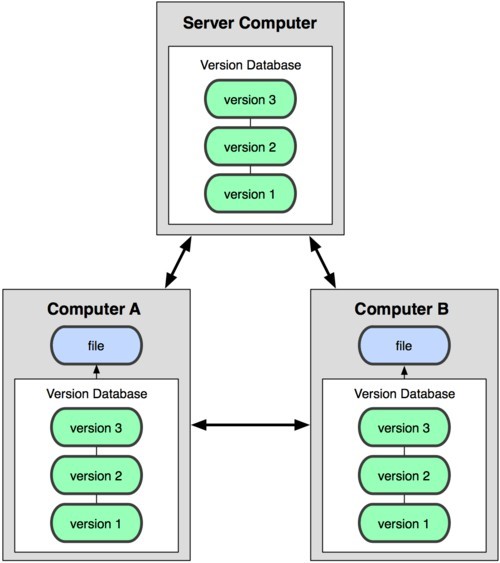
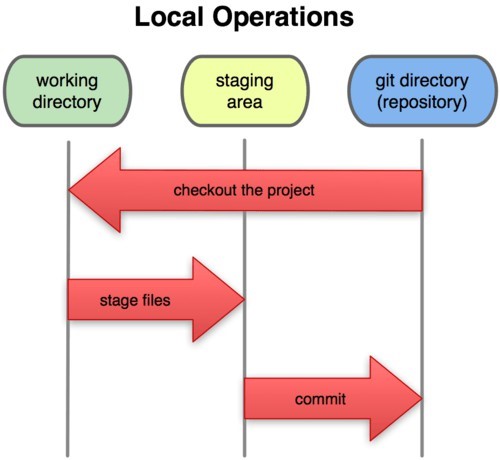
What is "version control"

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later

# Git: the three states

* Commited (stored in local database)
* Modified (file changed but not commited to database)
* Staged (modified file is marked to go into the next commit snapshot)



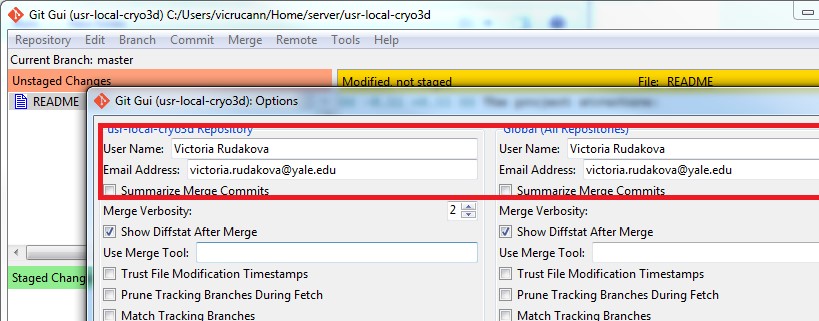
Git installation

* Windows: [http://www.git-scm.com](http://www.git-scm.com/) ● Linux:
* apt-get install git
* yum install git
* Already installed in cygwin

# Git config

● Using Git Bash (command line):

$ git config --global user.name "Name Surname" $ git config --global user.email name.surname@yale.edu ● Using Git GUI:



# Getting git repository

* To clone existing repository from server2:

$ git clone username@172.23.5.77:/usr/local/cryo3d/cryo3d.git

* To start version controlling edited existing (new) files (tracking and commiting to local repository):

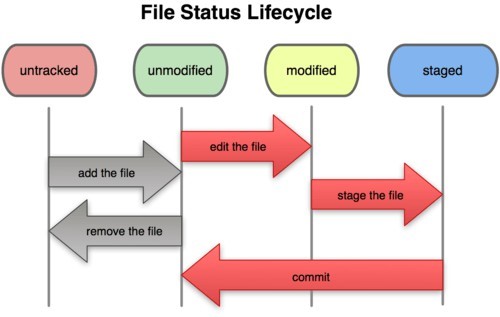
$ git add filename [start tracking new/edited filename]

$ git add . (git add -A) [start tracking all changed/new files]

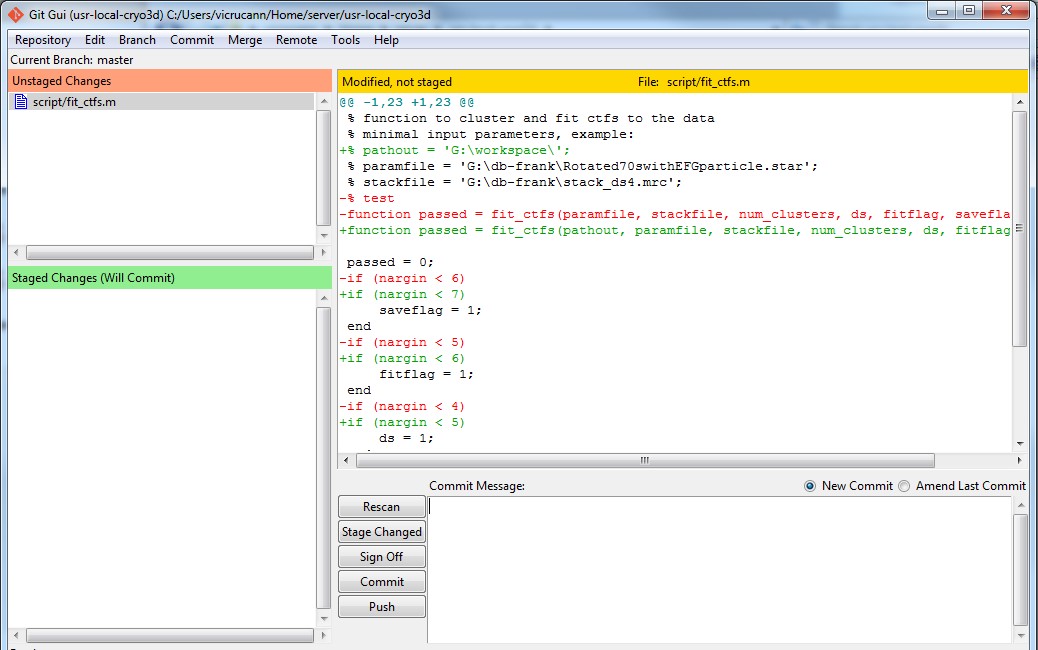
$ git commit -m 'Commit message: what changes were introduced'

[save changes to the local repository]

# Recording changes to the repository

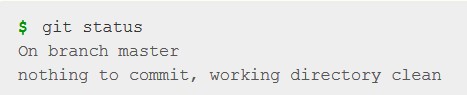


# Recording changes to the repository (GUI version)



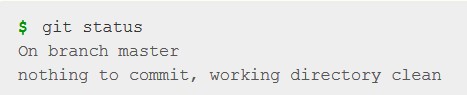
# repository: status

$ git status [Check status of your project]

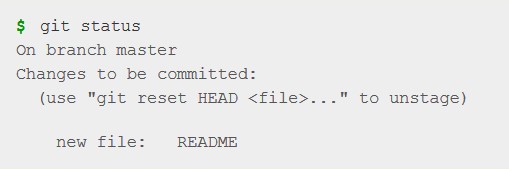


# repository: status

$ git status [Check status of your project]



# repository: tracking your files

$ git add [Begin tracking a new file

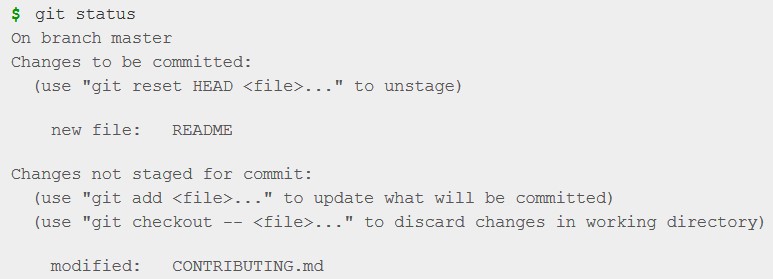
(directory)]



repository: staging modified files

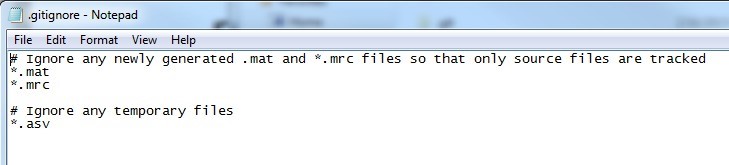
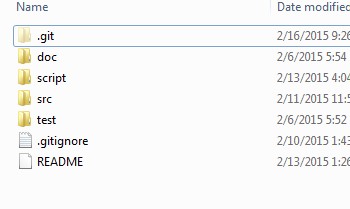
$ git add [Stage the file, add this

content to the next commit]



# repository: ignoring files

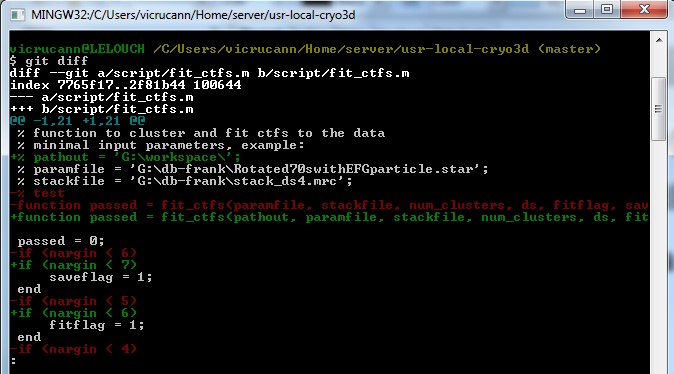
* If we do not want to track automatically generated files (e.g. Log files, build files etc)



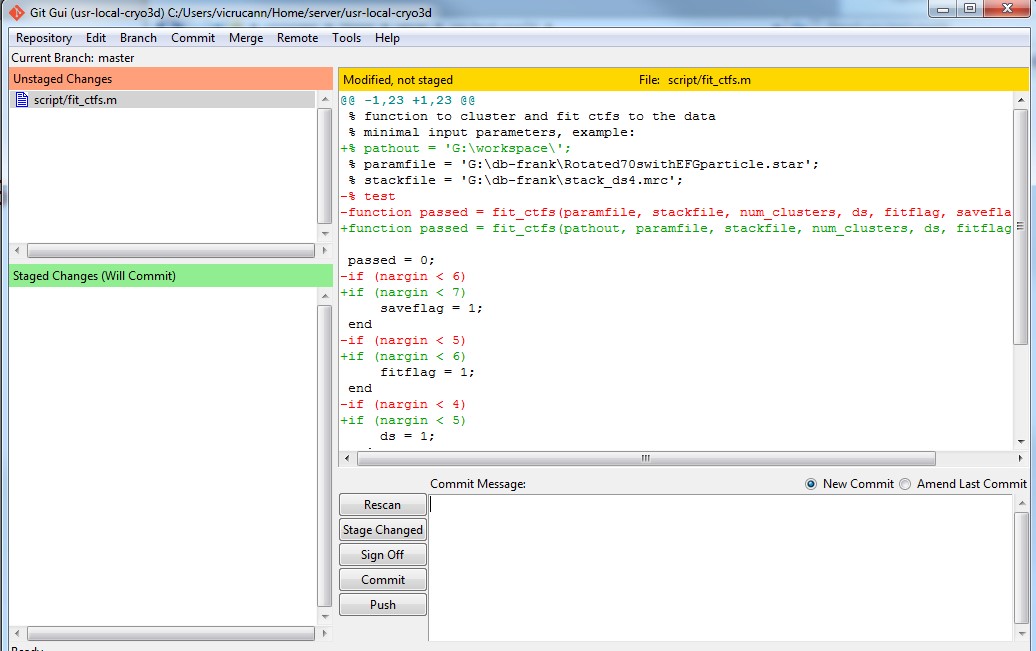
* .gitignore file

# Viewing staged and unstaged changes

$ git diff [what changed but not yet staged]

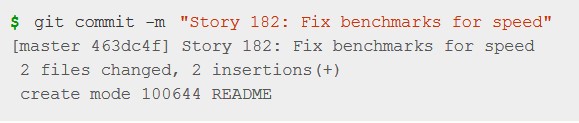


# Viewing staged and unstaged changes - GUI



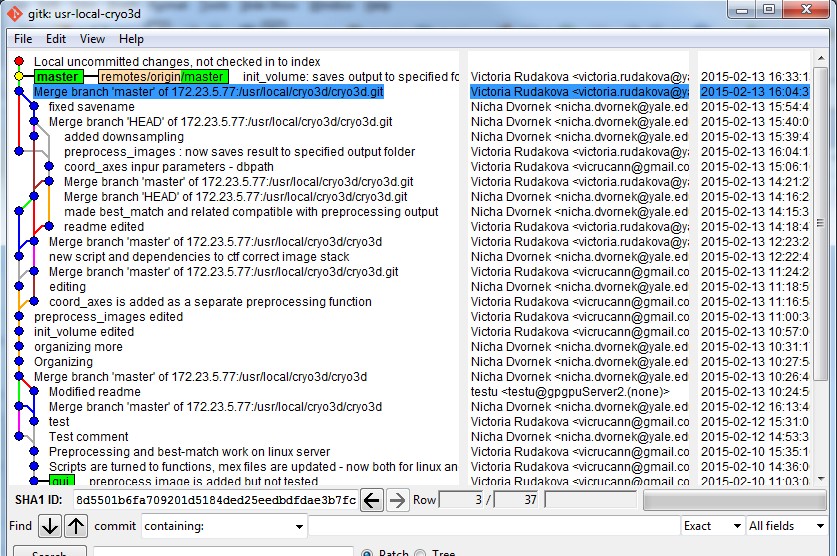
# Recording changes to the repository: commiting your changes

$ git commit [commit your changes to the local repository]



Viewing the commit history

$ git log



# Working with remotes

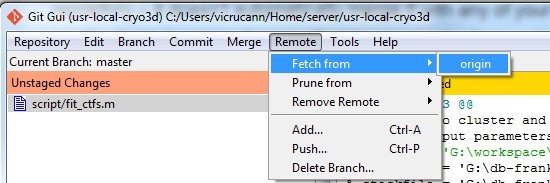
$ git fetch [fetch all the info you don't have from remote repository, no automatical merging]

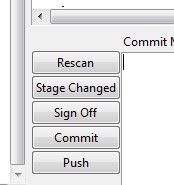
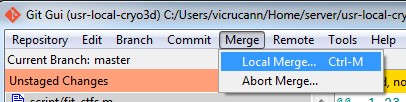
$ git merge [automatically merge data from remote with the your repository data]

$ git pull [fetch and merge automatically] pull = fetch + merge

$ git push origin master [push your version to the server]

# Working with remotes - GUI





# Pushing to already changed remote

* Git won't allow to push to the remote which is ahead of your version: first need to fetch

