## General

* Git &Jupyter are OK overall
* Good help from the guys
* Sticker system is very helpful
* Seija speaks fast, but OK
* I have no constructive critique...
* Tell about cases where we can lose our dat and how to avoid it
* The speed was too fast. The tasks were not difficult, but needed more time to process the lecture (Comment: This is critique to a specific lecture, but says not which one)
* In between the two Git sessions, hard to pay attention
* Practical exercise about rebasing would have been useful
* Very well organized, even though the schedule fluctuated a bit
* Although I’m not a software developer, the topics were really interesting and potentiall useful
* The assistants were great!
* Overall very interesting workshop!
* A very nice course, very useful.
* It would be useful to add something about versioning of your own project (X.Y.Z). In Git and how to do this using branches.
* Git should have been one of the first things I learned when doing research. I hope SCIENCE-IT does more of these workshops that are useful for keeping one’s sanity.

## Introduction to version control

* It was very useful. Git is not as magic as it used to be.
* Well structured, speed good, good response to questions
* Very good intro to git
* A nice session
* Good collective and individual support and assistance
* Part 2: Interactive, handson
* Pacing and information density was excellent. I did not fall asleep (which is rare)
* I learned how to use git
* Use the commands during the session
* The lectures on Git were excellent, good visualisations, useful materials, right pace
* Really nice introduction to git, good hands-on exercises.
* The Git workshops is pretty cool. I wish I had learned this very early in my PhD, also the make workshop
* More exercises
* Experienced conflict before the subject was covered
* More practice with resolving conflicts would help - it something we deal with regularly
* Live coding could have had a bit slower pace
* Part 2: Would be nice to stop less frequently. It takes long time to cover material this way
* It is a pity we did not have time for the extra parts, but I will read it at home
* Live coding too fast at times
* I learned many very useful things (especially on git). Thanks a lot!

## Jupyter notebooks

* Using ipython page is very nice
* Interesting introduction and application
* Very useful example notebooks (that were not covered)
* Basics of jupyter were well covered
* I came to the workshop for the Jupyter part. It was really good, although short. Now I also appreciate the possibilities with Git.
* The various possibilities of Jupyter was covered rather nicely considering the limited time
* It was useful to learn about the existence of this kind of program.
* Jupyter session was to fast
* Inefficient time management
* Too much time spent on keyboard shortcuts - concentrate on coding
* A bit too fast towards the end
* It was not useful to learn how to use it. I only cared about what I can do with Jupyter
* Some real-world research example would have been useful.

## Collaboration Distributed Version Control

* Interactive and interesting learning method, group work. Close guidance and assistance

## Reproducible research

* Good motivation to reproducibility
* Very useful
* Make part was an excellent thing
* The container lecture, although brief, was really interesting and possibly very useful
* Sumatra seemed interesting but it would have been nice to have some example or demo of it
* Some real-life research example could have been useful to describe containers
* Docker must be included in the list of software requirements
* The reproducible research workflow was too long and slow. It should be more concise. It ate the time for the next lecture. It needs reconstruction
* The docker example stole time from automatic testing. Too much trivial at the beginning

## IDE

* Good introduction to the basic principle of an IDE. Especially the ‘integration’ was well emphasized.
* Points and topics in this session is very suited for me
* Availability of materials for exercise
* Good idea and seems to be powerful program
* The help was good and efficient
* Very specific to a single software. I have a feeling that lifetime of IDEs is very short. Why consider single IDE, rather than something more stable, like Python Debugger.
* If Python 2.7 is recommended for the course, it should be used consistently
* I needed more time for each exercise
* People should set up the environment BEFORE the workshop in order not to spend time on that
* The IDE session went a bit fast

## Archeology

* Excellent exercise
* Git archeology was excellent and very handy
* The archeology session was very good
* Even more time could have been used on the subject

## Documentation

* Good and easy to follow instructions
* Improve instructions for Windows users
* The documentation lecture was useful and interesting
* but the lecturer watered it down. He should make his presentation more concise. The materials are excellent.
* The documentation session is completely new to me and I am glad to be exposed to it.

## CMake

* The make & cmake was a bit fast. Though, understandable. Maybe a clear & simple explanation of what CMake is would be appreciated :-) (for newbies).
* Really good introduction to CMake.
* Not all commands were explicitly present on course web page.
* CMake was very useful. Good examples, easy to follow. Google test will be useful for my daily work.
* It was a bit too fast.

## Testing

* The Travis testing was excellent. Nothing negative to say.

## Modular code development

* Excellent examples for modular code and pure functions. You should consider a day or a separate course on good C/Fortran coding practices.