"offline" ipython notebooks for reproducible research

Pynxton meeting 14/10/2014

Marco Galardini
marco@ebi.ac.uk
@mgalactus

Ipython notebooks (personal?) use cases

- Have a (quick) look at data

- Keep analysis and results in the same place

    - Reduces the risk of forgetting how a certain plot has been done

    - Send batteries included analysis to collaborators as html (nbconvert)

    - Easily re-run an analysis when new data arrives

Ipython notebooks (personal?) use cases

- Have a (quick) look at data

- Keep analysis and results in the same place

  - Reduces the risk of forgetting how a certain plot has been done

  - Send batteries included analysis to collaborators as html (nbconvert)

  - **Easily re-run an analysis when new data arrives**

    - Re-open the notebook

    - Run all cells

    - Close the notebook

    - Re-run ipython nbconvert –to=html notebook.ipynb

Ipython notebooks as automatic reporting systems?

Easily re-run an analysis when new data arrives

- Re-open the notebook
- Run all cells
- Close the notebook
- Re-run ipython nbconvert –to=html notebook.ipynb

This requires to manually operate the notebook in a browser and wait for it to finish, while it could easily fit into a makefile (or a bash script)

Is there a way to run an ipython notebook from the command line?

Running an ipython notebook non-interactively

15 Does anyone know if it is possible to run an ipython notebook non-interactively from the command line and have the resulting .ipynb file saved with the results of the run. If it isn't already possible, how hard would it be to implement with phantomJS, something to turn the kernel on and off, and something to turn the web server on and off?

7 To be more specific, let's assume I already have a notebook original.ipynb and I want to rerun all cells in that notebook and save the results in a new notebook new.ipynb, but do this with one single command on the command line with no requiring interaction either in the browser or to close the kernel or web server, and assuming no kernel or web server is already running.

example command:

```
$ ipython notebook run original.ipynb --output=new.ipynb
```

ipython   ipython-notebook

share | edit | flag   edited Jul 30 '13 at 3:37   asked Jul 28 '13 at 5:38
                                                               davidshinn
                                                               203 ● 2 ● 7

add a comment

start a bounty

3 Answers   active   oldest   **votes**

11 Yes it is possible, and easy, it will (mostly) be in IPython core for 2.0, I would suggest looking at those examples for now.

share | edit | flag   answered Jul 28 '13 at 23:04
                                       Matt
✓   4,825 ● 1 ● 9 ● 27

1 ▲ Thank you for pointing me in the right direction. Since the exact code wasn't posted, I forked one of the gists and added output code so that the following will run a notebook from the commandline just like a desired $ ipnbdocttest.py original.ipynb new.ipynb gist.github.com/davidshinn/6110231 — davidshinn Jul 30 '13 at 5:00 ✎

  Great! this functionality will be added in nbconvert 2.0 with a nicer API, don't hesitate to edit wiki to add a link to your script. — Matt Jul 30 '13 at 8:52

1 @Matt Did this make it into ipython 2? If so, how do I use it? — Simon Byrne May 11 at 9:38

3 ▲ No not in IPython 2, on its way for IPython 3 — Matt May 11 at 12:28

  Such a feature would be really interesting to make the notebooks a tool to perform reproducible analysis (like embedding the command in a makefile, i.e. running the notebook from the command line and then converting it to markdown/rst). Looking forward to it! — mgalardini Oct 1 at 16:02

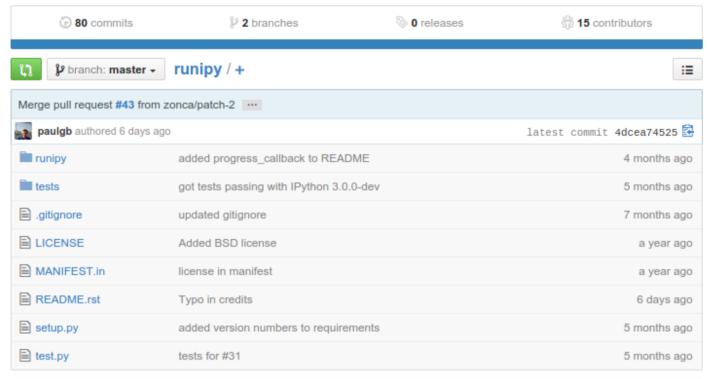 it is now in master. — Matt Oct 2 at 8:22

http://goo.gl/HkmlGD

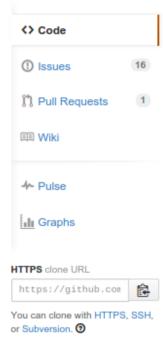- Not yet available in released version

- Apparently is available in the master branch
  - (couldn't find it)

- A little hack of an utility script will do the trick!

Run IPython notebooks as command-line scripts, generate HTML reports

⊙ **80** commits   ⑂ **2** branches   ⬡ **0** releases   ⬡ **15** contributors

⟲   ⑂ branch: **master** ▾   **runipy** / +   ☰

Merge pull request **#43** from zonca/patch-2   ⋯

paulgb authored 6 days ago   latest commit 4dcea74525 ⧉

📁 runipy	added progress_callback to README	4 months ago
📁 tests	got tests passing with IPython 3.0.0-dev	5 months ago
📄 .gitignore	updated gitignore	7 months ago
📄 LICENSE	Added BSD license	a year ago
📄 MANIFEST.in	license in manifest	a year ago
📄 README.rst	Typo in credits	6 days ago
📄 setup.py	added version numbers to requirements	5 months ago
📄 test.py	tests for #31	5 months ago

‹› Code

⊙ **Issues**   16

⑂ **Pull Requests**   1

▦ **Wiki**

✦ **Pulse**

▮▮▮ **Graphs**

HTTPS clone URL

| https://github.com | ⧉ |

You can clone with HTTPS, SSH, or Subversion. ⊙

⬇ **Download ZIP**

▦ **README.rst**

runipy : run IPython as a script

The IPython notebook provides an interactive interface to a Python interpreter.

- **Literate programming**: the IPython notebook is an ideal format for writing "literate" programs, in which the code is part of a larger multi-media document. `runipy` lets you run such programs directly, without first converting to a pure Python script.
- **Report generation**: `runipy` can run the notebook and convert it into HTML in one go, making it an easy way to automate reports.
- **Data pipeline**: if you use IPython notebooks to create a data pipeline, `runipy` lets you automate that pipeline without losing the notebook formatting.

http://goo.gl/4DK95G

Ipython notebooks as automatic reporting systems?

~/$ runipy my_notebook.ipynb tmp.ipynb

~/$ mv tmp.ipynb my_notebook.ipynb

~/$ ipython nbconvert –to=html my_notebook.ipynb

That's it! You can add a couple of rules to a makefile to obtain the updated html (or pdf or markdown, or latex...) every time some data changes or you add some new cell to the notebook.

Happy reproducible research!