SPCA Documentation

Release 0.1

Lisa Dang and Taylor James Bell

May 28, 2020

API Table of Contents:

SPCA is an open-source, modular, and automated pipeline for Spitzer Phase Curve Analyses.

CHAPTER 1

Installation Instructions

To install SPCA, run the following in a terminal:

```
git clone git@github.com:lisadang27/SPCA.git
cd SPCA
pip install .
```

Please note however that SPCA is in a state of alpha testing and is still under development. Frequent changes are expected over the upcoming few months as we finalize some aspects and encorporate PLD analyses as well as PSF fitting and nearby companion removal using PSF subtraction.

CHAPTER 2

Package Usage

1. To use SPCA, you must first download your data from the Spitzer Heritage Archive: https://sha.ipac.caltech.edu/applications/Spitzer/SHA/. Place the downloaded zip files in a directory with the same name as the planet (exluding spaces).

Most of the following commands have an .ipynb ending and .py ending option available, where the .py version is optimized for analyzing many data sets and the .ipynb file is optimized for viewing the analysis of a single data set. Each file has a portion at the top where you can set parameters which will determine the techniques.

2. Next, use the Make_Directory_Structure file to extract the data and setup the required directories.
3. Then use the Everything_Photometry file to perform a suite of different photometries on your data and have the code automatically select the best photometry (selecting the photometry that gives the lowest scatter after smoothing the raw flux by a boxcar filter of a width provided by the user).
4. Then use the QuickLook file to ensure that you have looked at the raw data and to determine whether you want to remove the first AOR (in case it is a short AOR before PCRS peak-up was used). By looking at the raw data, you can gain some insight into how successful different decorrelation models might be.
5. Then decorrelate the data using the Poly-BLISS-GP file. Most parameters here are explained with a nearby comment. One key parameter though is the "mode" which sets the decorrelation method used. Modes that contain "Poly#" use a 2D poly of order #, modes that contain "BLISS" use BiLinearly-Interpolated Subpixel Sensitivity mapping, modes that contain "GP" use a Gaussian Process using x and y centroid positions as covariates, and modes that contain "PLD" use Pixel Level Decorrelation. Each mode is then followed by an underscore and either "v1" or "v2" indicating the use of either a 1st or 2nd order sinusoidal model for the phase variations. If "ellipse" is present, phase variations due to the elliptical shape of the planet are modelled. Any other text can be added to the mode keyword for your own convenience (e.g. "Poly2_v1_run2").
6. Finally, some tables containing a selection of the fitted parameters from each model run can be made using the MakeTables file. These tables will also highlight the best decorrelation method for each analysis, determined using delta-BIC.

2.1 SPCA package

2.1.1 Submodules

2.1.2 SPCA.Photometry_Aperture module

SPCA.Photometry_Aperture.**A_photometry**(*image_data*, *bg_err*, *factor=1*, *ape_sum=None*, *ape_sum_err=None*, *cx=15*, *cy=15*, *r=2.5*, *a=5*, *b=5*, *w_r=5*, *h_r=5*, *theta=0*, *shape='Circular'*, *method='center'*)

Performs aperture photometry, first by creating the aperture (Circular, Rectangular or Elliptical), then it sums up the flux that falls into the aperture.

Parameters

- **image_data** (*3D array*) – Data cube of images (2D arrays of pixel values).
- **bg_err** (*1D array*) – Array of uncertainties on pixel value.
- **factor** (*float, optional*) – Electron count to photon count factor. Default is 1 if none given.
- **ape_sum** (*1D array, optional*) – Array of flux to append new flux values to. If None, the new values will be appended to an empty array
- **ape_sum_err** (*1D array, optional*) – Array of flux uncertainty to append new flux uncertainty values to. If None, the new values will be appended to an empty array.
- **cx** (*int, optional*) – x-coordinate of the center of the aperture. Default is 15.
- **cy** (*int, optional*) – y-coordinate of the center of the aperture. Default is 15.
- **r** (*int, optional*) – If shape is 'Circular', r is the radius for the circular aperture. Default is 2.5.
- **a** (*int, optional*) – If shape is 'Elliptical', a is the semi-major axis for elliptical aperture (x-axis). Default is 5.
- **b** (*int, optional*) – If shape is 'Elliptical', b is the semi-major axis for elliptical aperture (y-axis). Default is 5.
- **w_r** (*int, optional*) – If shape is 'Rectangular', w_r is the full width for rectangular aperture (x-axis). Default is 5.
- **h_r** (*int, optional*) – If shape is 'Rectangular', h_r is the full height for rectangular aperture (y-axis). Default is 5.
- **theta** (*int, optional*) – If shape is 'Elliptical' or 'Rectangular', theta is the angle of the rotation angle in radians of the semimajor axis from the positive x axis. The rotation angle increases counterclockwise. Default is 0.
- **shape** (*string, optional*) – shape is the shape of the aperture. Possible aperture shapes are 'Circular', 'Elliptical', 'Rectangular'. Default is 'Circular'.
- **method** (*string, optional*) – The method used to determine the overlap of the aperture on the pixel grid. Possible methods are 'exact', 'subpixel', 'center'. Default is 'center'.

Returns

ape_sum (1D array) Array of flux with new flux appended. ape_sum_err (1D array) Array of flux uncertainties with new flux uncertainties appended.

Return type tuple

class SPCA.Photometry_Aperture.**TestAperturehotometryMethods**(*methodName='runTest'*)
Bases: unittest.case.TestCase

test_centroiding()

test_circularAperture()

SPCA.Photometry_Aperture.**bgsubtract**(*img_data*, *bg_flux=None*, *bg_err=None*, *bounds=(11, 19, 11, 19)*)
Measure the background level and subtracts the background from each frame.

Parameters

- **img_data** (*ndarray*) – Data cube of images (2D arrays of pixel values).
- **bg_flux** (*ndarray, optional*) – Array of background measurements for previous images. Default is None.
- **bg_err** (*ndarray, optional*) – Array of uncertainties on background measurements for previous images. Default is None.
- **bounds** (*tuple, optional*) – Bounds of box around the target. Default is (11, 19, 11, 19).

Returns

bgsub_data (3D array) Data cube of background subtracted images. bg_flux (1D array) Updated array of background flux measurements for previous images. bg_err (1D array) Updated array of uncertainties on background measurements for previous images.

Return type tuple

SPCA.Photometry_Aperture.**binning_data**(*data*, *size*)
Median bin an array.

Parameters

- **data** (*1D array*) – Array of data to be binned.
- **size** (*int*) – Size of bins.

Returns

binned_data (1D array) Array of binned data. binned_data_std (1D array) Array of standard deviation for each entry in binned_data.

Return type tuple

SPCA.Photometry_Aperture.**centroid_FWM**(*image_data*, *xo=None*, *yo=None*, *wx=None*, *wy=None*, *scale=1*, *bounds=(14*, *18*, *14*, *18)*)
Gets the centroid of the target by flux weighted mean and the PSF width of the target.

Parameters

- **image_data** (*ndarray*) – Data cube of images (2D arrays of pixel values).
- **xo** (*list, optional*) – List of x-centroid obtained previously. Default is None.
- **yo** (*list, optional*) – List of y-centroids obtained previously. Default is None.
- **wx** (*list, optional*) – List of PSF width (x-axis) obtained previously. Default is None.
- **wy** (*list, optional*) – List of PSF width (x-axis) obtained previously. Default is None.
- **scale** (*int, optional*) – If the image is oversampled, scaling factor for centroid and bounds, i.e, give centroid in terms of the pixel value of the initial image.

- **bounds** (*tuple, optional*) – Bounds of box around the target to exclude background . Default is (14, 18, 14, 18).

Returns

xo, yo, wx, wy (list, list, list, list). The updated lists of x-centroid, y-centroid, PSF width (x-axis), and PSF width (y-axis).

Return type tuple

SPCA.Photometry_Aperture.**get_fnames**(*directory*, *AOR_snip*, *ch*)

Find paths to all the fits files.

Parameters

- **directory** (*string*) – Path to the directory containing all the Spitzer data.
- **AOR_snip** (*string*) – Common first characters of data directory eg. 'r579'.
- **ch** (*string*) – Channel used for the observation eg. 'ch1' for channel 1.

Returns

fname, lens (list, list). List of paths to all bcd.fits files, number of files for each AOR (needed for adding correction stacks).

Return type tuple

SPCA.Photometry_Aperture.**get_lightcurve**(*datapath*, *savepath*, *AOR_snip*, *channel*, *subarray*, *save=True*, *save_full='/ch2_datacube_full_AORs579.dat'*, *bin_data=True*, *bin_size=64*, *save_bin='/ch2_datacube_binned_AORs579.dat'*, *plot=True*, *plot_name='Lightcurve.pdf'*, *oversamp=False*, *saveoversamp=True*, *reuse_oversamp=False*, *planet='CoRoT-2b'*, *r=2.5*, *shape='Circular'*, *edge='hard'*, *addStack=False*, *stackPath=''*, *ignoreFrames=None*, *maskStars=None*, *moveCentroid=False*, ***kwargs*)

Given a directory, looks for data (bcd.fits files), opens them and performs photometry.

Parameters

- **datapath** (*string*) – Directory where the spitzer data is stored.
- **savepath** (*string*) – Directory the outputs will be saved.
- **AORsnip** (*string*) – Common first characters of data directory eg. 'r579'
- **channel** (*string*) – Channel used for the observation eg. 'ch1' for channel 1
- **subarray** (*bool*) – True if observation were taken in subarray mode. False if observation were taken in full-array mode.
- **shape** (*string, optional*) – shape is the shape of the aperture. Possible aperture shapes are 'Circular', 'Elliptical', 'Rectangular'. Default is 'Circular'.
- **edge** (*string, optional*) – A string specifying the type of aperture edge to be used. Options are 'hard', 'soft', and 'exact' which correspond to the 'center', 'subpixel', and 'exact' methods. Default is 'hard'.
- **save** (*bool, optional*) – True if you want to save the outputs. Default is True.

- **save_full** (*string, optional*) – Filename of the full unbinned output data. Default is '/ch2_datacube_full_AORs579.dat'.
- **bin_data** (*bool, optional*) – True you want to get binned data. Default is True.
- **bin_size** (*int, optional*) – If bin_data is True, the size of the bins. Default is 64.
- **save_bin** (*string, optional*) – Filename of the full binned output data. Default is '/ch2_datacube_binned_AORs579.dat'.
- **plot** (*bool, optional*) – True if you want to plot the time resolved lightcurve. Default is True.
- **plot_name** (*string, optional*) – If plot and save is True, the filename of the plot to be saved as. Default is True.
- **oversamp** (*bool, optional*) – True if you want to oversample you image. Default is False.
- **save_oversamp** (*bool, optional*) – True if you want to save oversampled images. Default is True.
- **reuse_oversamp** (*bool, optional*) – True if you want to reuse oversampled images that were previously saved. Default is False.
- **planet** (*string, optional*) – The name of the planet. Default is CoRoT-2b.
- **r** (*float, optional*) – The radius to use for aperture photometry in units of pixels. Default is 2.5 pixels.
- **ignoreFrames** (*list, optional*) – frame to remove first-frame systematic).
- **maskStars** (*list, optional*) – An array-like object where each element is an array-like object with the RA and DEC coordinates of a nearby star which should be masked out when computing background subtraction.
- **moveCentroid** (*bool, optional*) – True if you want the centroid to be centered on the flux-weighted mean centroids (will default to 15,15 when a NaN is returned), otherwise aperture will be centered on 15,15 (or 30,30 for 2x oversampled images). Default is False.
- ****kwargs** (*dictionary*) – Other arguments passed on to A_photometry.

Raises Error – If Photometry method is not supported/recognized by this pipeline.

SPCA.Photometry_Aperture.**get_stacks**(*calDir*, *dataDir*, *AOR_snip*, *ch*)

Find paths to all the background subtraction correction stacks FITS files.

Parameters

- **calDir** (*string*) – Path to the directory containing the correction stacks.
- **dataDir** (*string*) – Path to the directory containing the Spitzer data to be corrected.
- **AOR_snip** (*string*) – Common first characters of data directory eg. 'r579'.
- **ch** (*string*) – Channel used for the observation eg. 'ch1' for channel 1.

Returns List of paths to the relevant correction stacks

Return type list

SPCA.Photometry_Aperture.**get_time**(*hdu_list*, *time*, *ignoreFrames*)

Gets the time stamp for each image.

Parameters

- **hdu_list** (*list*) – content of fits file.

- **time** (*ndarray*) – Array of existing time stamps.
- **ignoreFrames** (*ndarray*) – Array of frames to ignore (consistently bad frames).

Returns Updated time stamp array.

Return type ndarray

SPCA.Photometry_Aperture.**oversampling**(*image_data*, *a=2*)

First, substitutes all invalid/sigma-clipped pixels by interpolating the value, then oversamples the image.

Parameters

- **image_data** (*ndarray*) – Data cube of images (2D arrays of pixel values).
- **a** (*int, optional*) – Sampling factor, e.g. if a = 2, there will be twice as much data points in the x and y axis. Default is 2. (Do not recommend larger than 2)

Returns Data cube of oversampled images (2D arrays of pixel values).

Return type ndarray

SPCA.Photometry_Aperture.**sigma_clipping**(*image_data, filenb=0, fname=['not provided'], tossed=0, badframetable=None, bounds=(13, 18, 13, 18), sigma=4, maxiters=2*)

Sigma clips bad pixels and mask entire frame if the sigma clipped pixel is too close to the target.

Parameters

- **image_data** (*ndarray*) – Data cube of images (2D arrays of pixel values).
- **filenb** (*int, optional*) – Index of current file in the 'fname' list (list of names of files) to keep track of the files that were tossed out. Default is 0.
- **fname** (*list, optional*) – List of names of files to keep track of the files that were tossed out.
- **tossed** (*int, optional*) – Total number of image tossed out. Default is 0 if none provided.
- **badframetable** (*list, optional*) – List of file names and frame number of images tossed out from 'fname'.
- **bounds** (*tuple, optional*) – Bounds of box around the target. Default is (13, 18, 13, 18).

Returns

sigma_clipped_data (3D array) - Data cube of sigma clipped images (2D arrays of pixel values). tossed (int) - Updated total number of image tossed out. badframetable (list) - Updated list of file names and frame number of images tossed out from 'fname'.

Return type tuple

2.1.3 SPCA.photometryBackend module

SPCA.photometryBackend.**comparePhotometry**(*basepath*, *planet*, *channel*, *AOR_snip*, *ignoreFrames*, *addStack*, *highpassWidth=5*, *trim=False*, *trimStart=None*, *trimEnd=False*)

SPCA.photometryBackend.**create_folder**(*fullname*, *auto=False*)

Create a folder unless it exists.

Parameters

- **fullname** (*string*) – Full path to the folder to be created.
- **auto** (*bool, optional*) – If the folder already exists, should the folder just be skipped (True) or should the user be asked whether they want to overwrite the folder or change the folder name (False, Default).

Returns The final name used for the folder.

Return type string

SPCA.photometryBackend.**get_RMS**(*Run_list*, *channel*, *AOR_snip*, *highpassWidth*, *trim=False*, *trimStart=0*, *trimEnd=0*)

SPCA.photometryBackend.**get_data**(*folderdata*, *channel*, *AOR_snip*)

SPCA.photometryBackend.**get_fnames**(*directory*, *tag='um'*)
Find paths to all the fits files.

Parameters

- **directory** (*string object*) – Path to the directory containing all the Spitzer data.
- **AOR_snip** (*string object*) – Common first characters of data directory eg. 'r579'
- **ch** (*string objects*) – Channel used for the observation eg. 'ch1' for channel 1

Returns

- **fname** (*list*) – List of paths to all bcd.fits files.
- **len(fnames)** (*int*) – Number of fits file found.

SPCA.photometryBackend.**get_full_data**(*foldername*, *channel*, *AOR_snip*)

SPCA.photometryBackend.**highpassflist**(*signal*, *highpassWidth*)

SPCA.photometryBackend.**run_photometry**(*basepath*, *addStack*, *planet*, *channel*, *subarray*, *AOR_snip*, *ignoreFrames*, *maskStars*, *photometryMethod*, *shape*, *edge*, *moveCentroid*, *radius*)

2.1.4 SPCA.frameDiagnosticsBackend module

SPCA.frameDiagnosticsBackend.**bgnormalize**(*image_data*)
Compute the normalized background from each stack.

Parameters **image_data** (*ndarray*) – FITS images stack.

Returns The background in each frame of a datacube, normalized by the median within that datacube.

Return type ndarray

SPCA.frameDiagnosticsBackend.**get_stacks**(*stackpath*, *datapath*, *AOR_snip*, *ch*)
Find paths to all the correction stack FITS files.

Parameters

- **stackpath** (*string*) – Full path to the folder containing background correction stacks.
- **datapath** (*string*) – Full path to the data folder containing the AOR folders with images to be corrected.
- **AOR_snip** (*string*) – AOR snippet used to figure out what folders contain data.
- **ch** (*string*) – String specifying which channel is being used.

Returns The calibration FITS file that should be used for background subtraction correction.

Return type ndarray

SPCA.frameDiagnosticsBackend.**get_stats**(*data*, *median_arr*, *std_arr*)
Compute the median and std. dev. from an array of data and add it to the previously computed values.

Parameters

- **data** (*ndarray*) – The array to get information from.
- **median_arr** (*ndarray*) – The previously computed median values to be appended to.
- **std_arr** (*ndarray*) – The previously computed std. dev. values to be appended to.

Returns

median_arr (ndarray; the median of the data), std_arr (ndarray; the std. dev. of the data).

Return type tuple

SPCA.frameDiagnosticsBackend.**load_data**(*path*, *AOR*)
Compute the normalized background from each stack.

Parameters **image_data** (*ndarray*) – FITS images stack.

Returns

flux (ndarray; the aperture sum from each frame, normalized by the median flux from its datacube), bg (ndarray; the background flux from each frame, normalized by the median background from its datacube), xdata (ndarray; the x-centroid from each frame, normalized by the median x-centroid from its datacube), ydata (ndarray; the y-centroid from each frame, normalized by the median y-centroid from its datacube), psfwx (ndarray; the x PSF width from each frame, normalized by the median x PSF width from its datacube), psfwy (ndarray; the y PSF width from each frame, normalized by the median y PSF width from its datacube), beta (ndarray; the noise pixel parameter from each frame, normalized by the median noise pixel parameter from its datacube).

Return type tuple

SPCA.frameDiagnosticsBackend.**noisepixparam**(*image_data*)
Compute the noise pixel parameter.

Parameters **image_data** (*ndarray*) – FITS images stack.

Returns The noise pixel parameter for each image in the stack.

Return type list

SPCA.frameDiagnosticsBackend.**run_diagnostics**(*planet*, *channel*, *AOR_snip*, *basepath*, *addStack*, *nsigma=3*)
Run frame diagnostics and choose which frames within a datacube are consistently bad and should be discarded.

Parameters

- **planet** (*string*) – The name of the planet.
- **channel** (*string*) – The channel being analyzed.
- **AOR_snip** (*string*) – AOR snippet used to figure out what folders contain data.
- **basepath** (*string*) – The full path to the folder containing folders for each planet.
- **addStack** (*bool*) – Whether or not to add a background subtraction correction stack (will be automatically selected if present).

- **nsigma** (*float*) – The number of sigma a frame's median value must be off from the median frame in order to be added to ignore_frames.

Returns The frames whose photometry is typically nsigma above/below the median frame and should be removed from photometry.

Return type list

2.1.5 SPCA.astro_models module

SPCA.astro_models.**area**(*time*, *t_sec*, *per*, *rp*, *inc*, *r2*, *r2off*)

Model the variations in projected area of a bi-axial ellipsoid over time, without assuming elongated axis is the sub-stellar axis.

Parameters

- **time** (*ndarray*) – Array of times at which to calculate the model.
- **t_sec** (*float*) – Time of secondary eclipse.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **r2** (*float*) – Planet radius along sub-stellar axis (in units of stellar radii).
- **r2off** (*float*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees).

Returns Modelled projected area of the ellipsoid over time.

Return type ndarray

SPCA.astro_models.**area_noOffset**(*time*, *t_sec*, *per*, *rp*, *inc*, *r2*)

Model the variations in projected area of a bi-axial ellipsoid over time, assuming elongated axis is the sub-stellar axis.

Parameters

- **time** (*ndarray*) – Array of times at which to calculate the model.
- **t_sec** (*float*) – Time of secondary eclipse.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **r2** (*float*) – Planet radius along sub-stellar axis (in units of stellar radii).

Returns Modelled projected area of the ellipsoid over time.

Return type ndarray

SPCA.astro_models.**check_phase**(*phis*, *A*, *B*, *C=0*, *D=0*)

Check if the phasecurve ever dips below zero, implying non-physical negative flux coming from the planet.

Parameters

- **phis** (*ndarray*) – Array of phases in radians at which to calculate the model, e.g. phis=np.linspace(-np.pi,np.pi,1000).
- **A** (*float*) – Amplitude of the first-order cosine term.

- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float, optional*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float, optional*) – Amplitude of the second-order sine term. Default=0.

Returns True if lightcurve implies non-physical negative flux coming from the planet, False otherwise.

Return type bool

SPCA.astro_models.**eclipse**(*time*, *t0*, *per*, *rp*, *a*, *inc*, *ecc*, *w*, *fp*, *t_sec*)

Get a model secondary eclipse lightcurve.

Parameters

- **time** (*ndarray*) – Array of times at which to calculate the model.
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecc** (*float*) – Orbital eccentricity.
- **w** (*float*) – Longitude of periastron (in degrees).
- **fp** (*float*) – Planet-to-star flux ratio.
- **t_sec** (*float*) – Time of secondary eclipse.

Returns flux. The model eclipse light curve.

Return type ndarray

SPCA.astro_models.**fplanet_model**(*time*, *anom*, *t0*, *per*, *rp*, *a*, *inc*, *ecc*, *w*, *u1*, *u2*, *fp*, *t_sec*, *A*, *B*, *C=0.0*, *D=0.0*, *r2=None*, *r2off=None*)

Model observed flux coming from the planet over time.

Parameters

- **time** (*ndarray*) – Array of times at which to calculate the model.
- **anom** (*ndarray*) – The true anomaly over time.
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecc** (*float*) – Orbital eccentricity.
- **w** (*float*) – Longitude of periastron (in degrees).
- **u1** (*float*) – Limb darkening coefficient 1.
- **u2** (*float*) – Limb darkening coefficient 2.
- **fp** (*float*) – Planet-to-star flux ratio.
- **t_sec** (*float*) – Time of secondary eclipse.

- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float, optional*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float, optional*) – Amplitude of the second-order sine term. Default=0.
- **r2** (*float, optional*) – Planet radius along sub-stellar axis (in units of stellar radii). Default=None.
- **r2off** (*float, optional*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees). Default=None.

Returns Observed flux coming from planet over time.

Return type ndarray

SPCA.astro_models.**ideal_lightcurve**(*time*, *t0*, *per*, *rp*, *a*, *inc*, *ecosw*, *esinw*, *q1*, *q2*, *fp*, *A*, *B*, *C=0*, *D=0*, *r2=None*, *r2off=None*)

Model observed flux coming from the star+planet system over time.

Parameters

- **time** (*ndarray*) – Array of times at which to calculate the model.
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecosw** (*float*) – Eccentricity multiplied by the cosine of the longitude of periastron (value between -1 and 1).
- **esinw** (*float*) – Eccentricity multiplied by the sine of the longitude of periastron (value between -1 and 1).
- **q1** (*float*) – Limb darkening coefficient 1, parametrized to range between 0 and 1.
- **q2** (*float*) – Limb darkening coefficient 2, parametrized to range between 0 and 1.
- **fp** (*float*) – Planet-to-star flux ratio.
- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float, optional*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float, optional*) – Amplitude of the second-order sine term. Default=0.
- **r2** (*float, optional*) – Planet radius along sub-stellar axis (in units of stellar radii). Default=None.
- **r2off** (*float, optional*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees). Default=None.

Returns Observed flux coming from star+planet system over time.

Return type ndarray

SPCA.astro_models.**phase_variation**(*time*, *t_sec*, *per*, *anom*, *ecc*, *w*, *A*, *B*, *C=0*, *D=0*)

Model first- or second-order sinusoidal phase variations.

Parameters

- **time** (*ndarray*) – Array of times at which to calculate the model.
- **t_sec** (*float*) – Time of secondary eclipse.
- **per** (*float*) – Orbital period.
- **anom** (*ndarray*) – The true anomaly over time.
- **ecc** (*float*) – Orbital eccentricity.
- **w** (*float*) – Longitude of periastron (in degrees).
- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float, optional*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float, optional*) – Amplitude of the second-order sine term. Default=0.

Returns Modelled phase variations.

Return type ndarray

SPCA.astro_models.**transit_model**(*time*, *t0*, *per*, *rp*, *a*, *inc*, *ecc*, *w*, *u1*, *u2*)
Get a model transit lightcurve.

Parameters

- **time** (*ndarray*) – Array of times at which to calculate the model.
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecc** (*float*) – Orbital eccentricity.
- **w** (*float*) – Longitude of periastron (in degrees).
- **u1** (*float*) – Limb darkening coefficient 1.
- **u2** (*float*) – Limb darkening coefficient 2.

Returns

flux, t_secondary, anom (ndarray, float, ndarray). Model transit lightcurve, time of secondary eclipse,

Return type tuple

2.1.6 SPCA.detec_models module

SPCA.detec_models.**detec_model_GP**(*input_data*, *gpAmp*, *gpLx*, *gpLy*, *sigF*)
Model the detector systematics with a Gaussian process based on the centroid.

Parameters

- **input_data** (*tuple*) – (flux, xdata, ydata, time, returnGp, astroModel) with dtypes (ndarray, ndarray, ndarray, ndarray, bool, ndarray). Formatted this way to allow for scipy.optimize.minimize optimization.

- **gpAmp** (*float*) – The natural logarithm of the GP covariance amplitude.
- **gpLx** (*float*) – The natural logarithm of the GP covariance lengthscale in x.
- **gpLy** (*float*) – The natural logarithm of the GP covariance lengthscale in y.
- **sigF** (*float*) – The white noise in units of F_star.

Returns The flux variations due to the detector systematics.

Return type ndarray

SPCA.detec_models.**detec_model_PSFW**(*input_data*, *d1=1*, *d2=0*, *d3=0*)

Model the detector systematics with a simple linear model based on the PSF width.

Parameters

- **detec_inputs** (*tuple*) – (x, y, mode) with dtypes (ndarray, ndarray, string). Formatted this way to allow for easy minimization with scipy.optimize.minimize.
- **d1** (*float*) – The constant offset term. #FIX - I don't think this should be here.
- **d2** (*float*) – The slope in sensitivity with the PSF width in the x direction.
- **d3** (*float*) – The slope in sensitivity with the PSF width in the y direction.

Returns The flux variations due to the detector systematics.

Return type ndarray

SPCA.detec_models.**detec_model_bliss**(*signal_input*, *astroModel*)

Model the detector systematics with a BLISS model based on the centroid.

Parameters

- **signal_input** (*tuple*) – (flux, time, psfxw, psfyw, nBin, nData, knotNdata, low_bnd_x, up_bnd_x, low_bnd_y, up_bnd_y, LL_dist, LR_dist, UL_dist, UR_dist, delta_xo, delta_yo, knot_nrst_x, knot_nrst_y, knot_nrst_lin, BLS, NNI, knots_x_mesh, knots_y_mesh, tmask_good_knotNdata, mode) with dtypes (????). # FIX dtypes!
- **astroModel** (*ndarray*) – The modelled astrophysical flux variations.

Returns The flux variations due to the detector systematics.

Return type ndarray

SPCA.detec_models.**detec_model_poly**(*detec_inputs*, *c1*, *c2*, *c3*, *c4*, *c5*, *c6*, *c7=0*, *c8=0*, *c9=0*, *c10=0*, *c11=0*, *c12=0*, *c13=0*, *c14=0*, *c15=0*, *c16=0*, *c17=0*, *c18=0*, *c19=0*, *c20=0*, *c21=0*)

Model the detector systematics with a 2D polynomial model based on the centroid.

Parameters

- **detec_inputs** (*tuple*) – (x, y, mode) with dtypes (ndarray, ndarray, string). Formatted this way to allow for easy minimization with scipy.optimize.minimize.
- **c1--c21** (*float*) – The polynomial model amplitudes.

Returns The flux variations due to the detector systematics.

Return type ndarray

SPCA.detec_models.**hside**(*time*, *s1*, *s2*)

Model the detector systematics with a heaviside step function at one AOR break.

Parameters

- **time** (*ndarray*) – The time.

- **s1** (*float*) – The amplitude of the heaviside step function.
- **s2** (*float*) – The location of the step in the heaviside function.

Returns The flux variations due to the detector systematics.

Return type ndarray

SPCA.detec_models.**signal**(*signal_input*, *t0*, *per*, *rp*, *a*, *inc*, *ecosw*, *esinw*, *q1*, *q2*, *fp*, *A*, *B*, *C*, *D*, *r2*, *r2off*, *c1*, *c2*, *c3*, *c4*, *c5*, *c6*, *c7*, *c8*, *c9*, *c10*, *c11*, *c12*, *c13*, *c14*, *c15*, *c16*, *c17*, *c18*, *c19*, *c20*, *c21*, *d1*, *d2*, *d3*, *s1*, *s2*, *m1*, *gpAmp*, *gpLx*, *gpLy*, *sigF*, *predictGp=True*, *returnGp=False*)

Model the flux variations as a product of astrophysical varations multiplied by a non-uniform detector sensitivity.

This is a super-function that sets up the framework of SPCA. It calls the relevant astrophysical functions and the relevant detector model functions, depending on the value of mode: the last variable in the signal_input parameter.

Parameters

- **signal_input** (*tuple*) – Varying contents depending on the detector model. The last value of the tuple is invariably the mode string.
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecosw** (*float*) – Eccentricity multiplied by the cosine of the longitude of periastron (value between -1 and 1).
- **esinw** (*float*) – Eccentricity multiplied by the sine of the longitude of periastron (value between -1 and 1).
- **q1** (*float*) – Limb darkening coefficient 1, parametrized to range between 0 and 1.
- **q2** (*float*) – Limb darkening coefficient 2, parametrized to range between 0 and 1.
- **fp** (*float*) – Planet-to-star flux ratio.
- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float*) – Amplitude of the second-order sine term. Default=0.
- **r2** (*float*) – Planet radius along sub-stellar axis (in units of stellar radii). Default=None.
- **r2off** (*float*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees). Default=None.
- **c1--c21** (*float*) – The polynomial model amplitudes.
- **d1** (*float*) – The constant offset term. #FIX - I don't think this should be here.
- **d2** (*float*) – The slope in sensitivity with the PSF width in the x direction.
- **d3** (*float*) – The slope in sensitivity with the PSF width in the y direction.
- **s1** (*float*) – The amplitude of the heaviside step function.
- **s2** (*float*) – The location of the step in the heaviside function.

- **m1** (*float*) – The slope in sensitivity over time with respect to time[0].
- **gpAmp** (*float*) – The natural logarithm of the GP covariance amplitude.
- **gpLx** (*float*) – The natural logarithm of the GP covariance lengthscale in x.
- **gpLy** (*float*) – The natural logarithm of the GP covariance lengthscale in y.
- **sigF** (*float*) – The white noise in units of F_star.
- **predictGp** (*bool, optional*) – Should the GP make predictions (True, default), or just return the GP (useful for lnlike).
- **returnGp** (*bool, optional*) – Should the GP model return the GP object (True, useful for lnlike) or not (False, default).

Returns The modelled flux variations due to the astrophysical model modified by the detector model.

Return type ndarray

SPCA.detec_models.**signal_GP**(*signal_input*, *t0*, *per*, *rp*, *a*, *inc*, *ecosw*, *esinw*, *q1*, *q2*, *fp*, *A*, *B*, *C*, *D*, *r2*, *r2off*, *d1*, *d2*, *d3*, *s1*, *s2*, *m1*, *gpAmp*, *gpLx*, *gpLy*, *sigF*, *predictGp=True*, *returnGp=False*)

Model the flux variations as a product of astrophysical varations multiplied by a GP detector sensitivity model.

Parameters

- **signal_input** (*tuple*) – (flux, time, xdata, ydata, psfwx, psfwy, mode) with dtypes (ndarray, ndarray, ndarray, ndarray, ndarray, string).
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecosw** (*float*) – Eccentricity multiplied by the cosine of the longitude of periastron (value between -1 and 1).
- **esinw** (*float*) – Eccentricity multiplied by the sine of the longitude of periastron (value between -1 and 1).
- **q1** (*float*) – Limb darkening coefficient 1, parametrized to range between 0 and 1.
- **q2** (*float*) – Limb darkening coefficient 2, parametrized to range between 0 and 1.
- **fp** (*float*) – Planet-to-star flux ratio.
- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float*) – Amplitude of the second-order sine term. Default=0.
- **r2** (*float*) – Planet radius along sub-stellar axis (in units of stellar radii). Default=None.
- **r2off** (*float*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees). Default=None.
- **d1** (*float*) – The constant offset term. #FIX - I don't think this should be here.
- **d2** (*float*) – The slope in sensitivity with the PSF width in the x direction.

- **d3** (*float*) – The slope in sensitivity with the PSF width in the y direction.
- **s1** (*float*) – The amplitude of the heaviside step function.
- **s2** (*float*) – The location of the step in the heaviside function.
- **m1** (*float*) – The slope in sensitivity over time with respect to time[0].
- **gpAmp** (*float*) – The natural logarithm of the GP covariance amplitude.
- **gpLx** (*float*) – The natural logarithm of the GP covariance lengthscale in x.
- **gpLy** (*float*) – The natural logarithm of the GP covariance lengthscale in y.
- **sigF** (*float*) – The white noise in units of F_star.
- **predictGp** (*bool, optional*) – Should the GP make predictions (True, default), or just return the GP (useful for lnlike).
- **returnGp** (*bool, optional*) – Should the GP model return the GP object (True, useful for lnlike) or not (False, default).

Returns The modelled flux variations due to the astrophysical model modified by the detector model.

Return type ndarray

SPCA.detec_models.**signal_bliss**(*signal_input*, *t0*, *per*, *rp*, *a*, *inc*, *ecosw*, *esinw*, *q1*, *q2*, *fp*, *A*, *B*, *C*, *D*, *r2*, *r2off*, *d1*, *d2*, *d3*, *s1*, *s2*, *m1*)

Model the flux variations as a product of astrophysical varations multiplied by a BLISS detector sensitivity model.

Parameters

- **signal_input** (*tuple*) – (flux, time, psfxw, psfyw, nBin, nData, knotNdata, low_bnd_x, up_bnd_x, low_bnd_y, up_bnd_y, LL_dist, LR_dist, UL_dist, UR_dist, delta_xo, delta_yo, knot_nrst_x, knot_nrst_y, knot_nrst_lin, BLS, NNI, knots_x_mesh, knots_y_mesh, tmask_good_knotNdata, mode) with dtypes (????). # FIX dtypes!
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecosw** (*float*) – Eccentricity multiplied by the cosine of the longitude of periastron (value between -1 and 1).
- **esinw** (*float*) – Eccentricity multiplied by the sine of the longitude of periastron (value between -1 and 1).
- **q1** (*float*) – Limb darkening coefficient 1, parametrized to range between 0 and 1.
- **q2** (*float*) – Limb darkening coefficient 2, parametrized to range between 0 and 1.
- **fp** (*float*) – Planet-to-star flux ratio.
- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float*) – Amplitude of the second-order sine term. Default=0.

- **r2** (*float*) – Planet radius along sub-stellar axis (in units of stellar radii). Default=None.
- **r2off** (*float*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees). Default=None.
- **d1** (*float*) – The constant offset term. #FIX - I don't think this should be here.
- **d2** (*float*) – The slope in sensitivity with the PSF width in the x direction.
- **d3** (*float*) – The slope in sensitivity with the PSF width in the y direction.
- **s1** (*float*) – The amplitude of the heaviside step function.
- **s2** (*float*) – The location of the step in the heaviside function.
- **m1** (*float*) – The slope in sensitivity over time with respect to time[0].

Returns The modelled flux variations due to the astrophysical model modified by the detector model.

Return type ndarray

SPCA.detec_models.**signal_poly**(*signal_input*, *t0*, *per*, *rp*, *a*, *inc*, *ecosw*, *esinw*, *q1*, *q2*, *fp*, *A*, *B*, *C*, *D*, *r2*, *r2off*, *c1*, *c2*, *c3*, *c4*, *c5*, *c6*, *c7*, *c8*, *c9*, *c10*, *c11*, *c12*, *c13*, *c14*, *c15*, *c16*, *c17*, *c18*, *c19*, *c20*, *c21*, *d1*, *d2*, *d3*, *s1*, *s2*, *m1*)

Model the flux variations as a product of astrophysical varations multiplied by a 2D polynomial detector sensitivity model.

Parameters

- **signal_input** (*tuple*) – (flux, time, xdata, ydata, psfwx, psfwy, mode) with dtypes (ndarray, ndarray, ndarray, ndarray, ndarray, ndarray, string).
- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecosw** (*float*) – Eccentricity multiplied by the cosine of the longitude of periastron (value between -1 and 1).
- **esinw** (*float*) – Eccentricity multiplied by the sine of the longitude of periastron (value between -1 and 1).
- **q1** (*float*) – Limb darkening coefficient 1, parametrized to range between 0 and 1.
- **q2** (*float*) – Limb darkening coefficient 2, parametrized to range between 0 and 1.
- **fp** (*float*) – Planet-to-star flux ratio.
- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float*) – Amplitude of the second-order sine term. Default=0.
- **r2** (*float*) – Planet radius along sub-stellar axis (in units of stellar radii). Default=None.
- **r2off** (*float*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees). Default=None.
- **c1--c21** (*float*) – The polynomial model amplitudes.

- **d1** (*float*) – The constant offset term. #FIX - I don't think this should be here.
- **d2** (*float*) – The slope in sensitivity with the PSF width in the x direction.
- **d3** (*float*) – The slope in sensitivity with the PSF width in the y direction.
- **s1** (*float*) – The amplitude of the heaviside step function.
- **s2** (*float*) – The location of the step in the heaviside function.
- **m1** (*float*) – The slope in sensitivity over time with respect to time[0].

Returns The modelled flux variations due to the astrophysical model modified by the detector model.

Return type ndarray

SPCA.detec_models.**tslope**(*time*, *m1*)

Model the detector systematics with a simple slope in time.

Parameters

- **time** (*ndarray*) – The time.
- **m1** (*float*) – The slope in sensitivity over time with respect to time[0].

Returns The flux variations due to the detector systematics.

Return type ndarray

2.1.7 SPCA.bliss module

SPCA.bliss.**bliss_dist**(*xo*, *yo*, *low_bnd_xk*, *up_bnd_xk*, *low_bnd_yk*, *up_bnd_yk*)

Compute the distance from each centroid to its adjacent knots.

Parameters

- **xo** (*ndarray*) – The x-centroids for each data point.
- **yo** (*ndarray*) – The y-centroids for each data point.
- **low_bnd_xk** (*ndarray*) – The x-position of the knot to the left of each centroid).
- **up_bnd_xk** (*ndarray*) – The x-position of the knot to the right of each centroid).
- **low_bnd_yk** (*ndarray*) – The y-position of the knot below each centroid).
- **up_bnd_yk** (*ndarray*) – The y-position of the knot above each centroid).

Returns

LL_dist (ndarray; the distance to the lower-left knot), LR_dist (ndarray; the distance to the lower-right knot), UL_dist (ndarray; the distance to the upper-left knot), UR_dist (ndarray; the distance to the upper-right knot).

Return type tuple

SPCA.bliss.**bound_knot**(*xo*, *yo*, *low_bnd_x*, *up_bnd_x*, *low_bnd_y*, *up_bnd_y*, *low_bnd_xk*, *up_bnd_xk*, *low_bnd_yk*, *up_bnd_yk*, *nData*)

Get the x and y coordinates of the knots adjacted to each centroid.

Parameters

- **xo** (*ndarray*) – The x-centroids for each data point.
- **yo** (*ndarray*) – The y-centroids for each data point.

- **low_bnd_x** (*ndarray*) – The index of the knot to the left of each centroid).
- **up_bnd_x** (*ndarray*) – The index of the knot to the right of each centroid).
- **low_bnd_y** (*ndarray*) – The index of the knot below each centroid).
- **up_bnd_y** (*ndarray*) – The index of the knot above each centroid).
- **low_bnd_xk** (*ndarray*) – The x-position of the knot to the left of each centroid).
- **up_bnd_xk** (*ndarray*) – The x-position of the knot to the right of each centroid).
- **low_bnd_yk** (*ndarray*) – The y-position of the knot below each centroid).
- **up_bnd_yk** (*ndarray*) – The y-position of the knot above each centroid).
- **nData** (*int*) – The number of data points.

Returns

knot_nrst_x (ndarray; the x-index of closest knot), knot_nrst_y (ndarray; the y-index of closest knot).

Return type tuple

SPCA.bliss.**get_knot_bounds**(*knots_x*, *knots_y*, *low_bnd_x*, *up_bnd_x*, *low_bnd_y*, *up_bnd_y*)

Get the x and y coordinates of the knots adjacted to each centroid.

Parameters

- **knots_x** (*ndarray*) – The detector coordinate for the x center of each knot.
- **knots_y** (*ndarray*) – The detector coordinate for the y center of each knot.
- **low_bnd_x** (*ndarray*) – The index of the knot to the left of each centroid.
- **up_bnd_x** (*ndarray*) – The index of the knot to the right of each centroid.
- **low_bnd_y** (*ndarray*) – The index of the knot below each centroid.
- **up_bnd_y** (*ndarray*) – The index of the knot above each centroid.

Returns

low_bnd_xk (ndarray; the x-position of the knot to the left of each centroid), up_bnd_xk (ndarray; the x-position of the knot to the right of each centroid), low_bnd_yk (ndarray; the y-position of the knot below each centroid), up_bnd_yk (ndarray; the y-position of the knot above each centroid).

Return type tuple

SPCA.bliss.**lh_axes_binning**(*xo*, *yo*, *nBin*, *nData*)

Create knots using the fitted centroids and an input number of knots.

Parameters

- **xo** (*ndarray*) – The x-centroids for each data point.
- **yo** (*ndarray*) – The y-centroids for each data point.
- **nBin** (*int*) – The number of knots you want along each axis.
- **nData** (*int*) – The number of data points.

Returns

low_bnd_x (ndarray; the index of the knot to the left of each centroid), up_bnd_x (ndarray; the index of the knot to the right of each centroid), low_bnd_y (ndarray; the index of the knot below each centroid), up_bnd_y (ndarray; the index of the knot above each

centroid), knots_x (ndarray; the detector coordinate for the x center of each knot), knots_y (ndarray; the detector coordinate for the y center of each knot), knotNdata (ndarray; the number of data assoicated with each knot), x_edg (ndarray; the x-coordinates for the edges of the knots), y_edg (ndarray; the y-coordinates for the edges of the knots), knots_x_mesh (ndarray; the 2D array of the x-position of each knot), knots_y_mesh (ndarray; the 2D array of the y-position of each knot).

Return type tuple

SPCA.bliss.**lh_knot_ass**(*knots_pos*, *cents*, *nBin*, *nData*)

Find the two knots adjacted to each knot to later be used with linear interpolation.

Parameters

- **knots_pos** (*ndarray*) – The detector coordinate for the x or y center of each knot.
- **cents** (*ndarray*) – The x or y centroids for each data point.
- **nBin** (*int*) – The number of knots you want along each axis.
- **nData** (*int*) – The number of data points.

Returns

low_bnd (ndarray; the index of the knot to the left of or below each centroid), up_bnd (ndarray; the index of the knot to the right of or above each centroid).

Return type tuple

SPCA.bliss.**map_flux_avgQuick**(*flux*, *astroModel*, *knot_nrst_lin*, *nBin*, *knotNdata*)

Compute the average sensitivity of each knot.

Parameters

- **flux** (*ndarray*) – The flux measurements for each data point.
- **astroModel** (*ndarray*) – The astrophysical model for each data point.
- **knot_nrst_lin** (*ndarray*) – Index of the knot assoicated with each data point.
- **nBin** (*int*) – The number of knots you want along each axis.
- **knotNdata** (*ndarray*) – The number of data assoicated with each knot.

Returns sensMap (The average flux/astroModel (aka ~ sensitivity) value for each knot).

Return type ndarray

SPCA.bliss.**precompute**(*flux*, *time*, *xdata*, *ydata*, *psfxw*, *psfyw*, *mode*, *astroGuess*, *nBin=10*, *savepath=None*, *plot=True*)

Precompute all of the knot associations, etc. that are needed to run BLISS in a fitting routine.

Parameters

- **flux** (*ndarray*) – The flux measurements for each data point.
- **time** (*ndarray*) – The time stamp for each data point.
- **xdata** (*ndarray*) – The x-centroid for each data point.
- **ydata** (*ndarray*) – The y-centroid for each data point.
- **psfxw** (*ndarray*) – The PSF width in the x-direction for each data point.
- **psfyw** (*ndarray*) – The PSF width in the y-direction for each data point.
- **mode** (*string*) – The string specifying which detector and astrophysical models should be used.

- **astroGuess** (*ndarray*) – The astrophysical model for each data point.
- **nBin** (*int*) – The number of knots you want along each axis.
- **savepath** (*string*) – The full path to where you would like to save plots that can be used to debug BLISS.
- **plot** (*boolean*) – Whether or not you want to make plots that can be used to debug BLISS (default is False).

Returns signal_input (All of the inputs needed to feed into the signal_bliss function)

Return type tuple

SPCA.bliss.**which_NNI**(*knotNdata*, *low_bnd_x*, *up_bnd_x*, *low_bnd_y*, *up_bnd_y*)

Figure out which data points can use BLISS and which need to use NNI.

Parameters

- **knotNdata** (*ndarray*) – The number of data assoicated with each knot.
- **low_bnd_x** (*ndarray*) – The index of the knot to the left of each centroid.
- **up_bnd_x** (*ndarray*) – The index of the knot to the right of each centroid.
- **low_bnd_y** (*ndarray*) – The index of the knot below each centroid.
- **up_bnd_y** (*ndarray*) – The index of the knot above each centroid.

Returns

nni_mask (ndarray; boolean array saying which data points will use NNI), bliss_mask (ndarray; boolean array saying which data points will use BLISS).

Return type tuple

2.1.8 SPCA.make_plots module

SPCA.make_plots.**plot_bestfit**(*x*, *flux*, *astro*, *detec*, *mode*, *breaks*, *savepath=None*, *showplot=True*, *peritime=-inf*, *nbin=None*, *fontsize=10*)

SPCA.make_plots.**plot_centroids**(*xdata0*, *ydata0*, *xdata*, *ydata*, *savepath=''*)

Makes a multi-panel plot from photometry outputs.

Parameters

- **xdata0** (*1D array*) – Initial modelled the fluxes for each time stamps. Discarded points not removed.
- **ydata0** (*1D array*) – Initial modelled astrophysical flux variation for each time stamps. Discarded points not removed.
- **xdata** (*1D array*) – Initial modelled the fluxes for each time stamps. Discarded points removed.
- **ydata** (*1D array*) – Initial modelled astrophysical flux variation for each time stamps. Discarded points removed.
- **savepath** (*string*) – Path to directory where the plot will be saved

Returns None or figure if savepath is None

SPCA.make_plots.**plot_init_guess**(*time*, *data*, *astro*, *detec_full*, *savepath=None*)

Makes a multi-panel plots for the initial light curve guesses.

Parameters

- **time** (*1D array*) – Time stamps.
- **data** (*1D array*) – Flux values for each time stamps.
- **astro** (*1D array*) – Initial modelled astrophysical flux variation for each time stamps.
- **detec_full** (*1D array*) – Initial modelled flux variation due to the detector for each time stamps.
- **savepath** (*str*) – Path to directory where the plot will be saved.

Returns None or figure if savepath is None

SPCA.make_plots.**plot_knots**(*xdata*, *ydata*, *flux*, *astroModel*, *knot_nrst_lin*, *tmask_good_knotNdata*, *knots_x*, *knots_y*, *knots_x_mesh*, *knots_y_mesh*, *nBin*, *knotNdata*, *savepath=None*)

Plot the Bliss map

SPCA.make_plots.**plot_photometry**(*time0*, *flux0*, *xdata0*, *ydata0*, *psfxw0*, *psfyw0*, *time*, *flux*, *xdata*, *ydata*, *psfxw*, *psfyw*, *breaks=[]*, *savepath=None*, *peritime=''*)

Makes a multi-panel plot from photometry outputs.

Parameters

- **time0** (*1D array*) – Array of time stamps. Discarded points not removed.
- **flux0** (*1D array*) – Array of flux values for each time stamps. Discarded points not removed.
- **xdata0** (*1D array*) – Initial modelled the fluxes for each time stamps. Discarded points not removed.
- **ydata0** (*1D array*) – Initial modelled astrophysical flux variation for each time stamps. Discarded points not removed.
- **psfxw0** (*1D array*) – Point-Spread-Function (PSF) width along the x-direction. Discarded points not removed.
- **psfyw0** (*1D array*) – Point-Spread-Function (PSF) width along the x-direction. Discarded points not removed.
- **time** (*1D array*) – Array of time stamps. Discarded points removed.
- **flux** (*1D array*) – Array of flux values for each time stamps. Discarded points removed.
- **xdata** (*1D array*) – Initial modelled the fluxes for each time stamps. Discarded points removed.
- **ydata** (*1D array*) – Initial modelled astrophysical flux variation for each time stamps. Discarded points removed.
- **psfxw** (*1D array*) – Point-Spread-Function (PSF) width along the x-direction. Discarded points removed.
- **psfyw** (*1D array*) – Point-Spread-Function (PSF) width along the x-direction. Discarded points removed.
- **break** (*1D array*) – Time of the breaks from one AOR to another.
- **savepath** (*string*) – Path to directory where the plot will be saved
- **pertime** (*float*) – Time of periapsis

Returns None or figure if savepath is None

SPCA.make_plots.**plot_rednoise**(*residuals*, *minbins*, *ingrDuration*, *occDuration*, *intTime*, *mode*, *savepath=None*, *showplot=True*, *showtxt=True*, *savetxt=False*, *fontsize=10*)

SPCA.make_plots.**triangle_colors**(*all_data*, *firstEcl_data*, *transit_data*, *secondEcl_data*, *fname=None*)

Make a triangle plot like figure to help look for any residual correlations in the data.

Parameters

- **all_data** (*list*) – A list of the all of the xdata, ydata, psfxw, psfyw, flux, residuals.
- **firstEcl_data** (*list*) – A list of the xdata, ydata, psfxw, psfyw, flux, residuals during the first eclipse.
- **transit_data** (*list*) – A list of the xdata, ydata, psfxw, psfyw, flux, residuals during the transit.
- **secondEcl_data** (*list*) – A list of the xdata, ydata, psfxw, psfyw, flux, residuals during the second eclipse.
- **fname** (*string, optional*) – The savepath for the plot (or None if you want to return the figure instead).

Returns None or figure if name is None

2.1.9 SPCA.helpers module

SPCA.helpers.**BIC**(*logL*, *Npar*, *Ndat*)

Compute the Bayesian Information Criterion.

Parameters

- **logL** (*float*) – The lnlikelihood.
- **Npar** (*int*) – The number of fitted parameters.
- **Ndat** (*int*) – The number of data fitted.

Returns The Bayesian Information Criterion.

Return type float

SPCA.helpers.**binValues**(*values*, *binAxisValues*, *nbin*, *assumeWhiteNoise=False*)

Bin values and compute their binned noise.

Parameters

- **values** (*ndarray*) – An array of values to bin.
- **binAxisValues** (*ndarray*) – Values of the axis along which binning will occur.
- **nbin** (*int*) – The number of bins desired.
- **assumeWhiteNoise** (*bool, optional*) – Divide binned noise by sqrt(nbinned) (True) or not (False, default).

Returns

binned (ndarray; the binned values), binnedErr (ndarray; the binned errors)

Return type tuple

SPCA.helpers.**binnedNoise**(*x*, *y*, *nbin*)

Compute the binned noise (not assuming white noise)

Parameters

- **x** (*ndarray*) – The values along the binning axis.
- **y** (*ndarray*) – The values which should be binned.
- **nbin** (*int*) – The number of bins desired.

Returns The binned noise (not assuming white noise).

Return type ndarray

SPCA.helpers.**chi2**(*data*, *fit*, *err*)

Compute the chi-squared statistic.

Parameters

- **data** (*ndarray*) – The real y values.
- **fit** (*ndarray*) – The fitted y values.
- **err** (*ndarray or float*) – The y error(s).

Returns The chi-squared statistic.

Return type float

SPCA.helpers.**clip_full_data**(*FLUX*, *FERR*, *TIME*, *XDATA*, *YDATA*, *PSFXW*, *PSFYW*, *nFrames=64*, *cut=0*, *ignore=array([]*, *dtype=float64)*)

Sigma cip the unbinned data.

Parameters

- **flux** (*ndarray*) – Flux extracted for each frame.
- **flux_err** (*ndarray*) – uncertainty on the flux for each frame.
- **time** (*ndarray*) – Time stamp for each frame.
- **xdata** (*ndarray*) – X-coordinate of the centroid for each frame.
- **ydata** (*ndarray*) – Y-coordinate of the centroid for each frame.
- **psfwx** (*ndarray*) – X-width of the target's PSF for each frame.
- **psfwy** (*ndarray*) – Y-width of the target's PSF for each frame.

Returns

flux (ndarray; Flux extracted for each frame), flux_err (ndarray; uncertainty on the flux for each frame), time (ndarray; Time stamp for each frame), xdata (ndarray; X-coordinate of the centroid for each frame), ydata (ndarray; Y-coordinate of the centroid for each frame), psfwx (ndarray; X-width of the target's PSF for each frame), psfwy (ndarray; Y-width of the target's PSF for each frame).

Return type tuple

SPCA.helpers.**evidence**(*logL*, *Npar*, *Ndat*)

Compute the Bayesian evidence.

Parameters

- **logL** (*float*) – The lnlikelihood.
- **Npar** (*int*) – The number of fitted parameters.
- **Ndat** (*int*) – The number of data fitted.

Returns The Bayesian evidence.

Return type float

SPCA.helpers.**expand_dparams**(*dparams*, *mode*)
Add any implicit dparams given the mode (e.g. GP parameters if using a Polynomial model).

Parameters

- **dparams** (*ndarray*) – A list of strings specifying which parameters shouldn't be fit.
- **mode** (*string*) – The string specifying the detector and astrophysical model to use.

Returns The updated dparams array.

Return type ndarray

SPCA.helpers.**getIngressDuration**(*p0_mcmc*, *p0_labels*, *p0_obj*, *intTime*)
Compute the transit/eclipse ingress duration in units of datapoints.

Warning - this assumes a circular orbit!

Parameters

- **p0_mcmc** (*ndarray*) – The array containing the fitted values.
- **p0_labels** (*ndarray*) – The array containing all of the names of the fittable parameters.
- **p0_obj** (*object*) – The object containing the default values for non-fitted variables.
- **intTime** (*float*) – The integration time of each measurement.

Returns The transit/eclipse ingress duration in units of datapoints.

Return type float

SPCA.helpers.**getOccultationDuration**(*p0_mcmc*, *p0_labels*, *p0_obj*, *intTime*)
Compute the full transit/eclipse duration in units of datapoints.

Warning - this assumes a circular orbit!

Parameters

- **p0_mcmc** (*ndarray*) – The array containing the fitted values.
- **p0_labels** (*ndarray*) – The array containing all of the names of the fittable parameters.
- **p0_obj** (*object*) – The object containing the default values for non-fitted variables.
- **intTime** (*float*) – The integration time of each measurement.

Returns The full transit/eclipse duration in units of datapoints

Return type float

SPCA.helpers.**get_data**(*path*)
Retrieve binned data.

Parameters **path** (*string*) – Full path to the data file output by photometry routine.

Returns

flux (ndarray; Flux extracted for each frame), flux_err (ndarray; uncertainty on the flux for each frame), time (ndarray; Time stamp for each frame), xdata (ndarray; X-coordinate of the centroid for each frame), ydata (ndarray; Y-coordinate of the centroid for each frame), psfwx (ndarray; X-width of the target's PSF for each frame), psfwy (ndarray; Y-width of the target's PSF for each frame).

Return type tuple

SPCA.helpers.**get_full_data**(*foldername*, *filename*)
Retrieve unbinned data.

Parameters

- **foldername** (*string*) – Full path to the data file output by photometry routine.
- **filename** (*string*) – File name of the unbinned data file output by photometry routine.

Returns

flux (ndarray; Flux extracted for each frame), flux_err (ndarray; uncertainty on the flux for each frame), time (ndarray; Time stamp for each frame), xdata (ndarray; X-coordinate of the centroid for each frame), ydata (ndarray; Y-coordinate of the centroid for each frame), psfwx (ndarray; X-width of the target's PSF for each frame), psfwy (ndarray; Y-width of the target's PSF for each frame).

Return type tuple

SPCA.helpers.**get_p0**(*function_params*, *fancy_names*, *dparams*, *obj*)
Initialize the p0 variable to the defaults.

Parameters

- **function_params** (*ndarray*) – Array of strings listing all parameters required by a function.
- **fancy_names** (*ndarray*) – Array of fancy (LaTeX or nicely formatted) strings labelling each parameter for plots.
- **dparams** (*ndarray*) – A list of strings specifying which parameters shouldn't be fit.
- **obj** (*object*) – An object containing the default values for all fittable parameters. #FIX: change this to dict later

Returns

p0 (ndarray; the initialized values), fit_params (ndarray; the names of the fitted variables), fancy_labels (ndarray; the nicely formatted names of the fitted variables)

Return type tuple

SPCA.helpers.**lnlike**(*p0*, *signalfunc*, *signal_input*)
Evaluate the ln-likelihood at the position p0.

Note: We assumine that we are always fitting for the photometric scatter (sigF).

Parameters

- **p0** (*ndarray*) – The array containing the n-D position to evaluate the log-likelihood at.
- **signalfunc** (*function*) – The super function to model the astrophysical and detector functions.
- **signal_input** (*list*) – The collection of other assorted variables required for signalfunc beyond just p0.

Returns The ln-likelihood evaluated at the position p0.

Return type float

SPCA.helpers.**lnprior**(*t0*, *per*, *rp*, *a*, *inc*, *ecosw*, *esinw*, *q1*, *q2*, *fp*, *A*, *B*, *C*, *D*, *r2*, *r2off*, *c1*, *c2*, *c3*, *c4*, *c5*, *c6*, *c7*, *c8*, *c9*, *c10*, *c11*, *c12*, *c13*, *c14*, *c15*, *c16*, *c17*, *c18*, *c19*, *c20*, *c21*, *d1*, *d2*, *d3*, *s1*, *s2*, *m1*, *gpAmp*, *gpLx*, *gpLy*, *sigF*, *mode*, *checkPhasePhis*)
Check that the parameters are physically plausible.

Parameters

- **t0** (*float*) – Time of inferior conjunction.
- **per** (*float*) – Orbital period.
- **rp** (*float*) – Planet radius (in units of stellar radii).
- **a** (*float*) – Semi-major axis (in units of stellar radii).
- **inc** (*float*) – Orbital inclination (in degrees).
- **ecosw** (*float*) – Eccentricity multiplied by the cosine of the longitude of periastron (value between -1 and 1).
- **esinw** (*float*) – Eccentricity multiplied by the sine of the longitude of periastron (value between -1 and 1).
- **q1** (*float*) – Limb darkening coefficient 1, parametrized to range between 0 and 1.
- **q2** (*float*) – Limb darkening coefficient 2, parametrized to range between 0 and 1.
- **fp** (*float*) – Planet-to-star flux ratio.
- **A** (*float*) – Amplitude of the first-order cosine term.
- **B** (*float*) – Amplitude of the first-order sine term.
- **C** (*float*) – Amplitude of the second-order cosine term. Default=0.
- **D** (*float*) – Amplitude of the second-order sine term. Default=0.
- **r2** (*float*) – Planet radius along sub-stellar axis (in units of stellar radii). Default=None.
- **r2off** (*float*) – Angle to the elongated axis with respect to the sub-stellar axis (in degrees). Default=None.
- **c1--c21** (*float*) – The polynomial model amplitudes.
- **d1** (*float*) – The constant offset term. #FIX - I don't think this should be here.
- **d2** (*float*) – The slope in sensitivity with the PSF width in the x direction.
- **d3** (*float*) – The slope in sensitivity with the PSF width in the y direction.
- **s1** (*float*) – The amplitude of the heaviside step function.
- **s2** (*float*) – The location of the step in the heaviside function.
- **m1** (*float*) – The slope in sensitivity over time with respect to time[0].
- **gpAmp** (*float*) – The natural logarithm of the GP covariance amplitude.
- **gpLx** (*float*) – The natural logarithm of the GP covariance lengthscale in x.
- **gpLy** (*float*) – The natural logarithm of the GP covariance lengthscale in y.
- **sigF** (*float*) – The white noise in units of F_star.
- **mode** (*string*) – The string specifying the detector and astrophysical model to use.
- **checkPhasePhis** (*ndarray*) – The phase angles to use when checking that the phase-curve is always positive.

Returns The default ln-prior evaluated at the position p0.

Return type float

SPCA.helpers.**lnprob**(*p0*, *signalfunc*, *lnpriorfunc*, *signal_input*, *checkPhasePhis*, *lnpriorcustom=None*)

Evaluate the ln-probability of the signal function at the position p0, including priors.

Parameters

- **p0** (*ndarray*) – The array containing the n-D position to evaluate the log-likelihood at.
- **signalfunc** (*function*) – The super function to model the astrophysical and detector functions.
- **lnpriorfunc** (*function*) – The function to evaluate the default ln-prior.
- **signal_input** (*list*) – The collection of other assorted variables required for signalfunc beyond just p0.
- **checkPhasePhis** (*ndarray*) – The phase angles to use when checking that the phasecurve is always positive.
- **lnpriorcustom** (*function, optional*) – An additional function to evaluate the a user specified ln-prior function (default is None).

Returns The ln-probability evaluated at the position p0.

Return type float

SPCA.helpers.**load_past_params**(*path*)
Load the fitted parameters from a previous run.

Parameters **path** (*string*) – Path to the file containing past mcmc result (must be a table saved as .npy file).

Returns p0 (the previously fitted values)

Return type ndarray

SPCA.helpers.**loglikelihood**(*data*, *fit*, *err*)
Compute the lnlikelihood.

Parameters

- **data** (*ndarray*) – The real y values.
- **fit** (*ndarray*) – The fitted y values.
- **err** (*ndarray or float*) – The y error(s).

Returns The lnlikelihood.

Return type float

SPCA.helpers.**make_lambdafunc**(*function*, *dparams=[]*, *obj=[]*, *debug=False*)
Create a lambda function called dynamic_funk that will fix the parameters listed in dparams with the values in obj.

Note: The module where the original function is needs to be loaded in this file.

Parameters

- **function** (*string*) – Name of the original function.
- **dparams** (*list, optional*) – List of all input parameters the user does not wish to fit (default is None.)
- **obj** (*string, optional*) – Object containing all initial and fixed parameter values (default is None.)
- **debug** (*bool, optional*) – If true, will print mystr so the user can read the command because executing it (default is False).

Returns dynamic_funk (the lambda function with fixed parameters.)

Return type function

class SPCA.helpers.**signal_params**(*name='planet'*, *t0=0.0*, *per=1.0*, *rp=0.1*, *a=8.0*, *inc=90.0*, *ecosw=0.0*, *esinw=0.0*, *q1=0.01*, *q2=0.01*, *fp=0.001*, *A=0.1*, *B=0.0*, *C=0.0*, *D=0.0*, *sigF=0.0003*, *mode=''*)

Bases: object

SPCA.helpers.**time_sort_data**(*flux*, *flux_err*, *time*, *xdata*, *ydata*, *psfxw*, *psfyw*, *cut=0*)

Sort the data in time and cut off any bad data at the start of the observations (e.g. ditered AOR).

Parameters

- **flux** (*ndarray*) – Flux extracted for each frame.
- **flux_err** (*ndarray*) – uncertainty on the flux for each frame.
- **time** (*ndarray*) – Time stamp for each frame.
- **xdata** (*ndarray*) – X-coordinate of the centroid for each frame.
- **ydata** (*ndarray*) – Y-coordinate of the centroid for each frame.
- **psfwx** (*ndarray*) – X-width of the target's PSF for each frame.
- **psfwy** (*ndarray*) – Y-width of the target's PSF for each frame.

Returns

flux (ndarray; Flux extracted for each frame), flux_err (ndarray; uncertainty on the flux for each frame), time (ndarray; Time stamp for each frame), xdata (ndarray; X-coordinate of the centroid for each frame), ydata (ndarray; Y-coordinate of the centroid for each frame), psfwx (ndarray; X-width of the target's PSF for each frame), psfwy (ndarray; Y-width of the target's PSF for each frame).

Return type tuple

SPCA.helpers.**walk_style**(*ndim*, *nwalk*, *samples*, *interv*, *subsamp*, *labels*, *fname=None*)

Make a plot showing the evolution of the walkers throughout the emcee sampling.

Parameters

- **ndim** (*int*) – Number of free parameters
- **nwalk** (*int*) – Number of walkers
- **samples** (*ndarray*) – The ndarray accessed by calling sampler.chain when using emcee
- **interv** (*int*) – Take every 'interv' element to thin out the plot
- **subsamp** (*int*) – Only show the last 'subsamp' steps
- **labels** (*ndarray*) – The fancy labels for each dimension
- **fname** (*string, optional*) – The savepath for the plot (or None if you want to return the figure instead).

Returns None

CHAPTER 3

Indices and tables

- genindex
- modindex

CHAPTER 4

Contributions

Lisa Dang contributed the initial code and idea for SPCA, and wrote much of the photometry and fitting routines, as well as the polynomial and PLD decorrelation methods.

Both authors worked extensively to debug the code and simplify the user experience.

Taylor James Bell further generalized and streamlined SPCA, allowing it to be run quickly and easily for any given planet with minimal effort. Taylor also contributed much of the documentation, and the GP decorrelation method.

CHAPTER 5

Acknowledgements

We thank Joel Schwartz for his aid in writing out BLISS decorrelation method. We also thank Dylan Keating for alpha testing SPCA.

CHAPTER 6

License & Attribution

If you make use of SPCA in your work, please cite the Dang et al. (2018) paper that was the first to use this pipeline (arXiv, ADS, BibTeX).

Python Module Index

S

Index

H

I

L

M

N

O

P

R

S

T

W