
hpproj Documentation

Release 0.4.0

Alexandre Beelen, Marian Douspis

Nov 22, 2017

Contents

1	Features	3
2	Installation	5
3	Contribute	7
4	Support	9
5	License	11
6	Contents:	13
6.1	Installation	13
6.2	Usage	14
7	Indices and tables	17

HealPixProjection is a project to allow easy and efficient projection of healpix maps onto planar grids. It can be used as a standalone program `cutsky`

```
$ cutsky 0.0 0.0 --mapfilenames HFI_SkyMap_857_2048_R2.00_full.fits
```

or as a python module

```
from hproproj import cutsky
result = cutsky([0.0, 0.0], maps={'HFI 857':
                                   {'filename': 'HFI_SkyMap_857_2048_R2.00_full.fits'}
                                   } )
```


CHAPTER 1

Features

- Galactic and equatorial system supported
- All projection system from `wcslib`
- Project several healpix maps at once, efficiently !
- Output in `fits`, `png` or `votable` for the central point source photometry

See usage for more information on how to use `cutsky`

CHAPTER 2

Installation

Install hpproj using pip :

```
$ pip install hpproj
```

or by running setuptools on [source](#). For more information see the installation page.

CHAPTER 3

Contribute

- [Issues Tracker](#)
- [Source Code](#)

CHAPTER 4

Support

If you are having issues, please let us know.

CHAPTER 5

License

This project is licensed under the LGPL+3.0 license.

6.1 Installation

hpproj is tested against python 2.7 and 3.5 and can be installed using *pip* or from *source*

6.1.1 pip

This will install the latest release of hpproj

6.1.2 source

```
$ git clone https://git.ias.u-psud.fr/abeelen/hpproj.git
$ cd hpproj
$ python setup.py install
```

This will install the master tree of hpproj. It is probably wiser to checkout a specific version before installation

```
$ git clone https://git.ias.u-psud.fr/abeelen/hpproj.git
$ cd hpproj
$ git checkout 0.4
$ python setup.py install
```

6.1.3 Dependencies

hpproj require the following librairies

- numpy>=1.11
- matplotlib>=1.5
- astropy>=1.2

- healpy>=1.9
- photutils>=0.2
- wcsaxes>=0.9

The specific versioning are those you are being used in the test suit. Both *pip* and *source* install should install those library if they are missing.

6.2 Usage

There is two main way to use “hproj”, the first way is to use the standalone program on the command line, this will efficiently produce cuts for similar maps, or use it programmatically from within a python script or program which will offer an additional speed-up on high memory system.

6.2.1 From the command line - cutsky

The command line program is called `cutsky` and takes 3 arguments at minimum, the longitude and latitude of the desired projection (by default in galactic coordinate, but see below) and a list of healpix map to cut from :

```
$ cutsky 0.0 0.0 --mapfilenames data/HFI_SkyMap_100_2048_R2.00_full.fits data/HFI_
↳ SkyMap_857_2048_R2.00_full.fits
```

by default this will produce two png files centered on galactic longitude and latitude (0,0). Fits images of central photometries can be obtain using the `--fits` or `--phot` options. Help on `cutsky` can be obtain by

```
$ cutsky -h

usage: cutsky [-h] [--npix NPIX | --radius RADIUS] [--pixsize PIXSIZE]
              [--coordframe {galactic,fk5}]
              [--ctype {AZP,SZP,TAN,STG,SIN,ARC,ZPN,ZEA,AIR,CYP,CEA,CAR,MER,COP,COE,
↳ COD,COO,SFL,PAR,MOL,AIT,BON,PCO,TSC,CSC,QSC,HPX,XPH}]
              [--mapfilenames MAPFILENAMES [MAPFILENAMES ...]] [--fits]
              [--png] [--votable] [--outdir OUTDIR] [-v | -q] [--conf CONF]
              lon lat
```

Reproject the spherical sky onto a plane.

positional arguments:

lon	longitude of the projection [deg]
lat	latitude of the projection [deg]

optional arguments:

-h, --help	show this help message and exit
--npix NPIX	number of pixels (default 256)
--radius RADIUS	radius of the requested region [deg]
--pixsize PIXSIZE	pixel size [arcmin] (default 1)
--coordframe {galactic,fk5}	coordinate frame of the lon. and lat. of the projection and the projected map (default: galactic)
--ctype {AZP,SZP,TAN,STG,SIN,ARC,ZPN,ZEA,AIR,CYP,CEA,CAR,MER,COP,COE, COD, COO, SFL, ↳ PAR,MOL,AIT,BON,PCO,TSC,CSC,QSC,HPX,XPH}	any projection code supported by wcslib (default:TAN)

input maps:

one of the two options must be present

```

--mapfilenames MAPFILENAMES [MAPFILENAMES ...]
                                absolute path to the healpix maps
--conf CONF                      absolute path to a config file

output:
--fits                          output fits file
--png                          output png file (Default: True if nothing else)
--votable                      output votable file
--outdir OUTDIR                output directory (default:..)

general:
-v, --verbose                  verbose mode
-q, --quiet                    quiet mode

```

It takes two float arguments, the latitude and longitude center of the requested projection, either in galactic or equatorial coordinate frame (controled by the `--coordframe` option) and a list of healpix maps, either on the command line with the `--mapfilenames` argument or describe in a config file (with the `--conf` option). Several other optional arguments can also be set like `--npix` the number of pixels, their size (`--pixsize`) or the projection type `--ctype`.

The cutted maps can be saved as fits (`--fits`) or png (`--png`) and central circular aperture photometry can be performed and saved as a votable (`--votable`). The output products directory can be tune using the `--outdir` option. All theses options can also be provided by the config file.

The config file follows a simple ini syntax with a global section `[cutsky]` to gather all previous options. The rest of the sections is used to describe the healpix maps used by `cutsky`. The section name `[test]` will be used as a legend and index by `cutsky`.

```

[cutsky]
npix = 256
pixsize = 2
coordframe = galactic
png = True

[SMICA]
filename = hproj/data/CMB_I_SMICA_128_R2.00.fits
doCut = False

[HFI 100]
filename = hproj/data/HFI_SkyMap_100_128_R2.00_RING.fits

[HFI 857]
filename = hproj/data/HFI_SkyMap_857_128_R2.00_NESTED.fits
doCut = True
doContour = True

```

6.2.2 As a function call `-cutsky()`

It is also possible to call `cutsky` from a python program or script, as a function :

```

from hproj import cutsky
result = cutsky([0.0, 0.0], maps=[ ( 'data/HFI_SkyMap_100_2048_R2.00_full.fits', {
    ↪ 'legend': 'HFI 100' } ),
                                   ( 'data/HFI_SkyMap_857_2048_R2.00_full.fits', {
    ↪ 'legend': 'HFI 857', 'doContour': True } ) ] )

```

The first argument is the latitude and longitude of the requested maps, by default in galactic frame (see the `coordframe` keyword), and the `maps` list define the healpix maps.

This will produce a list of dictionaries containing 4 keys:

- `legend`,
- `fits` an `~astropy.io.fits.ImageHDU`,
- `png`, a b61encoded png image of the fits
- `phot`, the corresponding photometry

Additional parameters can be passed to the function :

- `patch=[256, 1]` : the size of the patch in pixel, and the size of the pixels in arcmin
- `ctype='TAN'` : the desired type of projection

6.2.3 As an object - `CutSky()`

It is however more efficient to use `cutsky` as an object :

```
from hproproj import CutSky
cutsky = CutSky([ ( 'hproproj/data/HFI_SkyMap_100_2048_R2.00_full.fits', {'legend':
↪ 'HFI 100'} ),
                  ( 'hproproj/data/HFI_SkyMap_857_2048_R2.00_full.fits', {'legend':
↪ 'HFI 857', doContour: True} ) ], low_mem=False)

lonlat = [0.0, 0.0]
result = cutsky.cut_fits(lonlat) # Will only produce the 'fits' key
result = cutsky.cut_png(lonlat)  # Will only produce the 'png' key (and 'fits' if_
↪ absent)
result = cutsky.cut_phot(lonlat) # Will only produce the 'phot' key (and fits' if_
↪ absent)
```

The result product should be similar to the `cutsky()` function. However with the `low_mem` keyword the healpix maps will be read only once in memory, for all `cut_*` calls. Similar to `cutsky()` several keyword parameters can be passed to `CutSky()` :

- `npix=256` : the size of the patch in pixels
- `pixsize=1` : the size of the pixels in arcmin
- `ctype='TAN'` : the desired type of projection

6.2.4 Limitations

All the healpix maps *must* have a proper header defining their :

- frame using the `COORDSYS` keyword,
- order using the `ORDERING` keyword.

CHAPTER 7

Indices and tables

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