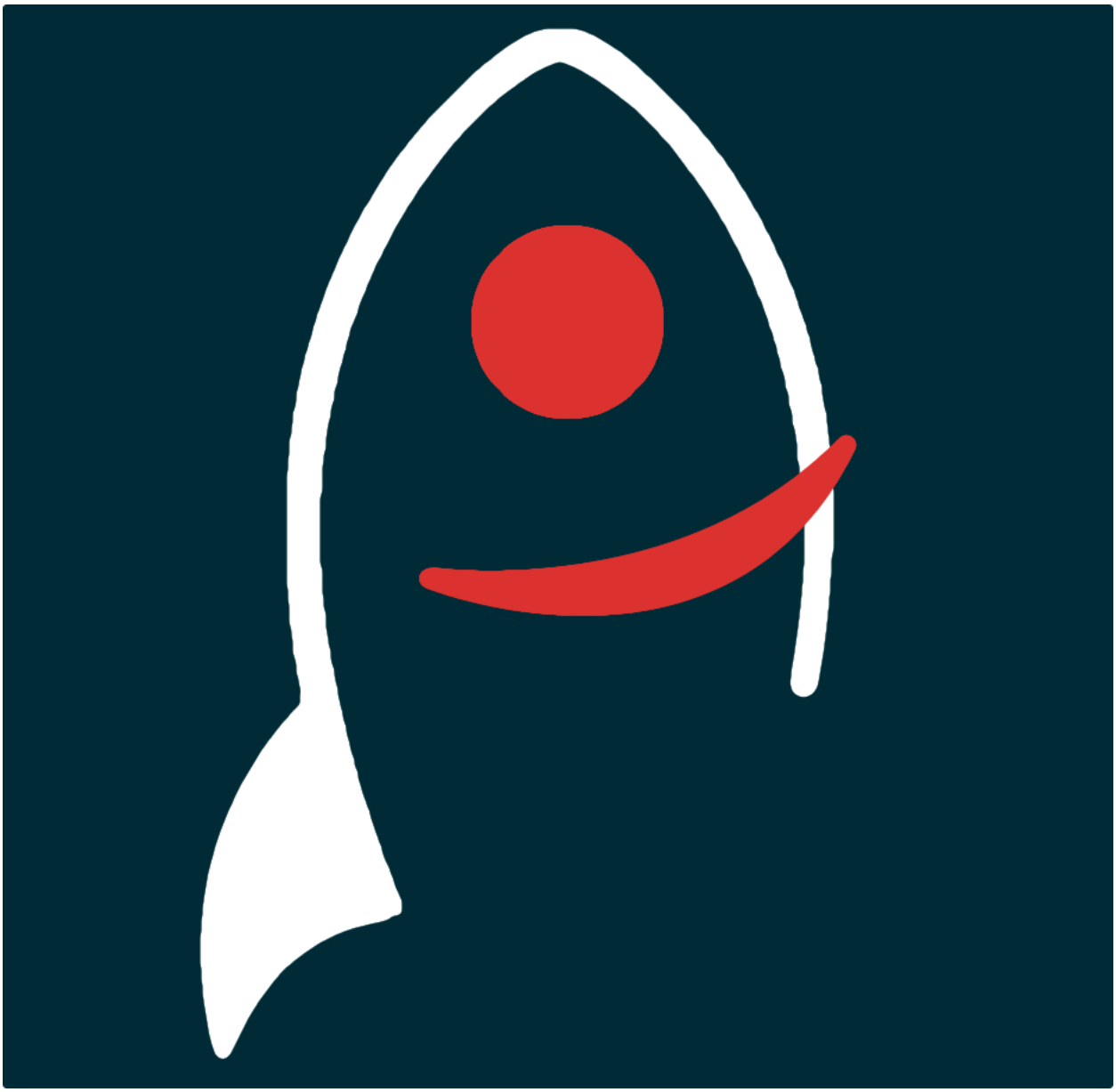

skytag Documentation

Release v0.3.2

Dave Young

2024

CONTENTS

Annotate transient sources or galaxies with the percentage credibility region they reside within on a given HealPix sky map.

Documentation for skytag is hosted by Read the Docs (development version and main version). The code lives on github. Please report any issues you find here. If you want to contribute, pull requests are welcomed!

true

CHAPTER

ONE

FEATURES

- A command-line tool to report the credibility region a sky-location is found within on a HealPix skymap.
- Providing a MJD will also return the time since the map event.
- A python interface to provide the same functionality reported above, but can handle large lists of sky-locations or transient events.
- works well in conjunction with gocart.

CHAPTER

TWO

INSTALLATION

The easiest way to install skytag is to use `conda`:

```
conda create -n skytag python=3.11 pip skytag -c conda-forge
conda activate skytag
```

To upgrade to the latest version of skytag use the command:

```
conda upgrade skytag -c conda-forge
```

It is also possible to install via pip if required:

```
pip install skytag
```

To check installation was successful run `skytag -v`. This should return the version number of the install.

CHAPTER

THREE

COMMAND-LINE

Here is the command-line usage:

```
Usage:
    skytag <ra> <dec> <mapPath>
    skytag <ra> <dec> <mjd> <mapPath>
```

If you need an example skymap, download one from here.

For example, to find the probability of the location RA=170.343532, Dec=-40.532255 then run:

```
skytag 170.343532 -40.532255 bayestar.multiorder.fits
```

This returns:

> This location is found in the 74.55 credibility region of the map.

If you also supply an MJD:

```
skytag 170.343532 -40.532255 60065.2232 bayestar.multiorder.fits
```

We get:

> This transient is found in the 74.55 credibility region, and occurred 2.85564 days after the map event.

Finally, we can request the localised event distance for this specific sky-position be returned:

```
skytag -d 170.343532 -40.532255 bayestar.multiorder.fits
```

> This transient is found in the 74.55% credibility region. At this sky-position the map event is localised to a distance of 75.03 ($\pm$19.72) Mpc.

CHAPTER

FOUR

PYTHON API

To use skytag in your own Python code, see here.

CHAPTER

FIVE

GOCART

skyTag works very well in conjunction with gocart, a tool to consume GCN Kafka alert streams and convert HealPix skymaps.

CHAPTER

SIX

HOW TO CITE SKYTAG

If you use `skytag` in your work, please cite using the following BibTeX entry:

```
@software{Young_skytag,
    author = {Young, David R.},
    doi = {10.5281/zenodo.7977905},
    license = {GPL-3.0-only},
    title = ,
    url = {https://zenodo.org/doi/10.5281/zenodo.7977905}
}
```

6.1 Release Notes

v0.3.2 - March 27, 2024

- **ENHANCEMENT**: probability densities can now be (optionally) returned from `prob_at_location`. Thanks to @RoyWilliams.

v0.3.1 - May 30, 2023

- **FEATURE**: localisation distances are returned for each sky-position (if requested)
- **ENHANCEMENT**: probability clearly reported as a % on the command-line
- **ENHANCEMENT**: support for bilby generated maps

v0.2.0 - April 28, 2023

- First release

6.2 Modules

skytag.commonutils	*common tools used throughout package*
skytag.utKit	*Unit testing tools*

6.2.1 commonutils *(module)*

common tools used throughout package

Functions

`prob_at_location`(ra, dec, mapPath[, mjd, ...])	*Return the probability contour a given sky-location resides within in a heaplix skymap*

6.2.2 utKit *(module)*

Unit testing tools

Classes

`utKit`(moduleDirectory[, dbConn])	*Override dryx utKit*

6.3 Functions

`skytag.commonutils.prob_at_location`	*Return the probability contour a given sky-location resides within in a heaplix skymap*

6.3.1 prob_at_location (*function*)

`prob_at_location`(*ra*, *dec*, *mapPath*, *mjd=False*, *log=False*, *distance=False*, *probdensity=False*)

Return the probability contour a given sky-location resides within in a heaplix skymap

Key Arguments:

- `ra` – right ascension in decimal degrees (float or list)
- `dec` – declination in decimal degrees (float or list)
- `mapPath` – path the the HealPix map
- `mjd` – MJD of transient event (e.g. discovery date). If supplied, a time-delta from the map event is returned (float or list)
- `log` – logger
- `distance` – return also a distance (if present). Default False
- `probdensity` – return also the probability density. Default False

Return:

- `probs` – a list of probabilities the same length as the input RA and Dec lists. One probability per location.

- `timeDeltas` – a list of time-deltas (days) the same length as the input RA, Dec and MJD lists. One delta per input MJD giving the time since the map event. Only returned if MJD is supplied.
- `distance` – a list of location specific distances and distance-sigmas. A list of tuples. Only returned if `distance=True`.
- `probdensity` – a list of location specific probability densities. Only returned if `probdensity=True`.

You can pass a single coordinate to return the probability contour that location lies within on the skymap:

```
from skytag.commonutils import prob_at_location
prob = prob_at_location(
    ra=10.343234,
    dec=14.345532,
    mapPath="/path/to/bayestar.multiorder.fits"
)
```

Or a list of coordinates:

```
from skytag.commonutils import prob_at_location
prob = prob_at_location(
    log=log,
    ra=[10.343234, 170.343532],
    dec=[14.345532, -40.532255],
    mapPath=pathToOutputDir + "/bayestar.multiorder.fits"
)
```

You can also pass in a list of MJDs to also return a list of time-deltas:

```
from skytag.commonutils import prob_at_location
prob, deltas = prob_at_location(
    log=log,
    ra=[10.343234, 170.343532],
    dec=[14.345532, -40.532255],
    mjd=[60034.257381, 60063.257381],
    mapPath=pathToOutputDir + "/bayestar.multiorder.fits"
)
```

Here probs = `[100.0, 74.55]` and deltas = `[-28.11018, 0.88982]`. Deltas are in days, with negative deltas occurring before the map event.

You can also request distance estimates at the locations:

```
from skytag.commonutils import prob_at_location
prob, deltas, distance = prob_at_location(
    log=log,
    ra=[10.343234, 170.343532],
    dec=[14.345532, -40.532255],
    mjd=[60034.257381, 60063.257381],
    mapPath=pathToOutputDir + "/bayestar.multiorder.fits",
    distance=True
)
```

The distances are returned as a list of tuples (distance in MPC, distance sigma in MPC)

You can also request probability density (per steradian) at the locations,

```
from skytag.commonutils import prob_at_location
prob, deltas, distance, probdensity = prob_at_location(
    log=log,
    ra=[10.343234, 170.343532],
    dec=[14.345532, -40.532255],
    mjd=[60034.257381, 60063.257381],
    mapPath=pathToOutputDir + "/bayestar.multiorder.fits",
    distance=True,
    probdensity=True
)
```

6.4 A-Z Index

PYTHON MODULE INDEX

INDEX

M

P

S