How to use the PynPoint module *TwoDMaps*

Gabriele Cugno

June 29, 2017

1 What is needed

Prior to the use of the *TwoDMaps* module, the data have to be reduced and prepared. This includes: dark subtraction, bad pixel cleaning, cut (the images must have the desired size), alignment and centering, background subtraction, stacking. Furthermore, the 'NEW_PARA' keyword must be inserted in the Header. This consists in an array of de-rotation angles, used to align all the images to the same direction at the end of the PSF subtraction. The datacube has to be imported in the Workspace directory through the *ReadFitsCubesDirectory* module. The *TwoDMaps* module will have access to the data only through this database.

A master frame of the stellar PSF must be prepared before running the module. This will be rescaled and introduced in the images as fake a fake signal. It has to be a .fits file.

2 Parameters

raw_data_in_tag: (String) Tag of the data, through which the module can access to the images in the database.
psf_in_file: (String) This image will be used for the creation of fake planets. The image can be passed as the path to a single .fits file (datacubes are not supported). The image is assumed to be centered on the central star, and can be cut to a different pixel size using the parameters *cutting_psf* and *psf_cut_ext*.
working_place_in: (String) It points to the working directory, where the database can be found and where the module should work.
input_place_in: (String) It points to the input directory of the pipeline. This parameter is not important for the results, but is needed to define pipelines during the analysis.
output_place_in: (String) It points to the output directory of the pipeline. This parameter is not important for the results, but is needed to define pipelines during the analysis.
inner_mask: (Float) Size of the inner mask to be placed on top of the central star during the PSF subtraction. It must be given as percentage of the whole image. Default is 0.1.
pc_number: (Integer) Amount of principal components of the stellar PSF one wants to subtract.
radius: (Float) Desired radius for the apertures (in pixels). Given the definition of FPF used in this function (see Mawet et al. (2014)), it is recommended to use as a radius half of the FWHM of the given wavelength.
rough_mag_ext: (Float) This value will be used as a starting point: the first magnitude contrast must corresponds to a 'visible' fake positive planet in the final image (i.e.: FPF below the threshold). The function checks if this is true for each position on the grid and eventually adjusts the rough mag ext value. The module then proceeds to increase the magnitude contrast until the fake positive planet disappears (i.e.: FPF above the threshold). For time sake, it should be set to be the almost the maximal contrast that can be reached.
stat: (String) It represents the statistics used during the the final step of the PSF subtraction. It could be 'mean', 'median', 'clean'. Default is 'mean'.
name_in: (String) It represents the name of the module. Default id "TwoDMaps".
cl: (Float between 0 and 1, not percentage) Confidence Level that will be used to set the FPF threshold. It is recommended to change this value carefully: if the *cl* is too small (i.e.: 0.68), this will result in a high FPF threshold (of order 10^-1) and it is likely that some positions of the dataset already have an FPF below this value. In these cases the function cannot perform a proper analysis and the magnitude

contrast is automatically set to a default value. The resulting 2D map and contrast curve will therefore be less precise and less useful. Default is 0.9999 (it roughly corresponds to a 5σ threshold).
radial_step_pix: (Integer) Spacing of the grid in the radial direction (in integer pixels). The function uses this value to decide how many positions can be used in the radial direction, taking into account the size of the frames, the inner maskradius and (eventually) the psf cut. This value should be chosen taking into account the pixel size of the frames, so that at least two positions along a direction are probed. Otherwise the interpolation does not work properly. Default is 5. N.b.: it is possible that the spacing in the radial direction varies of 1-2 pixel from this radial step pix value, since the function slightly adjusts the positions on the grid if they fall outside the image boundaries or inside the inner mask (also taking into account the size of the psf cut). Default is 5.
angular_step_deg: (Integer) Spacing of the grid in the azimuthal direction (in degrees). This value is used to create the final list of positions that will be tested (i.e.: the grid), given the radial step pix, the inner mask and (eventually) the psf cut. The value must not exceed 120, so that at least three different directions are probed. Otherwise the interpolation does not work properly. Default is 30 (i.e.: 12 different azimuthal directions are probed). Default is 30.
mag_step: (Float) Width of the steps with which the magnitude contrast is progressively increased until the fake positive planet disappears (FPF above the threshold). Default is 0.2.
cutting_psf: (Bool) Whether or not to cut the unsaturated image of the central star (unsat psf), before using it to create fake planets. Default is False.
psf_cut_ext: (Integer) If cutting_psf=True, the unsaturated image of the central star (unsat_psf) will be cut to an image of size psf_cut_ext×psf_cut_ext pixels. Default is 20.
method: (String) How to search for the best position given the initial planet position. Possibilities are:
'exact' : the given planet position is assumed to be the best one.
'search' : the best position is found performing a search for the local maximum pixel value around the given planet position in a square of side = (2×radius). The best position found in this way is in integer pixels.
'fit' : the best position is found fitting a 2D gaussian on the given planet position. The best position found in this way allows for floating pixels values.
Default is 'exact'.
fake_planets_out_tag: (String) Name with which the frames after the insertion of the fake signal are saved into the database.
fake_planets_psf_sub_out_tag: (String) Name with which the frames after the insertion of the fake signal and the PSF subtraction are saved into the database.
savefolder: (String) If save=True, this is the path where to save the results. If 'None', the results are saved in the same folder that contains the raw data.
plot: (Bool) If True, the function will plot three figures: the original image together with the positions that have been tested, the contrast curve and the 2D detection map. The figures can be saved as .pdf files if save=True.
save: (Bool) Whether or not to save the final results. If True, the function will save as .pdf files the images explained above. The 2D detection map is also saved as a .fits file. The pixel positions of the grid, together with the corresponding maximum positions on the final image and the magnitude contrast for the desired CL are saved as an ASCII table in a .txt file. The mean values for the magnitude contrast and the radial distance used for the contrast curve are also saved as an ASCII table. Default is False.
dir_fake: (String) Path to a folder where to save the frames that contain the fake planet. This should be an empty folder and is not important for the results. Default is 'Planet_fake'.
mag default: (Float) Default magnitude contrast value to be assigned to a position, in the case in which the FPF at that position is already below threshold, prior to any fake positive planet insertion (i.e.: if, given the desired cl that position would result in a detection). Default is 0.
subpix_precision: (Integer) Subpixel precision used to insert the fake negative planets. Default is 1.

3 Description

The function creates a set of positions that will be used as a grid (specified through the *radial_step_pix* and the *angular_step_deg* parameters) and will be investigated. In a first phase the positions on the horizontal line from the center to the right side of the image, with distance from each other equal the *radial_step_pix*, are chosen, taking into consideration also the central mask and the dimension of the

inserted rescaled PSF. In a second phase they are rotated by the angular step around the center of the images for the number of times necessary to complete a round. This process gives the positions the algorithm would investigate, distributed on $\frac{360°}{\text{angular step}}$ lines in the radial direction from the center.
At each position, the routine inserts a fake planet with an estimated magnitude contrast given by the user, and then calculates the FPF value of its aperture. If the FPF value is larger then the threshold, it inserts a planet 1 mag brighter until the value gets smaller than it. As soon as the FPF lays below the threshold, the brightness of the planet is changed again, with a contrast 0.2 mag higher, until the FPF crossed the threshold value. The final maximal magnitude for a detection is given by the linear interpolation of the contrast as a function of the FPF at the position of the threshold. When every position has a magnitude value, a 2D map of the surrounding region of the star can be produced, linearly interpolating the contrast values calculated at each position. This shows where the analysis achieves lower brightnesses.
Once the 2D map is completed, the module averages azimuthally the values of the positions at the same distance from the center, in order to obtain a radial profile of the reachable contrast. This final operation results in a contrast curve, which represents the sensitivity that can be reached with the current techniques and with the used data reduction as a function of the angular separation from the star.

4 Output

The module creates the contrast curve and the 2D detection map for the desired Confidence Level and plots them on the screen if requested. It is also possible to plot the positions that have been tested. These plots can be saved as .pdf files (using the save parameter). The 2D detection map can be saved as a .fits file if requested. It is also possible to save, as ASCII tables in two .txt files, the values used to create the contrast curve, as well as the pixel positions of the grid together with the correspondent maximum positions on the final image and the magnitude contrasts.
It is possible to specify where to save the results using the savefolder parameter. The function does not return anything.