Best practices for jupyter notebooks + git + conda MEOM experience

Aurélie Albert, Ocean Next

Julien LeSommer, MEOM@IGE

Mardi Café – Toolkit – 05/05/2020

Main objective for any computational analysis

- ★ Reproducibility is essential
  - Publishing results
  - Future work with different dataset/different method or param
  - Sharing with the community
- ★ Jupyter notebooks allow us to do that !

10 simple rules

OceanNext
Hydrosphere Data & Numerics

- ★ Summary of the workshop “Reproducible Research and Interactive Education – Application of Jupyter Notebooks” 2018 : https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007007

- ★ Illustration with personnal and MEOM shared material

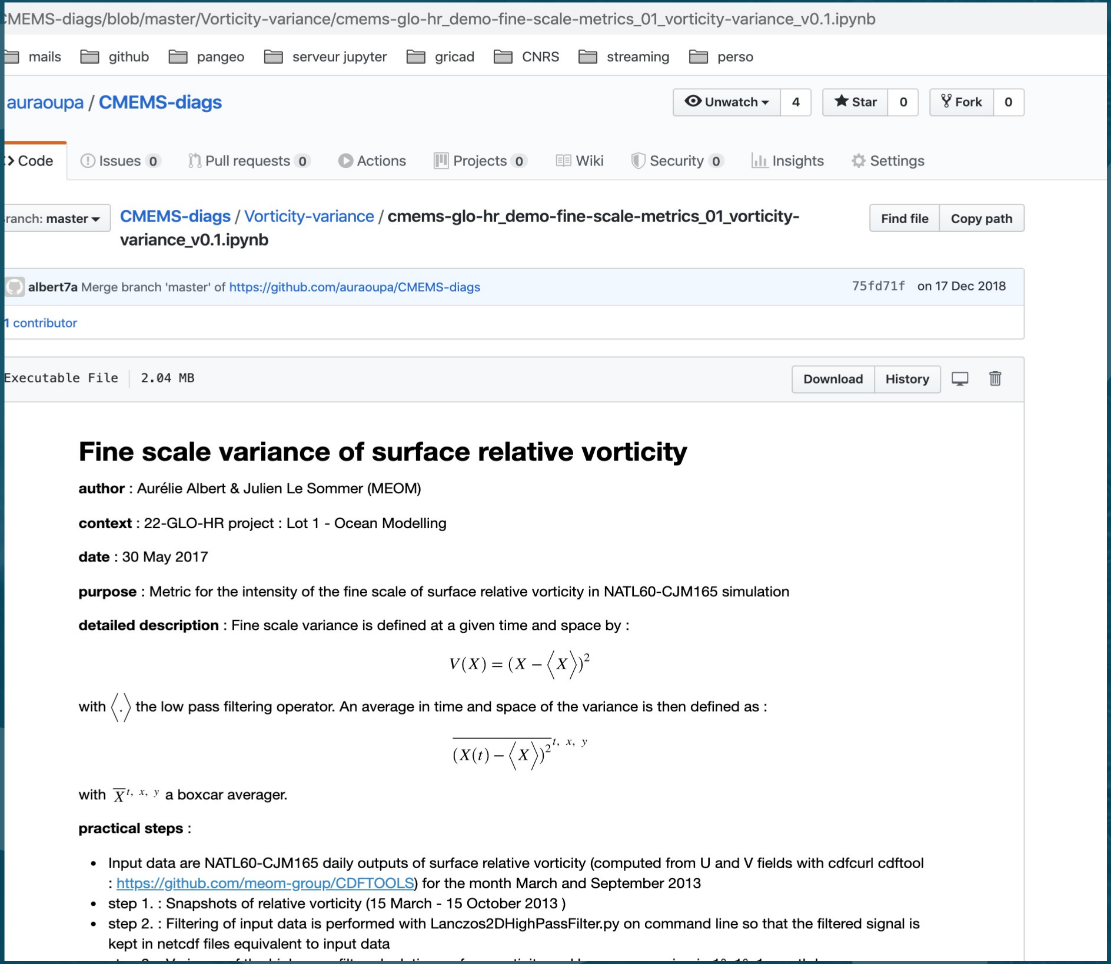
Rule 1 : Tell a story for an audience

OceanNext
Hydrosphere Data & Numerics

- ★ It starts with the title : when-who-what.ipynb
- ★ In a notebook, it is possible to
  - write text with markdown formatting
  - write equations
  - put links
  - import images, etc …

- Idem on github : README.md

Rule 1 : Tell a story for an audience


CMEMS-diags/blob/master/Vorticity-variance/cmems-glo-hr_demo-fine-scale-metrics_01_vorticity-variance_v0.1.ipynb
mails
github
pangeo
serveur jupyter
gricad
CNRS
streaming
perso
auraoupa / CMEMS-diags
Unwatch 4
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Actions
Projects 0
Wiki
Security 0
Insights
Settings
ranch: master
CMEMS-diags / Vorticity-variance / cmems-glo-hr_demo-fine-scale-metrics_01_vorticity-variance_v0.1.ipynb
Find file
Copy path
albert7a Merge branch 'master' of https://github.com/auraoupa/CMEMS-diags
75fd71f on 17 Dec 2018
1 contributor
Executable File 2.04 MB
Download
History

Fine scale variance of surface relative vorticity

author : Aurélie Albert & Julien Le Sommer (MEOM)

context : 22-GLO-HR project : Lot 1 - Ocean Modelling

date : 30 May 2017

purpose : Metric for the intensity of the fine scale of surface relative vorticity in NATL60-CJM165 simulation

detailed description : Fine scale variance is defined at a given time and space by :

$$V(X) = (X - \langle X \rangle)^2$$

with $\langle . \rangle$ the low pass filtering operator. An average in time and space of the variance is then defined as :

$$\overline{(X(t) - \langle X \rangle)^2}^{t,\ x,\ y}$$

with $\overline{X}^{t,\ x,\ y}$ a boxcar averager.

practical steps :

- Input data are NATL60-CJM165 daily outputs of surface relative vorticity (computed from U and V fields with cdfcurl cdftool : https://github.com/meom-group/CDFTOOLS) for the month March and September 2013
- step 1. : Snapshots of relative vorticity (15 March - 15 October 2013)
- step 2. : Filtering of input data is performed with Lanczos2DHighPassFilter.py on command line so that the filtered signal is kept in netcdf files equivalent to input data

OceanNext
Hydrosphere Data & Numerics

Rule 2 : Document the process, not just the results

- ★ it is tempting to erase the code that did not work …
- ★ development notebooks vs deliverable notebooks
- ★ even debug notebooks …
- ★ add 'failed' in the title

Rule 3 : Use cell divisions to make steps clear

- ★ Equilibrium between single-line block and 100-lines block
- ★ Text in mardown describing the following block
- ★ All the imports in one cell, the data imports etc ...

Rule 4 : Modularize code

- ★ It is tempting to copy/paste block of code changing one parameter, best use functions
- ★ If functions are used in different notebooks, make a module/package/library

Rule 5 : Record dependencies

- ★ Crucial for reproducibility (sometimes years after …)
- ★ Worst nightmare of any python user !
- ★ Explicitely print the version of the packages : watermark
- ★ Package manager : conda, pip
- ★ Conda on hpc not always possible : conda-pack

Rule 6 : Use version control

GitHub

GitLab

★ For any code, git allows you to

- back-up your work online
- keep track of changes
- deploy your code on different machines
- work in collaboration

Git commands

- ★ Retrieve a repo
  - git clone https://github.com/auraoupa/repo.git
  - git pull to refresh
- ★ Create a depot from a local dir
  - git init
  - git remote add origin https://github.com/auraoupa/repo.git
- ★ Add content
  - git add file/*/.
  - git commit -m 'comment'
  - git push

Git tips

- ★ .gitignore = list of files not tracked (*nc, slurm*)
- ★ Always git pull first !!!!
- ★ When conflict, vi the file and look for <<<
- ★ Participate on someone's code
  - git clone
  - git branch mycontrib
  - git checkout mycontrib
  - modify
  - git add, commit, push
  - on github, click on pull request

Rule 7 : Build a pipeline/workflow

★ When you're happy with your analysis, make a clean version

- Verify it holds when restarting kernel and rerunning all
- Key variable declaration at the beginning
- Transform your notebook in a script with parameter : papermill
- Test and continuous integration

- From raw data to scientific result
  -

Rule 8 : Share and explain your data

- Ideally entire data is available alongside with notebook
- Describe the upstream process to produce it
- Intermediate small dataset on hosting services (figshare, zenodo)

Rule 9 : Design your notebooks to be read, run and explored

- ★ To be read
  - renders on github (be careful on gitlab), html version, nbviewer
- ★ To be run
  - configuration/dependencies file
  - demo in a binder
  - deployment in a container, docker
- ★ To be explored
  - Simple change in the notebook
  - ipy-widget

Rule 10 : Advocate for open research

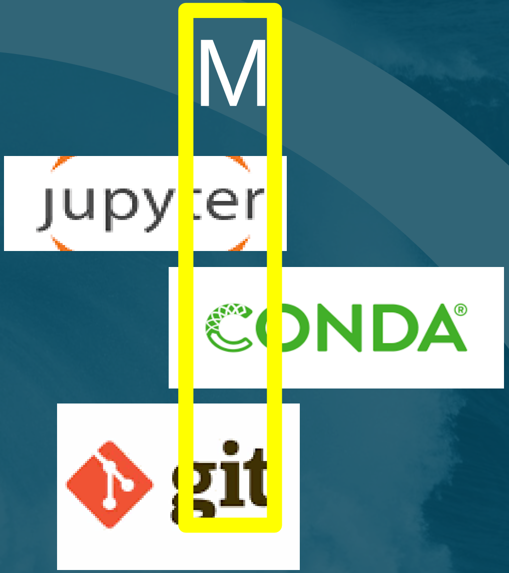
- ★ What I'm doing now !
- ★ Be an example
- ★ Teach your students, convince your co-workers

MEOM experience

Julien LS's advice to his students, 3 types of notebooks :

- 'Lab' notebook : ~development, daily (messy), archived on github
- 'Synthesis' notebook : narration with material from lab notebooks, very illustrative (plots and equations), converted in html for exchanges
- 'Diffusion' notebook : alongside with article or report, reproduce plots or analysis, distributed on github with zenodo tag

Also in the team : teaching material by Emmanuel Cosme, demonstrations on https://github.com/meom-group/tutos



de votre attention ...