pspy Documentation

Release 0.0.1

Thibaut Louis, Steve Choi, Dongwon 'DW' Han

Nov 28, 2018

CONTENTS:

CHAPTER

ONE

SO POWER SPECTRUM PIPELINE

A framework for creating the power spectrum pipeline for the Simons Observatory. Documentation is still under development, but details about how to contribute can be found in CONTRIBUTING.md.

Installation

PSpipe will have 3 mains module, pspy, psc, and pslike. To install *pspy*, just clone this repository and run `bash python setup.py install ` (add *–user* if you don't have permissions, which is probably the case at e.g. NERSC).

Once installed, you can test the installation by running `bash python test/test_projection.py `

which project a HEALPIX CMB map into a CAR template.

The SO map class

We should be able to work with both CAR and HEALPIX pixellisation. We have written a so map class that can be used to: read/write/plot/SHT/upgrade/downgrade Healpix and CAR maps (using healpy and pixell functionalities). Simple examples for how to use this class can be found in test/

`python from pspy import so_map m=so_map.read_map('map.fits') #map.fits can be a HEALPIX or a CAR map m.info() m.plot() `

CHAPTER

TWO

1 REFERENCE

2.1 1.1 so_config - basic configuration

@brief: a module to handle base configuration

pspy.so_config.**get_data_dir**()
return default data directory

pspy.so_config.**get_output_dir**()
return default output directory

2.2 1.2 so_map - a module for handling healpix and car maps

@brief: so map class for handling healpix and car maps @author: this is a wrapper around healpix and enlib (pixell).

pspy.so_map.**car2car**(*map*, *template*)
@brief project a CAR map into another CAR map, see the pixell enmap.project documentation

pspy.so_map.**car_template**(*ncomp*, *ra0*, *ra1*, *dec0*, *dec1*, *res*)
@brief create a so map template with car pixellisation in equ coordinates. @param ncomp: the number of component of the map can be 1 or 3 @param ra0,dec0,ra1,dec1: in degrees @param res: resolution in arcminute @return: a so template with car pixellisation

pspy.so_map.**getbox**(*ra0*, *ra1*, *dec0*, *dec1*)
@brief create box in equatorial coordinate @param ra0,dec0,ra1,dec1 in degrees

pspy.so_map.**healpix2car**(*map*, *template*, *lmax=None*)
@brief Project a HEALPIX so map into a CAR so map the projection will be done in harmonic space, you can specify a lmax to choose a range of multipoles considered in the projection. If the coordinate of the map and the template differ, a rotation will be performed @param template: the car template you want to project into @param lmax: the maximum multipole in the HEALPIX map to project @return the projected map in the template pixellisation and coordinates

pspy.so_map.**healpix_template**(*ncomp*, *nside*, *coordinate=None*)
@brief create a so map template with healpix pixellisation. @param ncomp: the number of component of the map can be 1 or 3 @param nside @param coordinate: the coordinate of the template can be gal or equ

pspy.so_map.**read_map**(*file*, *coordinate=None*, *verbose=False*)
@brief Reads a FITS file and creates a so map object out of it. The FITS file can be either an enmap object or a healpix object. @return a so_map instance.

class pspy.so_map.**so_map**
Class describing a so map object.

copy()
@brief Create a copy of the so map object.

downgrade(*factor*)
@brief downgrade the so map @param factor need to be a factor of 2 @return a so_map instance downgraded by factor.

info(*showHeader=False*)
@brief Print information about the so map object.

plot(*color='planck'*, *color_range=None*, *file_name=None*, *ticks_spacing_car=1*, *title=''*, *cbar=True*)
@brief Plot a so map, color is a maplotlib colormap or the planck colormap. @param color: a colormap @param color_range: should be a scalar if you want to plot only a single component, and a len(3) list if you want to plot T,Q,U @param file_name: file_name is the name of the png file that will be created, if None the plot will be displayed @param title: the title of the plot @param cbar: wether you display the colorbar or not @param ticks_spacing_CAR: for CAR plot, choose the spacing of the ticks

synfast(*clfile*)
@brief generate a cmb gaussian simulation in so map @param clfile: a lensed power spectrum file from CAMB @return: the so map with lensed CMB

upgrade(*factor*)
@bried upgrade the so map @param factor need to be a factor of 2 @return a so_map instance upgraded by factor.

write_map(*file_name*)
@brief Write the so map.

2.3 1.3 so_cov - a module for covariance matrix estimation

@brief: python routines for covariance matrix estimation.

2.4 1.4 so_mcm - a module for mode coupling calculation

@brief: python routines for mode coupling calculation. Should include an option for pure B mode estimation

2.5 1.5 so_spectra - a module for power spectra estimation and debiasing

@brief: python routines for power spectra estimation and debiasing.

2.6 1.6 so_window - a module for window function generation

@brief: python routines for window function generation

2.7 1.7 so_map_processing - a module for map processing

2.8 1.8 sht_tools - a helper module for spherical harmonic transformation

CHAPTER

THREE

CONTRIBUTING

How to contribute

This file contains guidelines to the expected mode of work for anyone contributing to the power spectrum pipeline. Please read and try to follow them as closely as possible.

Adding new functionality If you think something is missing in *PSpipe* or any of the associated pipelines, proceed as follows:

- Open a new issue describing the new functionality you'd like to see addressed.
- If you want to implement this yourself (or if you want to address an issue opened by someone else), create a new branch and implement any relevant modifications there.
- Make a pull request (PR) for your new branch as soon as you're happy to do it. You don't have to finish the work to open the PR. Often it's useful to open PRs early on so other contributors know that you're working on this specific issue. For the same reason, it's probably a good idea to advertise the new work in the slack channel if you think others will be interested.
- As soon as you think your work is done, ask for volunteers to review your PR. Note that you can nominate specific PR reviewers by clicking on 'Reviewers', on the top right corner of your PR.
- When your PR gets accepted, merge it onto master!

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- search

PYTHON MODULE INDEX

p

INDEX