

Basic Installation

Download from <https://www.enthought.com/products/canopy> either the free or academic version of Enthought Canopy.

Make sure you have the latest version.

- Go to <https://github.com/samedling/MCSAS> and click Download ZIP in the lower right.
- Or a current zipfile may be at <https://drive.google.com/open?id=0B8EbmzXGZtaZV3MxUlhSUjczQm8>
- Or run `git clone https://github.com/samedling/MCSAS.git` to download the entire repository.

If possible, run `make` to compile the fortran code to achieve 2-100x speedup (depending on number of cores). Or, try copying the `fastmath-OS_CPU.so` file to `fastmath.so`.

Run `python newgui.py` on the command line.

Linux Installation

Default is `gfortran`; to use `ifort` simply edit the makefile.

Tested using `gfortran` on Ubuntu14.10, `ifort` on a rajjin login node (no OpenMP), and both on CentOS.

OS X Fortran Installation

Apple doesn't provide a recent version of `gfortran`, but you can download one from <http://hpc.sourceforge.net>

If you also don't have Xcode Tools, follow the directions at <https://wiki.helsinki.fi/display/HUGG/Installing+the+GNU+compilers+on+Mac+OS+X>

1. Install XCode Tools from the App Store.
2. Install the Command Line Tools by running `xcode-select --install`
3. Download the latest stable `gfortran` version from <http://hpc.sourceforge.net>

Tested on OS X 10.10 "Yosemite".

Some Troubleshooting

OS X/EPD/Tkinter: Make sure you have Canopy. EPD might tell you it's updated everything, but it's still not the same as Canopy.

Centos/F2PY: Make sure you are using the version of F2PY which matches your version of Python (2.7+), otherwise just loading the Fortran module causes a Segmentation Fault.

OS X (and Windows?): You need to close all the old plots before you can run things again. Otherwise, it's otherwise unresponsive for some reason.

PIL: On older systems you may need to manually remove PIL and install Pillow (`sudo pip uninstall PIL` and `sudo pip install Pillow`); newer systems should simply come with Pillow. Otherwise Image won't be able to read the funny SAXS TIF files.

Adding Models

First, make sure you have the most recent version.

To add a Monte Carlo model:

1. Open `density_formula.py`. At the very bottom of the file, create another "elif:" block like the ones above it. Remember the parameters your function uses. Remember the number you assigned. Save and close the file.
2. Open `newgui.py` and go to the line defining `MC_num_and_name`. After the last number (currently around line 80), add a line like the ones above it using the number from step 1.
3. Go to the beginning of the `Fit_Parameter` class definition In the elif block (currently around line 450), add a pair of lines with the number and the parameters from step 1. Save and close the file.

To add an analytic model:

1. Open `analytic_formula.py`. Near the bottom of the file but above where `theory_csv` is defined, create another "elif:" block like the ones above it. Remember the number you assigned. Save and close the file.
2. Open `newgui.py` and go to the line defining `Analytic_options`. After the last number (currently around line 90), add a line like the ones above it using the number from step 1. Save and close the file.

Finally (after some testing), use git to add/commit/push or e-mail a collaborator to do it for you.

Git

Git Setup

Make sure you have git and a github account.

E-mail `scott.medling@anu.edu.au` with your username so I can add you as a collaborator.

Optionally, to configure your local git, run

```
git config --global user.name "<Name>"
git config --global user.email "<E-mail Address>"
git config --global color.ui auto          #Improves readability.
git config --global core.editor vim        #If you like vim.
```

To download the respository, run

```
git clone https://github.com/samedling/MCSAS.git
```

Basic Git Use

Every time you edit

```
git pull origin master    #Download latest changes.  Run every time you start.
```

After editing a file, or number of files

```
git add <filename>        #Adds a filename
git add <filename2>       #Adds another filename, etc.

git status                #Tells you which files have been changed/added.
```

```
git commit -m "<Insert short message here.>" #Saves added changes locally.  
git push -u origin master #Uploads committed changes to respository.
```

Other Useful Git Commands

```
git log --oneline      #Displays summary of each commit.  
git log -<n>          #Displays last n commit details.  
git log --after="<yyyy-mm-dd>"
```

Advanced Git Use

If there's a collision/conflict/whatever (usually at the push stage) because you and someone both editing the same part of the same file, you'll need to manually fix it, which sometimes sucks. You may need to separately run

```
git fetch  
git merge
```

If you want to go back just to look, make sure you've committed any changes and then run

```
git checkout <hex_number>
```

To create a new branch so you can make commits based on an older version (again, make sure you've committed any changes), run

```
git checkout -b <branch_name> <hex_number>To add a Monte Carlo model:
```