QuantNet 2.0 @ GitHub

Daniel Neuhoff

Humboldt-Universität zu Berlin CRC 649

November 2015

Outline

GitHub

Terminology and Workflow

Accessing GitHub

GitHub and QuantNet 2.0

GitHub

Terminology and Workflow
Accessing GitHub

GitHub and QuantNet 2.0

What is GitHub?

- A distributed version control system (Git)
- A collaboration platform (Hub)
- Quasi-standard among software developers:
  42.9% of professional software developers use git in some fashion

Why use GitHub?

- Version control
- Distributed development
- Easy branching and merging
- Integration with many IDEs
- Issue management

GitHub

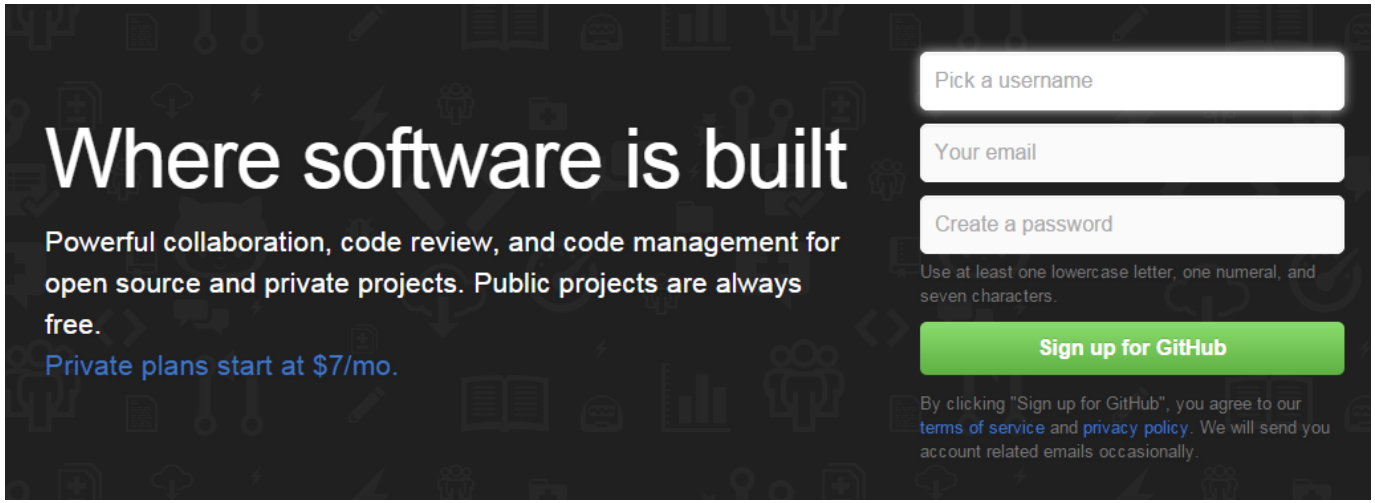
Terminology and Workflow

Accessing GitHub

GitHub and QuantNet 2.0

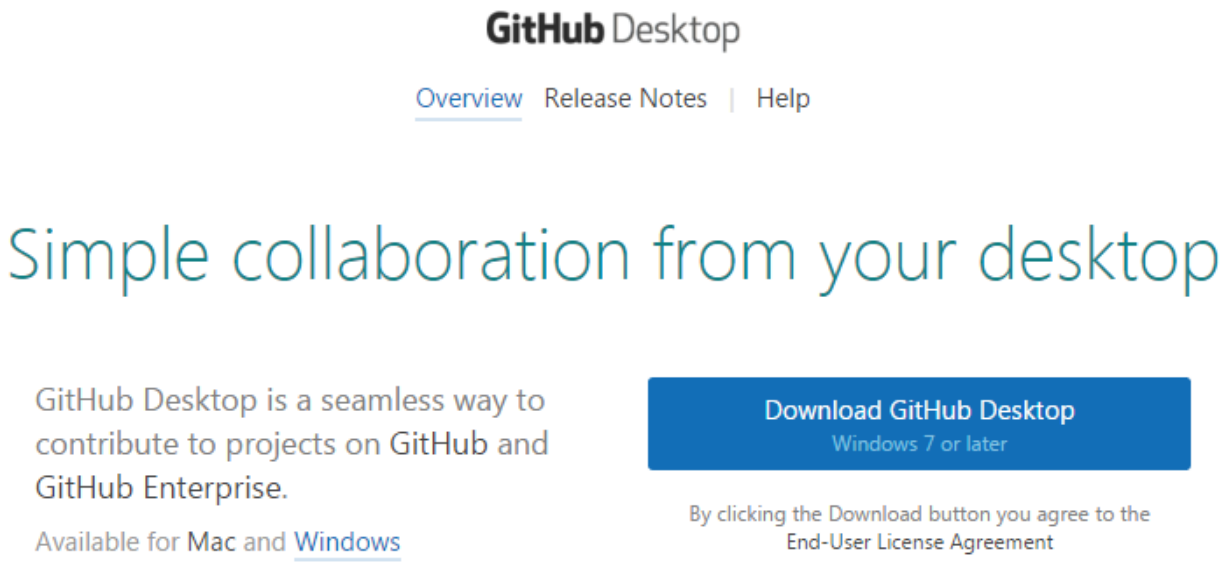
Create GitHub Account

Go to www.github.com to create an account

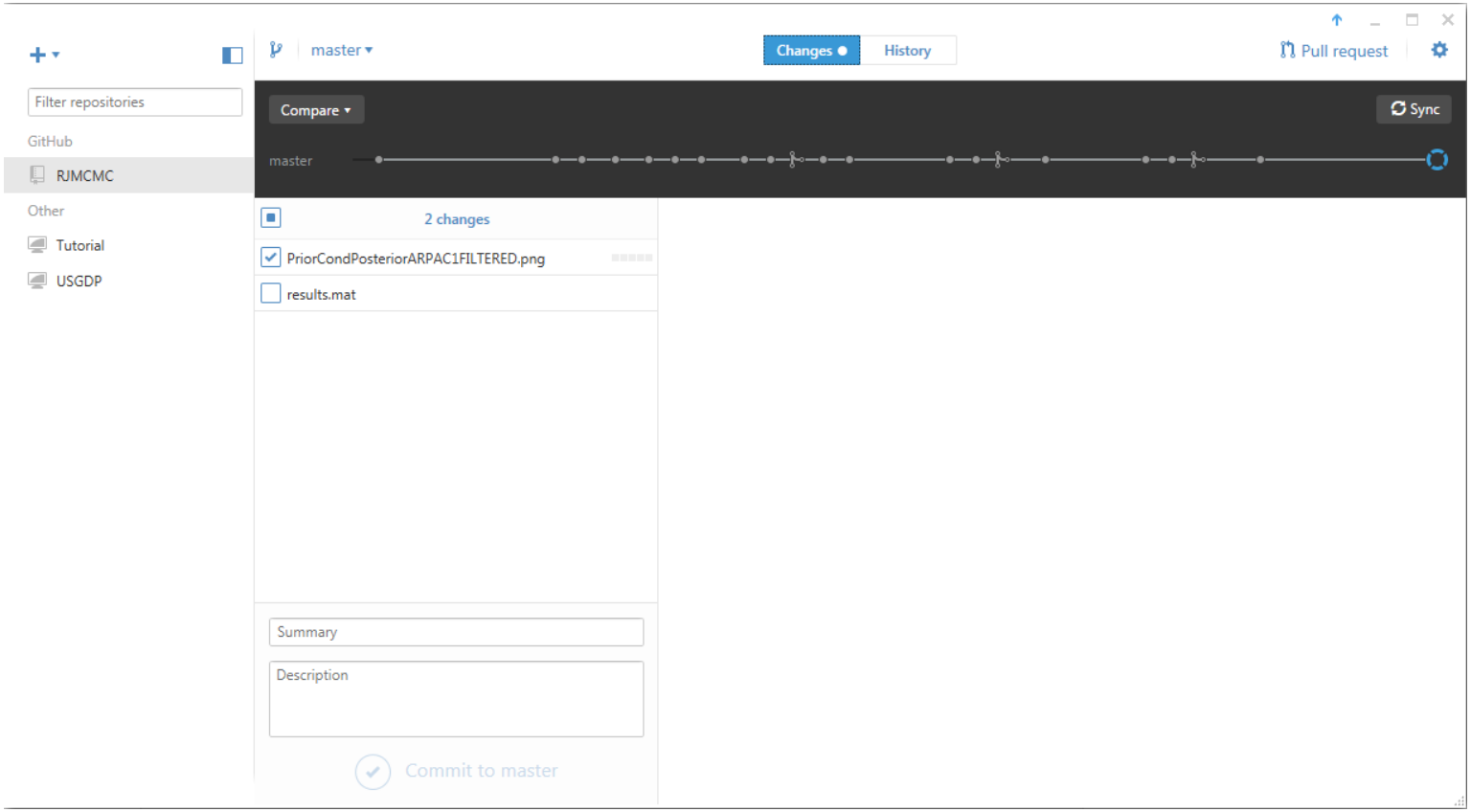


Install GitHub Application

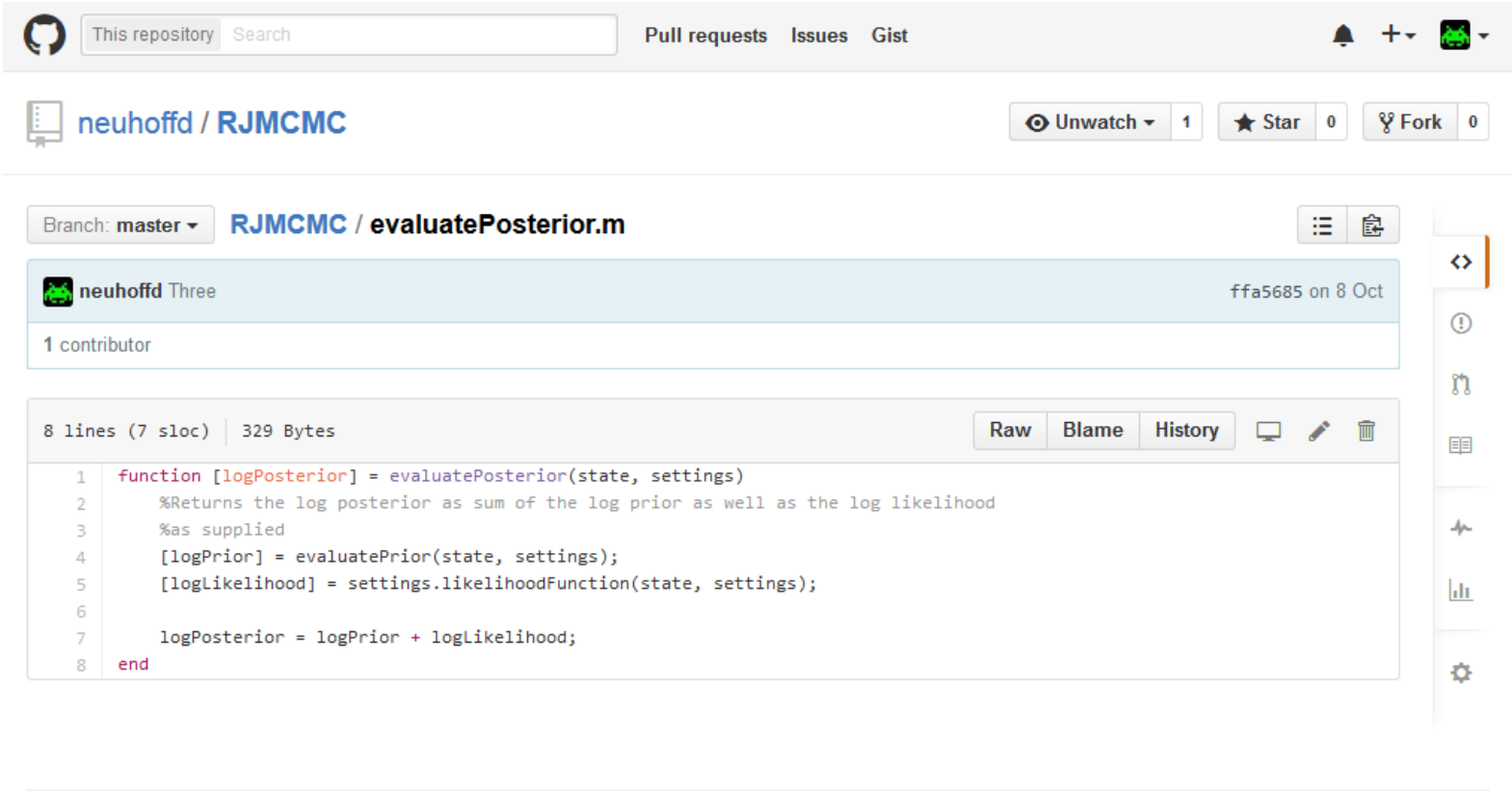
From www.github.com download and install the desktop application:



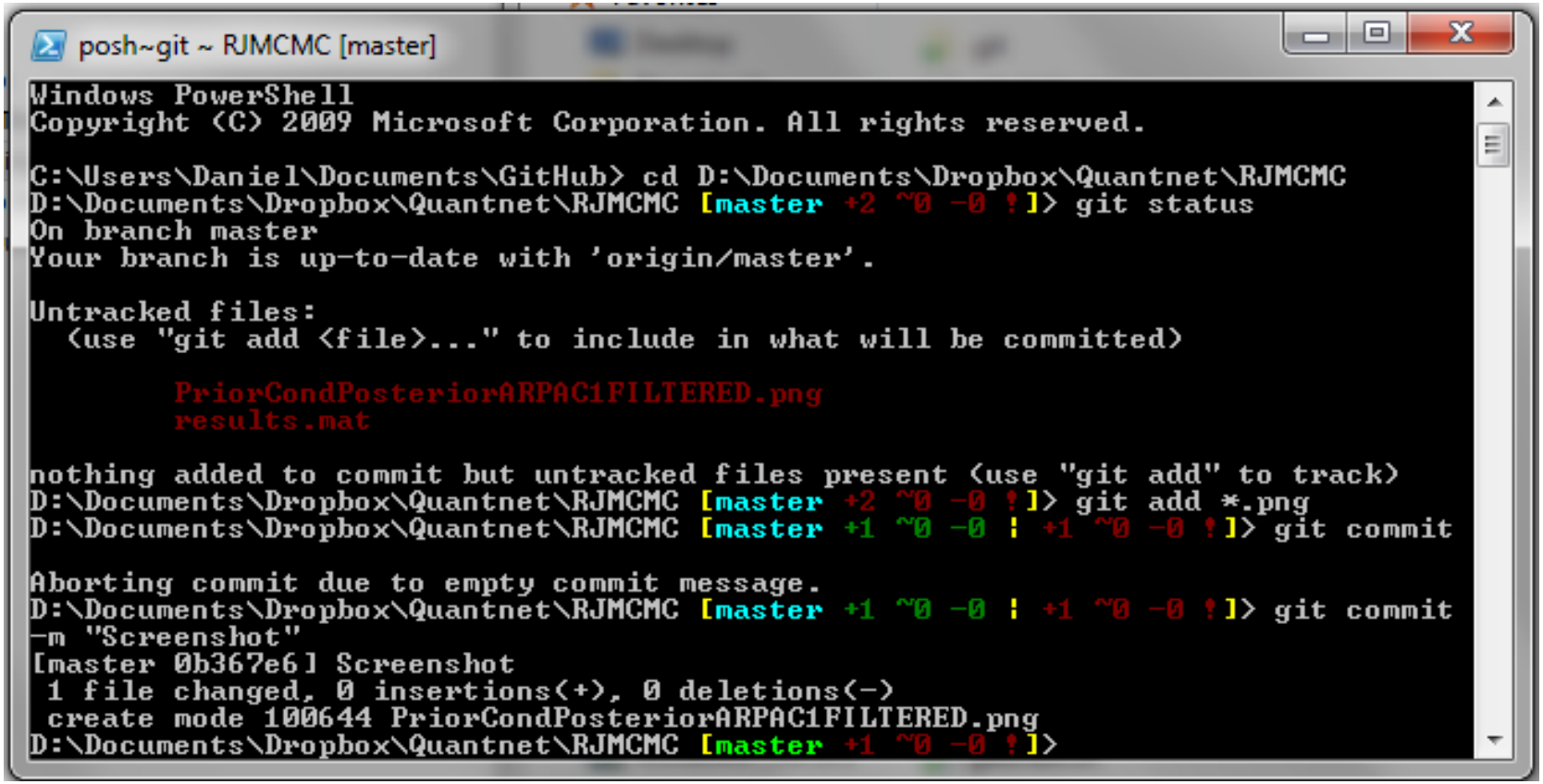
GitHub Desktop App



Web Interface



Command Line



Create Repository

- Most basic element of GitHub
- A project folder containing *all* project files
- Also contains a revision history for each file
- Contains an issue tracker

Three ways to create a repository:

1. Desktop app (recommended)
2. Web interface
3. Command line (advanced)

Develop

Develop your project using GitHub features where sensible:

- Manage issues
- Branch/fork code
- Commit changes
- Use pull requests

Keep the style guide in mind!

Manage Issues

Use GitHub issues to record and discuss

- Ideas
- Bugs
- Enhancements
- Tasks

You get a searchable history of your discussions!

You can neatly organize any discussion with issue classes

Branch Code

Branching allows you to

- work on a copy of the master branch
- to make changes without affecting the whole of the code base

Commit Changes

A commit

- essentially uploads new versions of files
- is tracked, so you have a history of changes available
- can be rolled back

Issue Pull Request

A pull request

- asks your collaborators to consider your changes for integration into the master branch (merge)
- can be issued at any time, also for example to share screenshots
- can be augmented by a pull request message to ask for help or @mention other contributors in order to induce them to comment
- initiate a discussion of the changes you made

Merge Branches

A merge

- integrates your code into the master branch
- preserves a history of your changes by keeping the pull requests (searchable)

Finishing up

- Create Metainfo.txt containing information about your Quantlet
- Create readme.md as user guide (good practice)
- Check your code against style guide (somewhat automated for R code)
- Inform QuantNet team

GitHub

Terminology and Workflow

Accessing GitHub

GitHub and QuantNet 2.0

What I did

1. Create GitHub repository
2. Move code into GitHub repository
3. Develop with an eye on style guidelines
4. Write readme.md
5. Declare running version ready for audit

After the audit is complete, the code is forked to a specific GitHub repository and appears on the QuantNet 2.0 page

What else could one do?

- Collaborate with other scientists around the world using the GitHub workflow
- Fork existing Quantlets in order to improve or extend them (ask the original author!!!!)
- Use Pulse and Graphs to track contributions and progress
- Use Milestones to structure a project

Thank you for your attention!