Brain Alignment with Ubuntu

– GD Marquart, Burgess Lab, NICHD – Jan. 2016

Assumptions

This guide assumes you have 16-bit unsigned .nrrd files corresponding to the vglut2a reference channel (e.g., **new_line-fish01-01.nrrd**) as well as the pattern to align (e.g., **new_line-fish01-02.nrrd**). *A basic grasp of running terminal commands is also assumed.*

Summary

1) **Installing FFTW & CMTK from source**
2) **Registering brains to a reference**

---

1 Installing FFTW & CMTK from source

While precompiled CMTK binaries are available for download, they lack optimizations that greatly decrease the registration time. We therefore recommend compiling from source as described below.

1.1 Download and extract source files into a temporary directory
1.2 Install FFTW from source
1.3 Install packages required for building CMTK from source
1.4 Install CMTK from source
1.5 Test commands

1.1 Download and extract source files into a temporary directory

- **Download source files**
  - **https://www.nitrc.org/frs/download.php/7357/CMTK-3.2.3-Source.tar.gz**
  - **http://fftw.org/fftw-3.3.4.tar.gz**

- **Extract tars**

TERMINAL
tar -zxvf CMTK-3.2.3-Source.tar.gz **tar -zxvf fftw-3.3.4.tar.gz**

1.2 Install FFTW from source

- **Build and install FFTW's double floating point files**

TERMINAL
./configure --enable-threads --enable-sse2 --enable-avx **make** **sudo make install**

NOTE: this configuration assumes that you have a recent multithreaded Intel processor supporting SSE, SSE2, & AVX runtimes. If you do not, try configuring and installing without these flags.

- **Build and install FFTW's single floating point files**

TERMINAL
./configure --enable-threads --enable-sse2 --enable-avx --enable-single --enable-sse **make** **sudo make install**

NOTE: this configuration assumes that you have a recent multithreaded Intel processor supporting SSE, SSE2, & AVX runtimes. If you do not, try configuring and installing without these flags.

1.3 Install packages required for building CMTK from source

- **Install cmake, cmake-curses-gui, and build-essential dependencies**

TERMINAL
sudo apt-get install build-essential cmake cmake-curses-gui

1.4 Install FFTW from source

- **Make a cmake build subdirectory within CMTK source tree**
- **Run cmake to compile source in the build directory**
- **Run cmake-curses-gui (ccmake) to modify cmake build parameters**

TERMINAL
mkdir build **cd build** **cmake ..** **ccmake .**

- **Change advanced parameters to improve CMTK performance**

press [t] to toggle advanced mode

CMAKE_BUILD_TYPE	Release
CMAKE_CXX_FLAGS	-march=nocona -mmmx -msse -msse2 -mfpmath=sse
CMAKE_C_FLAGS	-march=nocona -mmmx -msse -msse2 -mfpmath=sse

press [c] to configure

press [g] to generate make and exit

NOTE: this configuration assumes that you have a recent multithreaded Intel processor supporting SSE, SSE2, & MMX runtimes. If you do not, try configuring and installing without these flags.

- **Build and install the CMTK**

TERMINAL
make **sudo make install**

- **Add the CMTK binaries to your bash command path**

TERMINAL
nano ~/.bash_profile

TERMINAL
nano ~/.bash_profile

File: ~/.bash_profile
.bash_profile PATH=/usr/local/lib/cmtk/bin:$PATH export PATH

press [ctrl + o] to save
press [ctrl + x] to exit

TERMINAL
source ~/.bash_profile

1.5 Test commands

- **Run a CMTK command that has been added to your bash command path (e.g., registration)**

TERMINAL
registrationx --help-all

---

2 Registering brains to a reference

2.1 Generate rigid registrations
2.2 Generate nonrigid registrations
2.3 Apply transformations
2.4 Generate average brains
2.5 Convert to 16-bit
2.6 Convert to .tifs

2.1 Generate rigid registrations

- **Generate rigid transformations from vglut2a reference channels**

TERMINAL
registrationx --dofs 12 --min-stepsize 1 \ -o new_line-f01_rigid.xform refbrain.nrrd new_line-fish01-01.nrrd

Explanation:
new_line-f01_rigid.xform is the output from the rigid mapping of **new_line-fish01-01.nrrd** to the **refbrain.nrrd** with our recommended parameters (ie., **--dofs 12 --min-stepsize 1**).

2.2 Generate nonrigid registrations

- **Generate nonrigid transformations from the rigid transformations**

TERMINAL
warpx --fast --grid-spacing 100 --smoothness-constraint-weight 1e-1 --grid-refine 2 --min-stepsize 0.25 --adaptive-fix-thresh 0.25 \ -o new_line-f01_nonrigid.xform new_line-f01_rigid.xform

Explanation:
new_line-f01_nonrigid.xform is the output from the nonrigid registration based on the rigid mapping in **new_line-f01_rigid.xform** with our recommended nonrigid parameters (i.e., **--fast --grid-spacing 100 --smoothness-constraint-weight 1e-1 --grid-refine 2 --min-stepsize 0.25 --adaptive-fix-thresh 0.25**).

2.3 Apply transformations

- **Apply nonrigid transformations to the information channels**

TERMINAL
reformatx --pad-out 0 ** **-o new_line-f01-02_warp.nrrd --floating new_line-f01-01.nrrd refbrain.nrrd new_line-f01_nonrigid.xform

Explanation:
new_line-f01-02_warp.nrrd is the resulting registered image stack from **new_line-f01-01.nrrd** aligned to the **refbrain.nrrd** using the **new_line-f01_nonrigid.xform** transformation.

2.4 Generate averages

- **Average across all registered image stacks**

TERMINAL
average_images --n ** **-o new_line-mean3-32bit.nrrd new_line-f01-02_warp.nrrd new_line-f02-02_warp.nrrd new_line-f03-02_warp.nrrd

Explanation:
new_line-mean3-32bit.nrrd is the 32-bit average image stack generated from **new_line-f01-02_warp.nrrd**, **new_line-f02-02_warp.nrrd**, and **new_line-f03-02_warp.nrrd**.

2.5 Convert to 16-bit

- **Convert resulting 32-bit average image stacks to 16-bit**

TERMINAL
convertx **--ushort ** **new_line-mean3-32bit.nrrd new_line-mean3-16bit.nrrd**

Explanation:
new_line-mean3-16bit.nrrd is the 16-bit conversion of **new_line-mean3.nrrd**.

2.6 Convert to .tifs

- **Open 16-bit .nrrds in Fiji and save as .tifs**