

Coast Guard

Documentation

Coast Guard is still under development. There are almost certainly bugs, and room for improvement. Some unfinished programs are included. Also, some features, say listed in the help text, may not be implemented. That said, the package is usable, and the information in this document is accurate, at least for now.

Please let me know if you use Coast Guard for any publications.

Patrick Lazarus, Oct. 23, 2013
(plazarus@mpifr-bonn.mpg.de)

Requirements

The Coast Guard package requires python 2.7, psrcat, and the PSRCHIVE python bindings. The install the python bindings follow the instructions at <http://psrchive.sourceforge.net/manuals/python/>

Set-up

To get started with Coast Guard you must create a configurations directory. The location of the base configurations directory must be specified with an environment variable called "COASTGUARD_CFG"

The files in the configurations directory specify default arguments. Different defaults may be specified for different backends, receivers and telescopes. Coast Guard will look for configurations in the following order:

1. `$COASTGUARD_CFG/default.cfg`
2. `$COASTGUARD_CFG/telescopes/<telescope-name>.cfg`
3. `$COASTGUARD_CFG/receivers/<receiver-name>.cfg`
4. `$COASTGUARD_CFG/backends/<backend-name>.cfg`

`<telescope-name>`, `<receiver-name>`, `<backend-name>` should be the lower-case versions of what is reported by vap.

Some basic configurations, not-related to the particular archive being analysed are specified in `$COASTGUARD_CFG/global.cfg`.

Cleaning Archives

The program `clean.py` can be used to remove interference from archives. `clean.py` supports multiple RFI excision algorithms, called "cleaners".

To get a list of cleaners available, and a short description use the `--list-cleaners`

command line option:

```
$ clean.py --list-cleaners
Available Cleaners:
hotbins -- Replace profile bins that are significantly brighter than the
           profile average with white noise.
rcvrstd -- Prune, and tidy the observing band by trimming edges, and
           removing bad
           channels/freq ranges.
surgical -- De-weight profiles that stand out compared to others in the same
           subint/channel using multiple stats.
```

A cleaner algorithm is specified with the “-F” command line option (alternatively the long-name “--cleaner” flag can also be used). Multiple cleaner algorithms may be specified. They will act on the data sequentially. For example,

```
$ clean.py -F hotbins -F surgical
```

will first use the “hotbins” algorithm to clean the archive. After that, the “surgical” algorithm will be applied.

Cleaners have configurable behaviour. To configure a cleaner provide a configuration string with the “-c” command line option. The “-c” options apply to the most recently added cleaner. Multiple “-c” options may be used to configure a single cleaner. They are cumulative. If the same configuration is specified multiple times. The last value specified is used. Alternatively, multiple configurations can be specified in the same configuration string. The configurations should be comma-separated. For example,

```
$ clean.py -F hotbins -c “threshold=7” -c “onpulse=45:63”
```

is equivalent to

```
$ clean.py -F hotbins -c “threshold=7,onpulse=45:63”
```

In both cases the “hotbins” algorithm is configured to use a threshold of 7-sigma, and will ignore the on-pulse region between bins 45 and 63.

To find out what configurations can be specified for each algorithm provide the name of the cleaner to the “-h” command line option:

```
$ clean.py -h hotbins
hotbins -- Replace profile bins that are significantly brighter than the
           profile average with white noise.
Parameters:
    fscrunchfirst -- Determine which bins to removed by looking at
                    frequency scrunched data. Remove the hot bins in all
                    frequency channels.

    Type: bool - The following values are recognised (case
                  insensitive): true, 1, y, yes, false, 0, n, no
    Default: False
```

```
onpulse -- On-pulse regions to be ignored when computing
          profile statistics. A list of 2-tuples is expected.

          Type: list of integer pairs - a list of integer pairs
                <int>:<int>[;<int>:<int>...].
          Default:

threshold -- The threshold (in number of sigmas) for a bin to
             be removed.

          Type: float
          Default: 5
```

The help text describes the cleaner, lists each configuration along with a description, the type, how the configuration string should be specified, and the default value (if applicable).

Other options to the “clean.py” program can be listed by providing the “-h” command line option with no additional argument.

The cleaners framework is modular. If you have an idea for a new cleaning algorithm, it should be relatively straightforward to include it. In this case, please contact me for instructions and/or look at the existing cleaner plugins in the “cleaners” directory.