
ObservationTools Documentation

Release 0.1

IA

Nov 02, 2018

Contents:

1	Installation	3
1.1	With git	3
1.2	Without git	3
1.3	Updates	3
2	Visibility	5
2.1	Modes	5
2.2	Other options	7
3	Radial Velocity	9
3.1	Parameter file	9
3.2	Usage examples	9
3.3	Modes	11
4	Contributors	13
5	Indices and tables	15

A set of tools and handy scripts for astronomers.

CHAPTER 1

Installation

1.1 With git

If you have git installed (or if you want to install git and use it for the first time), then the tools can be installed with the following few commands in the terminal:

```
git clone https://github.com/iastro-pt/ObservationTools
cd ObservationTools
pip install -r requirements.txt # You may need to use sudo here
```

1.2 Without git

If you do not have git installed, you can just download the entire directory [here](#):

```
unzip ObservationTools-master.zip
cd ObservationTools-master
pip install -r requirements.txt # You may need to use sudo here
```

1.3 Updates

If you want to update your tools and installed it with *git*, simply change the directory to this folder and do a *git pull*. If you don't used git, you have to do the installation again as described above.

The script `visibility.py` is used to plot the observability of objects to aid the planning of astronomical observations. It is inspired by [STARALT](#) and [PyAstronomy's Transit Visibility](#) tools.

2.1 Modes

Currently there are two user modes of visibility. `staralt` (default) and `starobs`. The usage of these is outlined in the following sections.

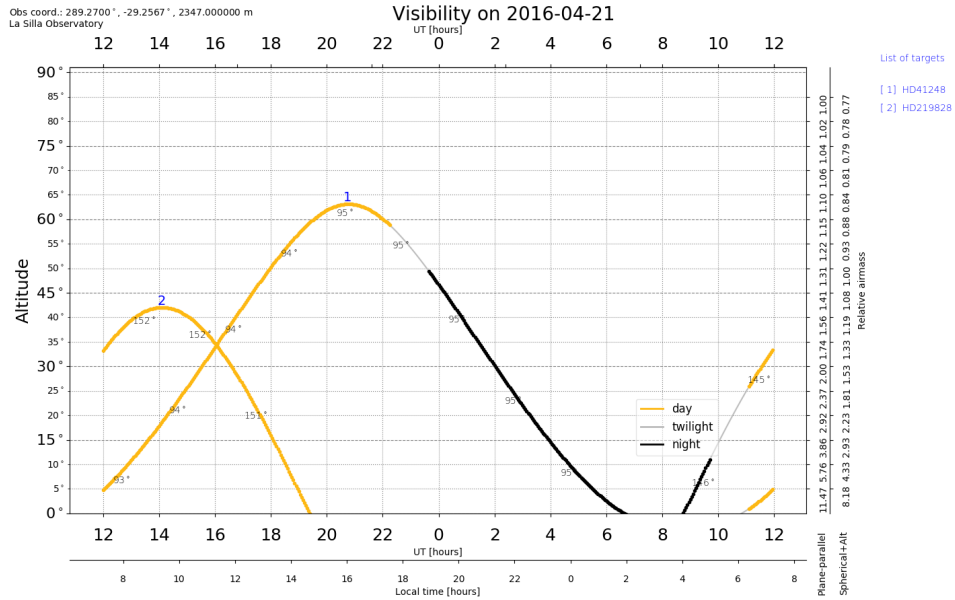
2.1.1 `staralt`

The `staralt` mode displays the altitude verse time of targets for a particular night.

For example:

```
python visibility.py HD41248,HD219828 -s esolasilla -d 2016-04-21
```

Results in the following image.



It is the default mode if no mode is specified. If the `-d`, or `--date` is not provided with the `YYYY-MM-DD` format is then it defaults to today/.

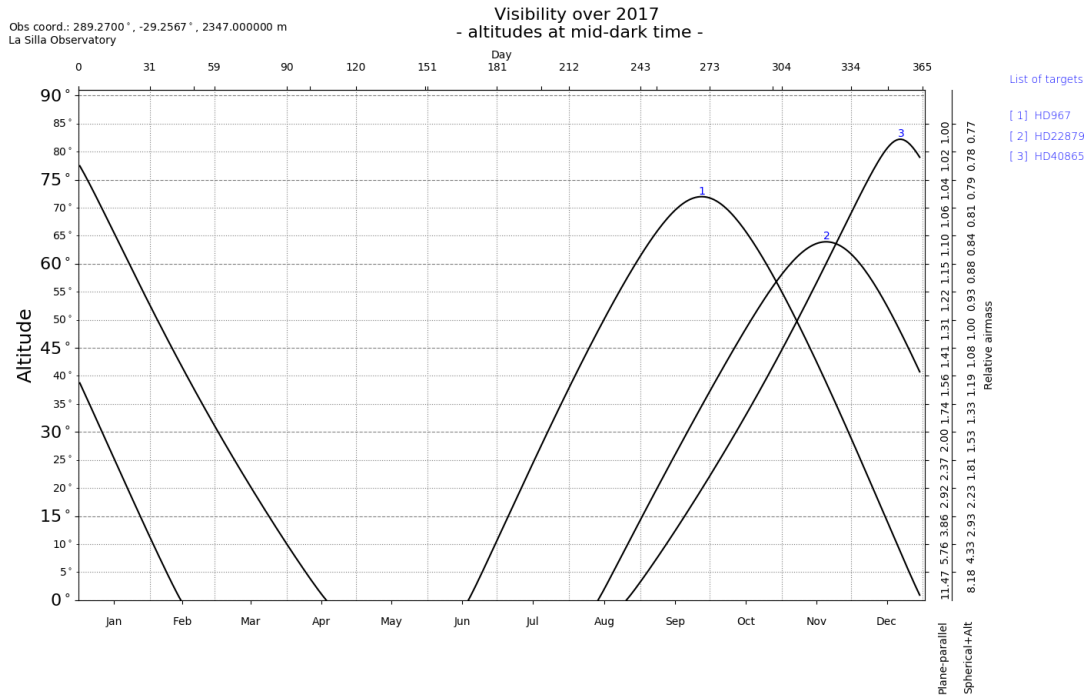
The `observatory` location can be specified using the `-s` or `--site` flag. The default observatory is ESO To find the list of available observatories and name codes run:

```
from __future__ import print_function, division
from PyAstronomy import pyasl
# List all available observatory data
pyasl.listObservatories()
```

2.1.2 starobs

The `starobs` mode shows how the altitude, at the *mid-dark time*, of each target changes over the course of the year. e.g.

```
python visibility.py HD967,HD22879,HD40865 -m starobs
```



Only the year *YYYY* is to be specified for the `--date` flag in this mode.

2.2 Other options

2.2.1 coordinates

The `-c` flag can be used to just return the coordinates of the targets in STARALT format then exit.

CHAPTER 3

Radial Velocity

The `rv.py` script allows you to perform radial velocity (rv) calculations and create radial velocity plots to aid in planning of radial velocity observations.

3.1 Parameter file

`rv.py` requires a parameter file to specify the orbital parameters of the system you wish to analysis. A template is provided in `data/template_params.txt` to help you get started. Comment lines starting with `#` and in-line comments are ignored.

For a basic rv calculations the standard rv parameters are required, `k1` [km/s], `omega` [deg], `eccentricity`, `tau` [days], `period` [days], as well as the `name` parameter.

If the mean system rv offset, `mean_val` (usually referred to as γ), is not provided in the parameter file it is set to 0 km/s. The `ignore_mean` keyword in some functions can also be used to use a 0 km/s `mean_val`.

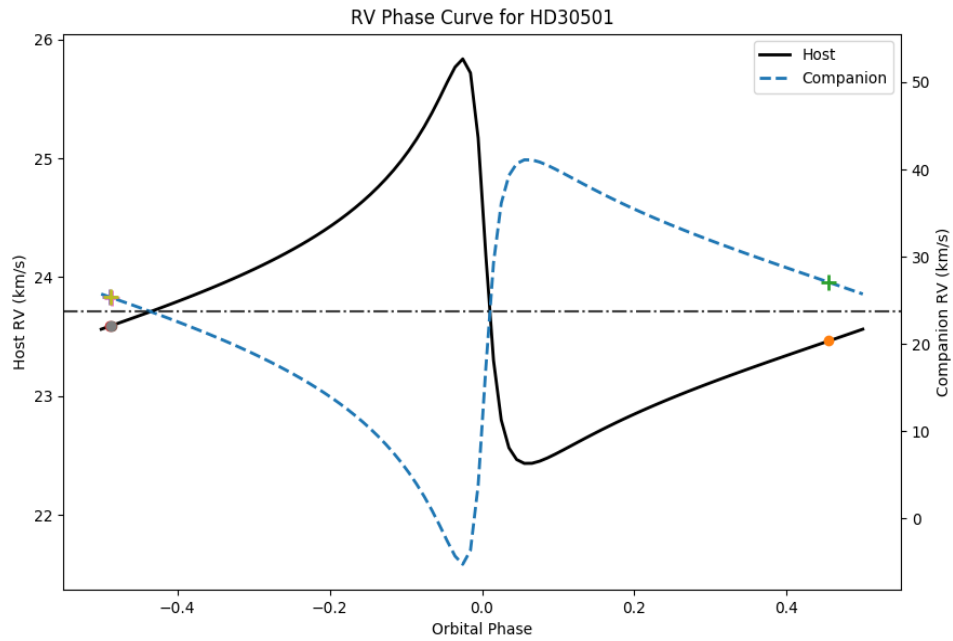
To include the rv of a companion the parameters `m_star` (star mass) and `msini` or `m_true` (companion mass) are required or `k2` the semi-major amplitude of the companion. If `k2` is not provided it is calculated from `k1` and the star and companion masses.

Note: A future version could maybe have the option to obtain parameters from planetary databases such as exoplanet.eu. Although this functionality would be limited to the stars/planets of the databases.

3.2 Usage examples

Simple usage cases:

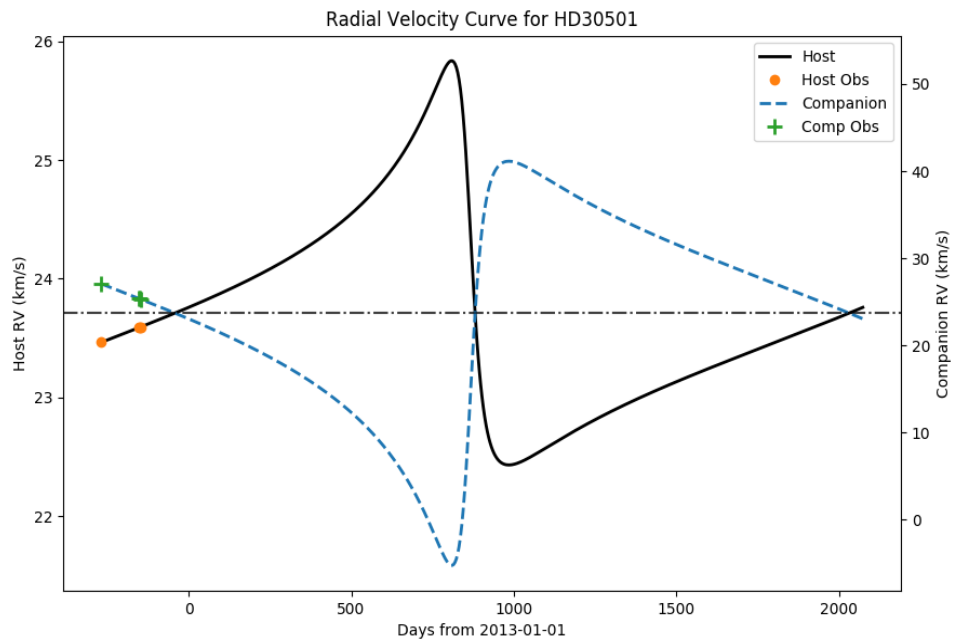
```
python rv.py data/HD30501_params.txt
```



displays the phase curve of HD30501 and its companion.

```
python rv.py data/HD30501_params.txt -l data/HD30501_obs.txt -m time -d 2013-01-01
```

Will create a temporal RV curve, marking the locations of the observations provided in the observation list file.



3.3 Modes

Different functionality can be accessed from the modes flag. `-m`, or `--mode`. The default mode is `phase`.

3.3.1 phase

Produces the RV phase curve of the system between a phase -0.5 and 0.5. If the `k2` parameter is provided or the mass of the host (`m_host`) and companion (`msini` or `m_true`), then the RV for the companion is plotted on the second y-axis.

3.3.2 time

Produces a temporal RV curve of the system over one phase beginning today.

The `-d`, `--date` flag can be used to specify a different reference date for the `time` mode.

If observations dates are provided with the `-o`, or `-l` flags then the curve starts from the earlier of the reference date and the first observation and continues till one period after the reference date.

3.3.3 debug

You can turn on debugging information using the `--debug` flag, e.g.:

```
python rv.py data/HD30501_params.txt -l data/hd30501_obs.txt --debug
```


CHAPTER 4

Contributors

The list of contributors to this project so far are

- [Daniel Andreasen](#)
- [João Faria](#)
- [Jason Neal](#)
- [Andressa Ferreira](#)

You are more than welcome to do pull requests, open issues, give suggestions, etc. on the [github repo](#). Then your name could appear here also.

CHAPTER 5

Indices and tables

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