QReport - A daily GALAH data quality report

Carlos Bacigalupo

May 20, 2015

Qreport is a set of 2 core python scripts plus a collection of custom reports. 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.

The second script core.py is the actual main report. It follow the general standards require for custom reports to be recognised.

1 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/curr_module/	Current module processing folder
QReport/IC	Input catalogue folder
QReport/IC/galahic_v2.0L.ebf	Input catalogue

2 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.zip

3 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.
- Create an array named **ouput_files** containing the list of files to be added to the final report in the desired order.

3.1 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
```

All this variable 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
```

3.2 Custom report example

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

do some analysis that produces myFile.txt

do some other analysis that produces myPlot.png

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