

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

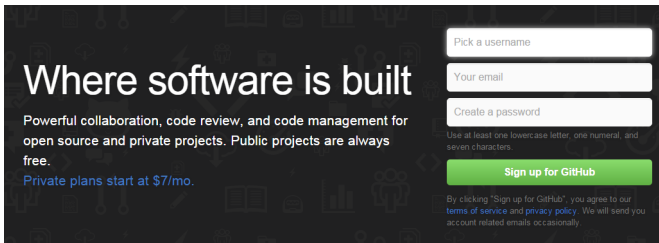
Terminology and Workflow

Accessing GitHub

GitHub and QuantNet 2.0

Create GitHub Account

Go to www.github.com to create an account

A screenshot of the GitHub sign-up page. The background is dark with faint icons. On the left, the text 'Where software is built' is prominent, followed by a description of GitHub's features and pricing. On the right, there are three input fields for 'Pick a username', 'Your email', and 'Create a password', each with a placeholder text. Below these is a green 'Sign up for GitHub' button. At the bottom right, there is a small disclaimer about agreeing to terms of service and privacy policy.

Where software is built

Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free.

Private plans start at \$7/mo.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.

Install GitHub Application

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

GitHub Desktop

[Overview](#) | [Release Notes](#) | [Help](#)

Simple collaboration from your desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise.

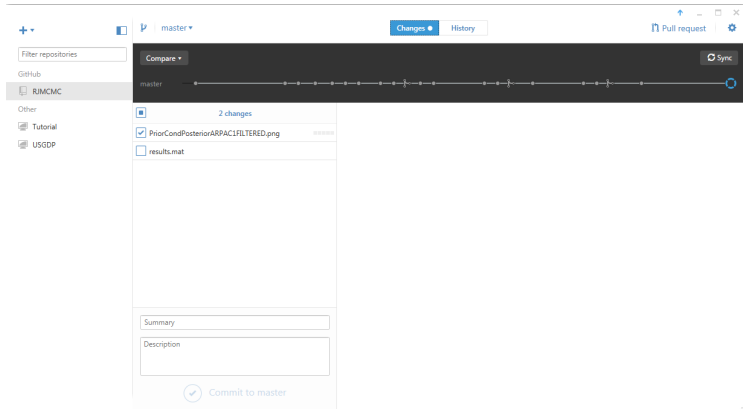
Available for Mac and [Windows](#)

Download GitHub Desktop


Windows 7 or later

By clicking the Download button you agree to the
End-User License Agreement




GitHub Desktop App




Web Interface

 This repository


[Pull requests](#) [Issues](#) [Gist](#)

 **neuhoffd / RJMCMC**




[Unwatch](#) 1 [Star](#) 0 [Fork](#) 0

Branch: master **RJMCMC / evaluatePosterior.m**

 **neuhoffd** Three ffa5685 on 8 Oct

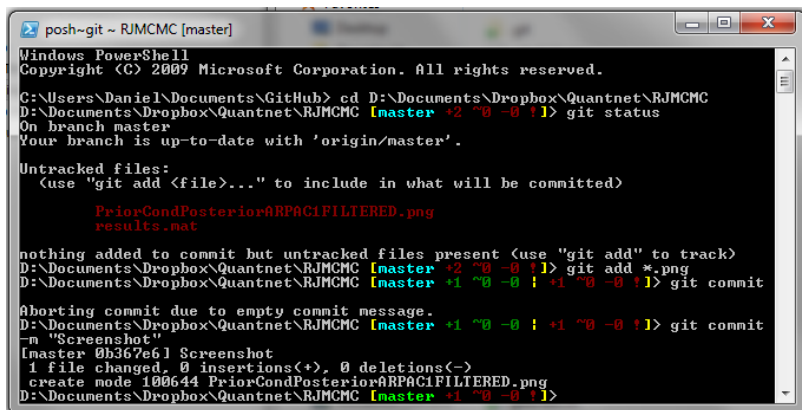
1 contributor

8 lines (7 sloc) | 329 Bytes

[Raw](#) [Blame](#) [History](#)   

```
1 function [logPosterior] = evaluatePosterior(state, settings)
2     %Returns the log posterior as sum of the log prior as well as the log likelihood
3     %as supplied
4     [logPrior] = evaluatePrior(state, settings);
5     [logLikelihood] = settings.likelihoodFunction(state, settings);
6
7     logPosterior = logPrior + logLikelihood;
8 end
```

Command Line



```
posh~git ~ RJMCMC [master]
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Daniel\Documents\GitHub> cd D:\Documents\Dropbox\Quantnet\RJMCMC
D:\Documents\Dropbox\Quantnet\RJMCMC [master +2 ~0 -0 !]> git status
On branch master
Your branch is up-to-date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        PriorCondPosteriorARPA1FILTERED.png
        results.mat

nothing added to commit but untracked files present (use "git add" to track)
D:\Documents\Dropbox\Quantnet\RJMCMC [master +2 ~0 -0 !]> git add *.png
D:\Documents\Dropbox\Quantnet\RJMCMC [master +1 ~0 -0 ! +1 ~0 -0 !]> git commit
Aborting commit due to empty commit message.
D:\Documents\Dropbox\Quantnet\RJMCMC [master +1 ~0 -0 ! +1 ~0 -0 !]> git commit
-m "Screenshot"
[master 0b367e6] Screenshot
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 PriorCondPosteriorARPA1FILTERED.png
D:\Documents\Dropbox\Quantnet\RJMCMC [master +1 ~0 -0 !]>
```

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 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

Advantages

- ▶ QuantNet will be fully integrated with GitHub in the near future!
- ▶ It will be easy for other researchers to find your code
- ▶ Your code will be checked by the audit team!

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!