## General

* Thank you again very much for organizing such a great workshop, I really enjoyed it and learned a lot!

## Git-intro

* Great overview!
* Consider using 2 screens?
* Good overview of basic concepts + working with github
* Everything ok
* Good way to cover all the basics
* The morning session on git felt very unprepared. Especially the part on branch merging was confusing
* It’s great that we got to test all the weird scenarios we wanted to try
* Maybe it would be good to rehearse better the tutorial and proofread/test all the commands beforehand
* Explanations were great
* Too many tools should have been pre-installed
* Overall quite good
* A bit too fast paced when dealing with advanced topic. Can have more exercises to practice together with instructor
* It was good, thanks, but...
* … in some part it was fast and I lose them
* Good description about how commits are nodes in a tree, with only parent refs. This “extra” description really helps
* Maybe be more extra explicit while doing merge conflicts etc; describe how the tree works \*first\* a bit more in details
* Excellent example. It made git very easy to grasp and try your hand at
* The first introductory block for git felt a bit slow, even for beginners

## Jupyter

* It was very useful and interesting, thanks
* First time using it. A bit overwhelming but good introduction
* A bit too fast paced for beginners
* Very good demo of how jupyter can be used for interactive documentation
* Not clear how much complex code it can handle, in terms of speed etc
* Good overview of general use
* Improved documentation of importance of package for widgets
* Very nice and instructive presentation and exercises
* Rushed over a lot of parts. Rather less material and move detailed
* Bra med mest interaktiva övningar, bra urval
* Kanske bra att fundera över hur gäst wifi mm ska fungera i förväg
* Good material and examples
* Lack of “behind the scene”, where is the file stored etc
* The jupyter format was perfect for following along and testing while you presented it. Well made introductory notebook!
* It would have been great with some walkthrough/intro on suggested approaches to organize your work into notebooks
* Good content and clear examples
* It was a bit rushed at some points, while there were long dead intervals doing the (relatively simple) exercises
* Good content, many exercises
* Too many extensions. Why offer R, octave and julia all at the same time?
* Good session! I am now motivated to use this. Good balance between information and hands-on
* Good structure: as language agnostic it can be with the help of little magic… :-)
* More content ex. of dependency handling; setting up venv via jupyter, etc (?)
* Hands-on exercises
* Lots of material in short time

## Git-collaborative

* Good material and coverage
* The local mirror example is maybe confusing
* Good clear picture on the forking concept - good demarcation Git/GitHub-specific (et al) feature
* It was detailed interactively on board; but consider discussing more in lecture “when to fork and not to fork”, and collaborative hierachical layouts (branches or forks, code review etc)
* Great material!
* Too little time
* Nice overview. Good exercises. Sometimes time is limited to do exercise.

## Git archeology

* Nice to see
* Quite complicated
* Nice overview. Good exercises
* Sometimes time is limited to do exercise
* The whole thing feels very thorough. The exercises make you feel like you are really progressing and learning the relevant skills/features
* Some questions (although kind of interesting) spawn large tangents, which slows down the lecture considerably. Maybe tell some quicker answer and then offer to give more detailed answers after class.

## Automated testing

* Great!
* Good explanation why
* The exercise with Travis was maybe too large
* Good with well prepared examples since testing in itself took some time; enabled the gist of it
* Maybe mention some on testing implementation best practices; i.e. when in your workflow is it good to write test w.r.t efficiency and “redundancy” -> from your experience
* Very advanced! (positive)
* I learned many things and also I didn’t learn many things! But it’s one of the best and useful workshops which I participated in
* Using travis CI and coveralls seems really useful
* The example tests seemed too artificial
* Good introduction for beginners
* Nice structured exercises
* Nice mixture of theory and practice
* Everything very well explained
* Good explanations of how the different tools work, you covered just what is needed to get started and didn’t bog the lecture down with all the different extra features of details available
* It was not really clear beforehand that the accounts were needed. It was good that you reminded us yesterday!
* Great that we talked about Travis CI!
* There could have been more examples and exercise on how to write test functions

## Documentation

* Good selection of material to cover
* Maybe could discuss some alternatives to sphinx?
* Great introduction to sphinx!
* A more clear overview of RST/MD would have been nice
* Good mix of instructions and code-along
* Very informative, interactive and easy to follow with examples
* Very instructive presentation and examples
* Would be nice to maybe just quickly show the alternative doc tools
* very good overview of existing tools and practical exercise
* a bit quick at some stages
* Clear explanation of how Sphinx works
* Explain better how to integrate docs in the git repo where your code is (branching etc.)
* Useful tools digestible presented, markdown & rst simplicity/syntax
* Would be interesting to see how jupyter protocols can be converted into gh-pages/RTD framework; as we are scientists and will write protocols more than we write code docs.

## CMake

* very interesting overview
* lots of new material
* Very much a useful subject! A lot of very good information
* You need to motivate more clearly why it is not realistic to create and maintain large Makefiles.

## Modular code development session

* Good general topic/coverage
* Too big groups for useful group discussion
* Good discussion assignment
* Too large focus on pure functions. Not every function has to (or can) be pure, and that is not a problem.
* Great topic + explanation of pure/inpure
* Maybe make room for some discussion on language spec. ways of doing it; ex. short example for Python/C++/Fortran and how they relate abstractly
* Good slides
* Maybe too many slides? Examples would have been maybe a good idea.