QReport - A daily GALAH data quality report

May 31, 2015

1 Introduction

QReport is a set of python scripts that produce reports on a nights GALAH survey observing. The reports that it produces at the moment (using the core.py module) are:

- main.txt: a summary of the observing run including the number of the stars and fields observed.
- A V-K vs. V plot of the input catalogue stars observed.
- A map of the fields observed.

Users can add scripts to produce custom reports if desired.

2 Structure

The basic structure of QReport is:

QReport/	Base folder
QReport/QReport.py	Main script
QReport/Rmodules/	Report modules folder
QReport/Rmodules/core.py	Core report module
QReport/Rmodules/common.py	File with globals for report access
QReport/Rmodules/__init__.py	Empty file that allows the import of the report modules
QReport/curr_module/	Current module processing folder

The main script, **QReport.py**, is at the top of the hierarchy and is the one to execute. The main tasks of this script are:

- Centralise the declaration of global variables.
- Load all submodules.
- Verify the location of the GALAH data and custom reports.
- Run all reports.
- Collect and compress the results from all reports.
- Email the results to an elite group of people.

core.py is the script that produces the reports listed in Section 1. It follows the general standards required for custom reports to be recognised (See Section 4).

3 Running QReport

```
python QReport.py [yymmdd]
```

If [yymmdd] is missing it assumes the previous day.

Output: Set of files created by reports zipped in curr_report/yymmdd.tar

4 Adding custom report modules

Custom reports can be added by dropping a python script in the report modules folder (default is **Rmodules/**)

The requirements are:

- Use the variables imported from common.py where applicable. This file is created dynamically by QReport.py.
- Create an array named **ouput_files** containing the list of files to be added to the final report in the desired order.

4.1 Rmodules/common.py

The file common.py is created dynamically by QReport.py. It provides a common reference for all reports to be able to use the information verified by QReport. It contains the following variables:

```
galah_dir - the location of the galah data
d - the date to be analysed (yymmdd)
base_folder - the location of the QReport script
IC_folder - the location of the input catalogue
IC_filename - the file name of the input catalogue
```

All these variables can be imported from any custom report in the following format:

```
import common as const
```

where const can be any alias. For example: To print the GALAH data location from a custom report:

```
print const.galah_dir
```

4.2 Custom report example

```
import common as const
import some module
import some other module

do some analysis that produces myFile.txt

do some pretty plot that produces myPlot.png

output_files = ['myFile.txt', 'myPlot.png']
```

5 Repository, bugs and comments

The code is hosted in Github:
https://github.com/CarlosBacigalupo/QReport

To install and run:

```
git https://github.com/CarlosBacigalupo/QReport.git
cd QReport
python QReport.py [yymmdd]
```

Issues and comments can be added to the issues tracking section of the repository:
https://github.com/CarlosBacigalupo/QReport/issues