100 makeHist true histapp aSpecificThing
```

Typical pointspread:

```
./banana in myFileWithSameSizeAndCoords.fits filePrefix halpha outascii ./results/ make-  
Pointspread true smooth 250 stars ./files/allObjectLists
```

Typical point pattern:

```
./banana in myfile.fits filePrefix halpha outascii ./results/ outlinedens ./linedens/ makePat-  
tern linedensity smooth 40 s 2 avg true lineScale 0.3 lineThresh 12
```

Typical region combination:

```
./banana in myFileOrElseItCrashes.fits combineRegions true combination bubbles_sorted_listoflists  
mask bubbles_sorted_combined_list
```