QuantNet 2.0 @ GitHub

Daniel Neuhoff

Humboldt-Universität zu Berlin CRC 649

November 2015

Outline

Reversible Jump Markov Chain Monte Carlo

QuantNet 2.0

GitHub

Terminology and Workflow
- Accessing GitHub

GitHub and QuantNet 2.0

Reversible Jump Markov Chain Monte Carlo

QuantNet 2.0

GitHub

Terminology and Workflow
- Accessing GitHub

GitHub and QuantNet 2.0

Reversible Jump MCMC

Standard practice for approximation of posterior distributions for model parameters: Metropolis-Hastings samplers

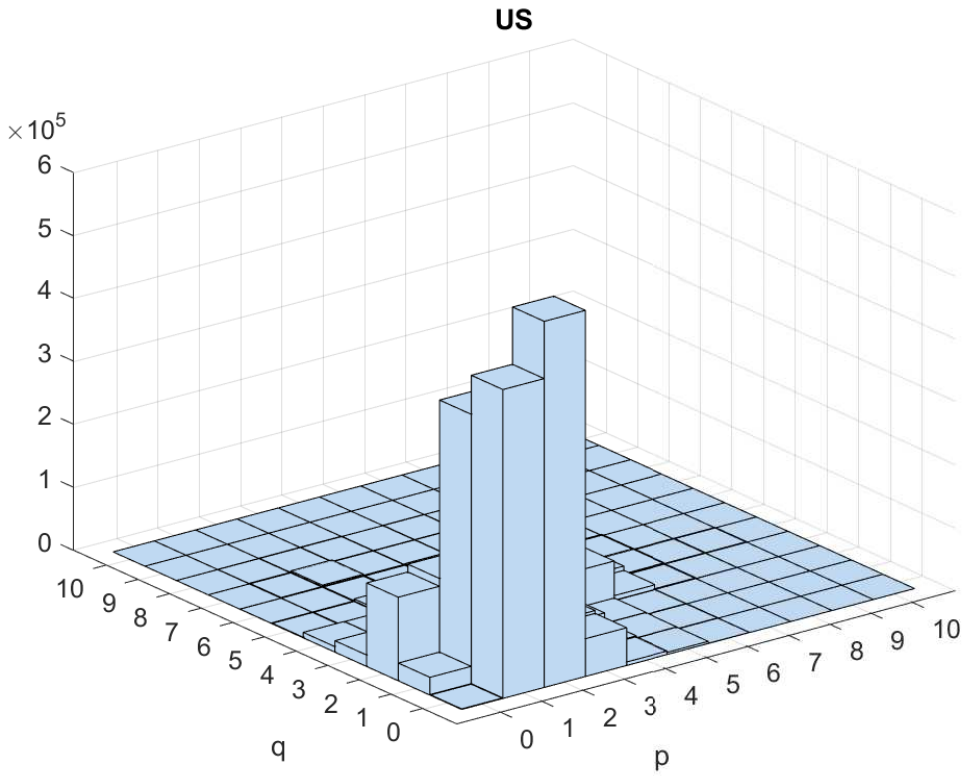
Problem: Want to analyze posterior distribution also spanning model space
$\Rightarrow$ Dimensionality of parameter space varies

Solution: Reversible Jump Markov Chain Monte Carlo

- Generalization of Metropolis-Hastings samplers
- Samples from a joint posterior distribution across different models and their corresponding parameter spaces

Posterior Distribution Across Models

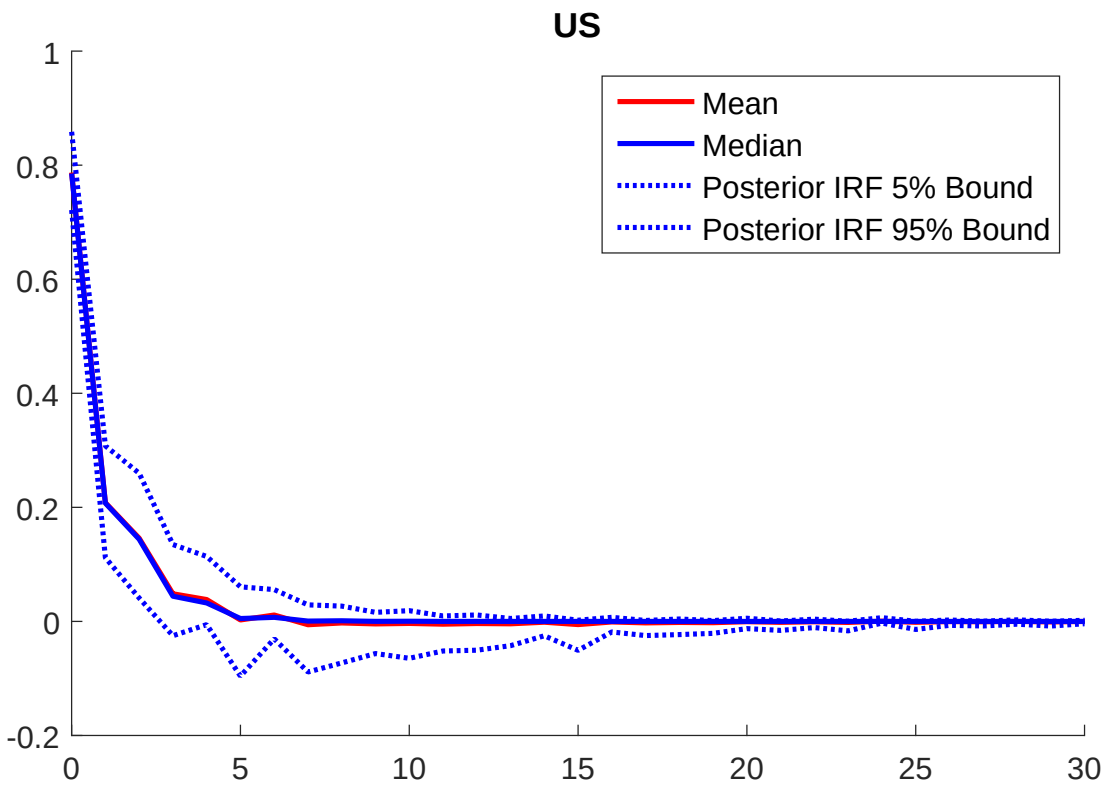
Posterior distribution across ARMA(p,q) models:



⇒ Posterior model probabilities

Posterior Distribution: Impulse Responses

Can analyze posterior distribution for any statistic while accounting for model uncertainty!



Modern Scientific Paradigm

Modern scientific practice:

- Transparency
- Reproducibility

Also: Want to publicize new technologies!

Problem: Need to publish source codes and data!

Reversible Jump Markov Chain Monte Carlo

QuantNet 2.0

GitHub

Terminology and Workflow
Accessing GitHub

GitHub and QuantNet 2.0

The Solution

QuantNet 2.0

1. provides a technology to easily share data and programs
2. provides a platform focused on scientific applications
3. makes technology searchable
4. supports transparency and reproducibility
5. enhances collaboration through GitHub integration

Reversible Jump Markov Chain Monte Carlo

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

Reversible Jump Markov Chain Monte Carlo

QuantNet 2.0

GitHub

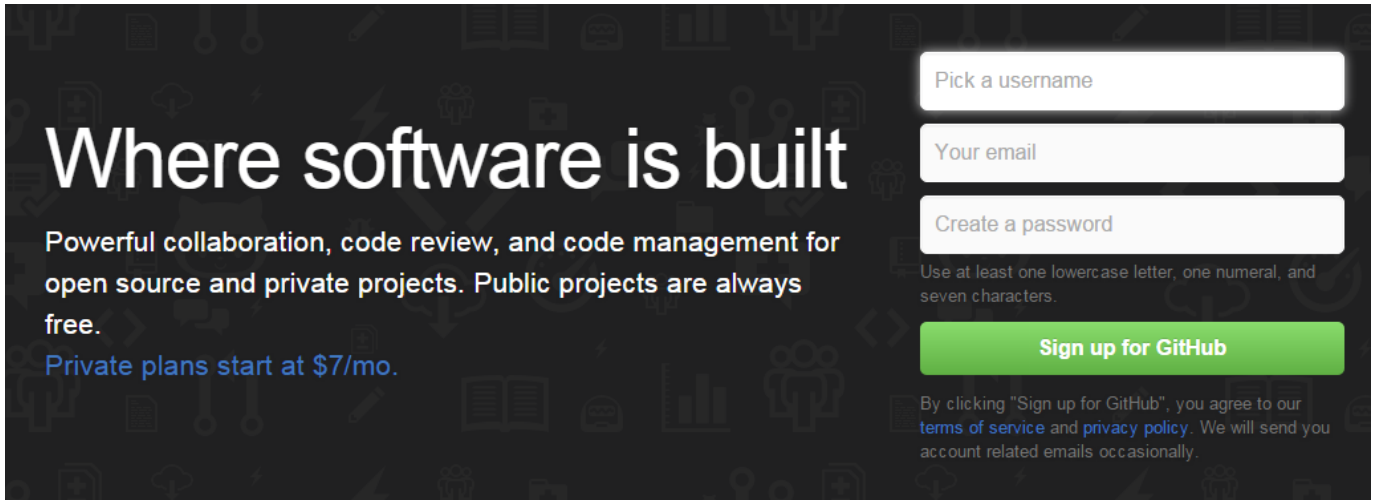
Terminology and Workflow

Accessing GitHub

GitHub and QuantNet 2.0

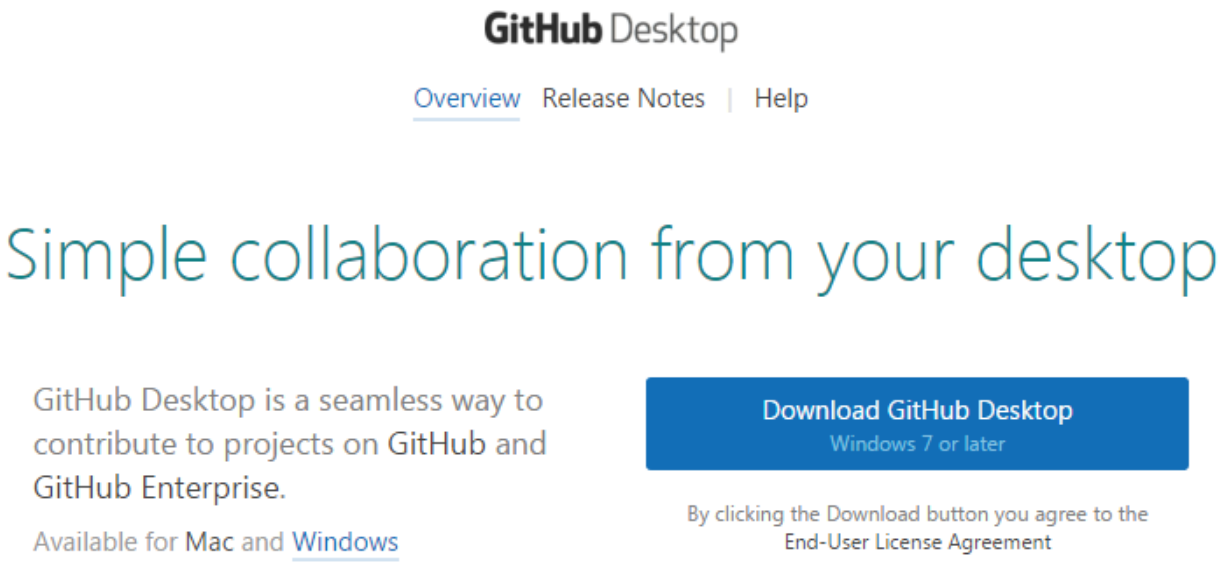
Create GitHub Account

Go to www.github.com to create an account

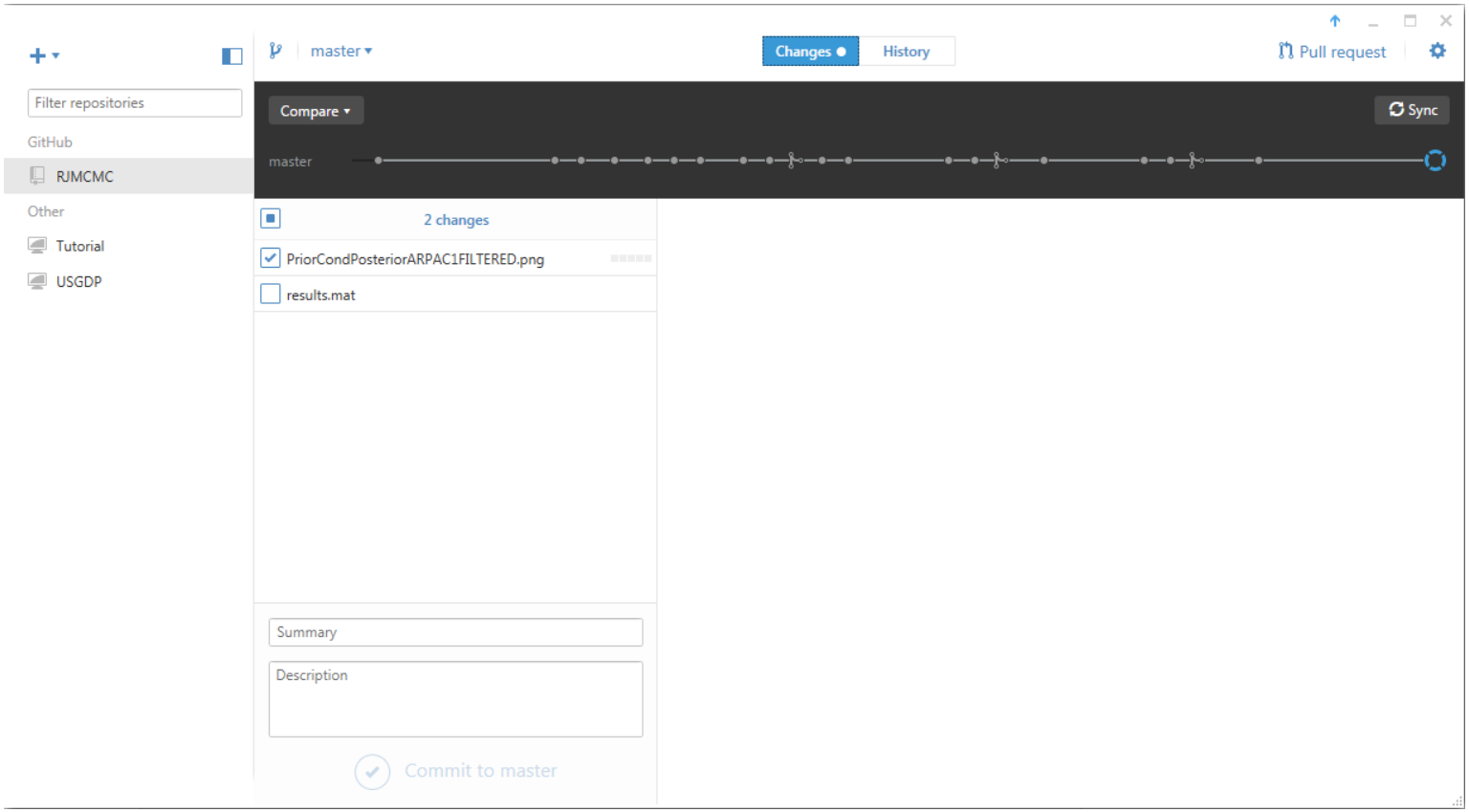


Install GitHub Application

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



GitHub Desktop App



Web Interface

This repository Search 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

```
function [logPosterior] = evaluatePosterior(state, settings)
    %Returns the log posterior as sum of the log prior as well as the log likelihood
    %as supplied
    [logPrior] = evaluatePrior(state, settings);
    [logLikelihood] = settings.likelihoodFunction(state, settings);

    logPosterior = logPrior + logLikelihood;
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)

        PriorCondPosteriorARPAC1FILTERED.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 PriorCondPosteriorARPAC1FILTERED.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

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

Reversible Jump Markov Chain Monte Carlo

QuantNet 2.0

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!