Some **AFNI** Scripts

For Nonlinear Warping

and For Time Series Regression:

Real Scripts Used Recently by

RW Cox

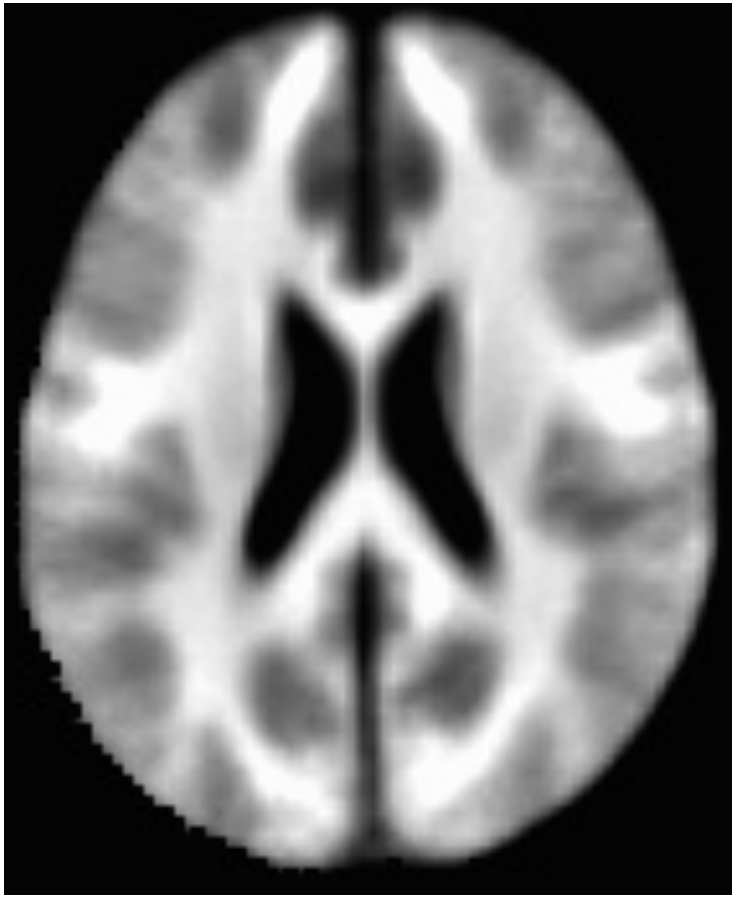
Also see https://arxiv.org/abs/1709.07471
Appendix has processing scripts

Oct 2017

Nonlinear Warping to MNI Template

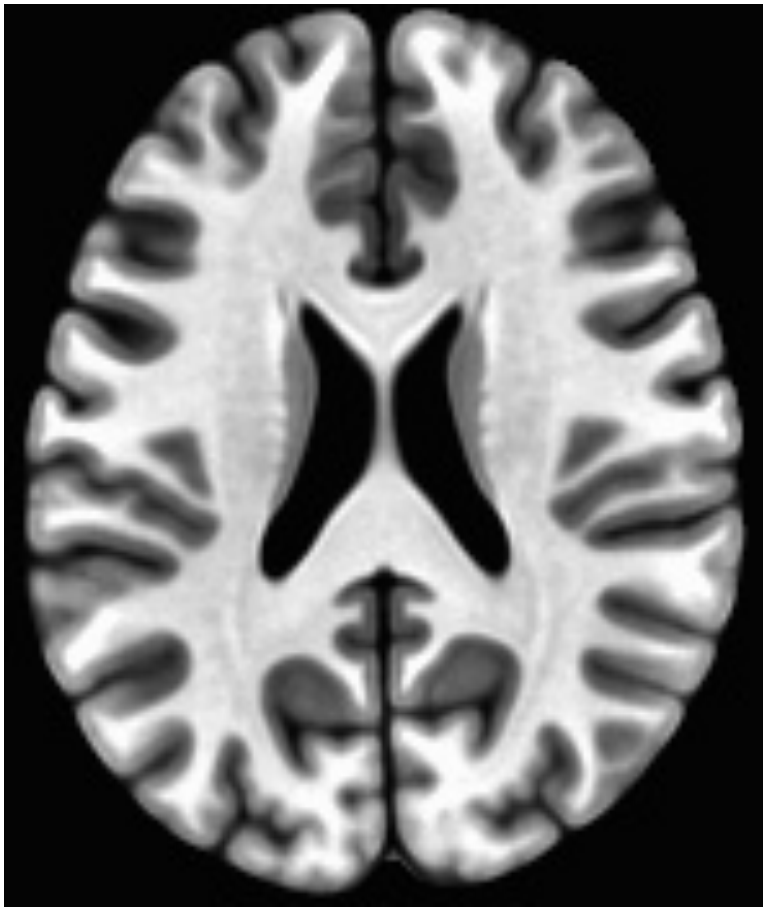
- afni_proc.py *can* do the nonlinear warping for you
  - But, nonlinear warping is slow
  - If you need to re-rerun subject analysis, nonlinear warping will slow the re-run script down *a lot*
- Solution: do the nonlinear warping *before* using afni_proc.py, then supply the warping results so that afni_proc.py will skip doing the warping itself
- Mechanism: the **@SSwarper** script (tcsh)
  - Does Skull Stripping (”SS”) and nonlinear warping
  - Base dataset is **MNI152_2009_template.nii.gz**
    - Nonlinearly warped, not too blurry

Two MNI Templates



MNI152_1mm_uni+tlrc

Affine alignments



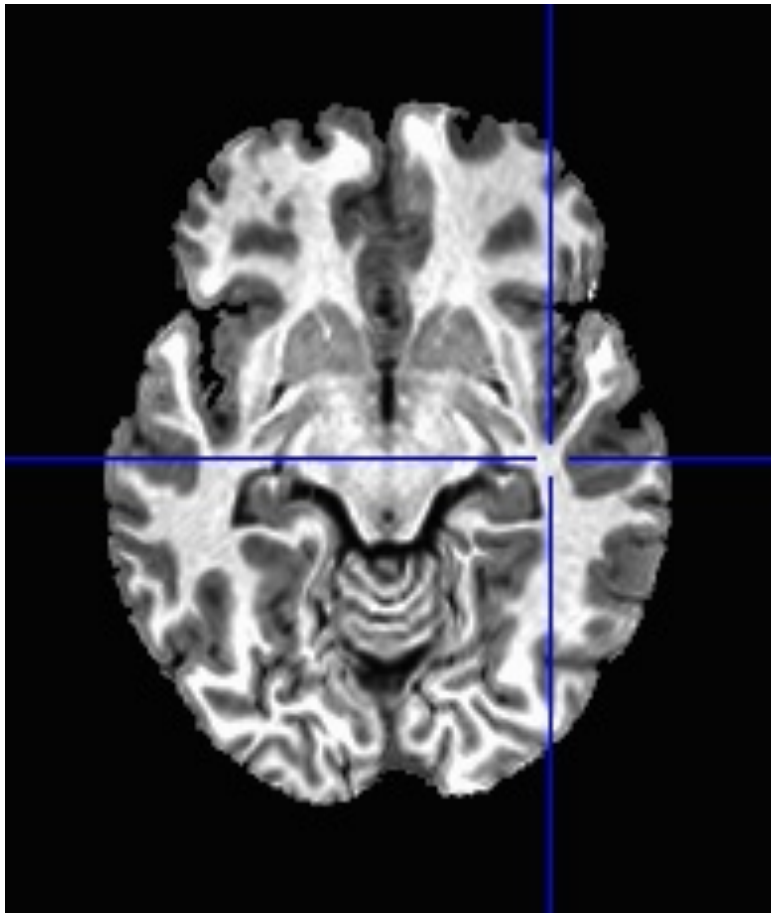
MNI152_2009_template.nii.gz

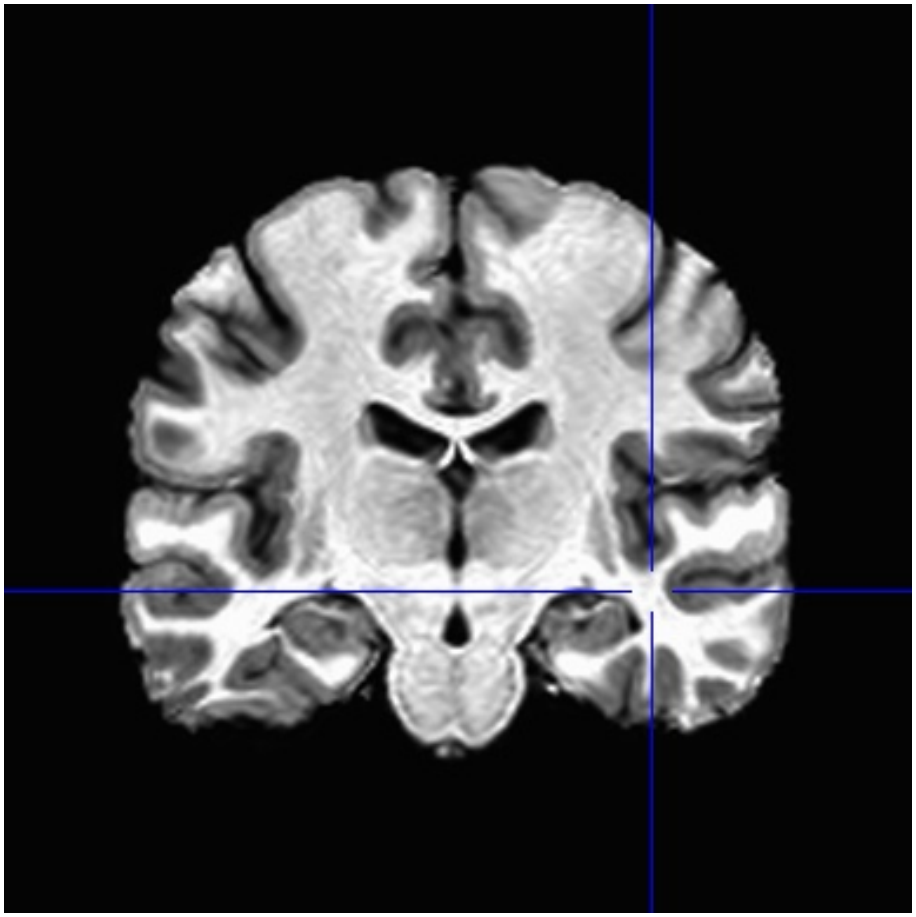
Nonlinear alignments

What @SSwarper Produces

- Inputs:
  - T1-weighted anatomical image of subject (skull-on)
  - Subject ID code, for names of output files
- Outputs (subject ID = **sub007**):
  - **anatSS.sub007.nii**
    - skull-stripped dataset in original coordinates
  - **anatQQ.sub007.nii**
    - skull-stripped dataset, nonlinearly warped to MNI template
  - **anatQQ.sub007.aff12.1D**
    - affine matrix to transform original dataset to MNI template
  - **anatQQ.sub007_WARP.nii**
    - incremental warp from affine transformation to nonlinearly aligned dataset
- These files are needed for later use in afni_proc.py

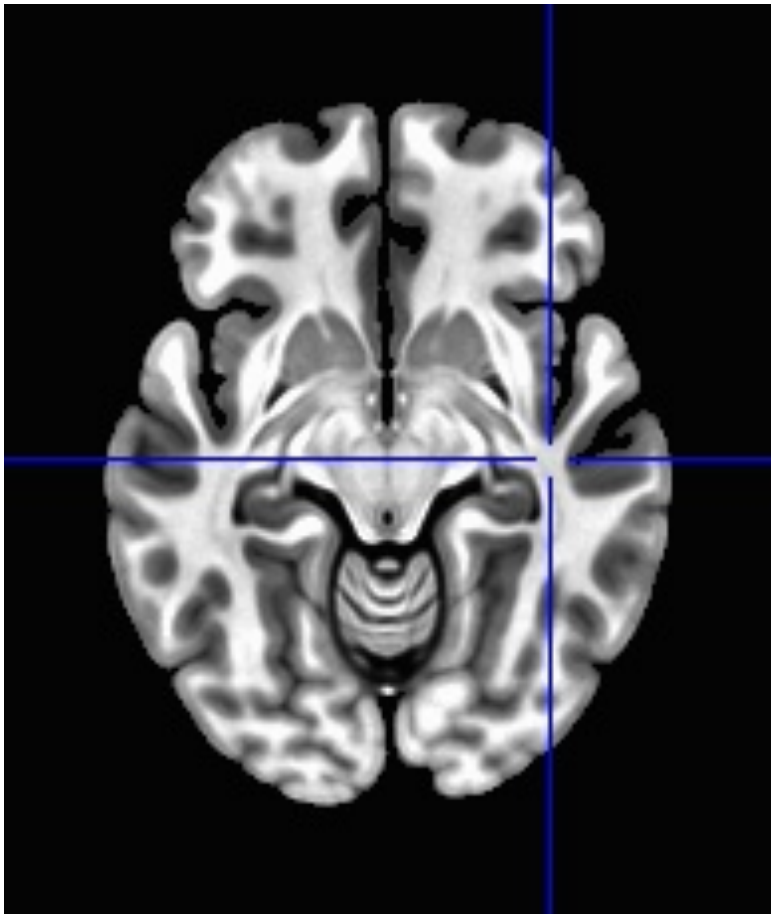
@SSwarper Results

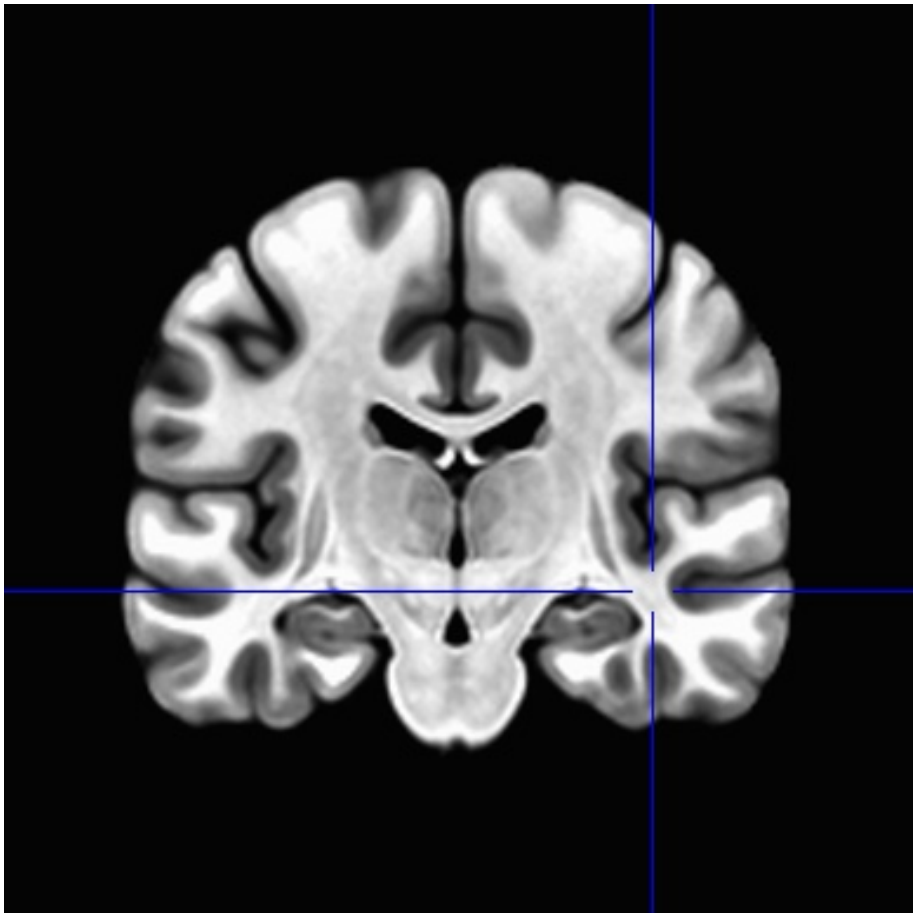




sub00440 from Beijing-Zang
in the FCON-1000 collection

MNI Template Slices





For comparison

Script to Warp One Dataset – page 1

```
#!/bin/tcsh
### Only command line argument is subject ID
set sub = $argv[1]
# set thread count if we are running SLURM
if( $?SLURM_CPUS_PER_TASK )then
  setenv OMP_NUM_THREADS $SLURM_CPUS_PER_TASK
endif
# don't log AFNI programs in ~/.afni.log
# don't try any version checks
# don't auto-compress output files
setenv AFNI_DONT_LOGFILE  YES
setenv AFNI_VERSION_CHECK NO
setenv AFNI_COMPRESSOR    NONE
# topdir = directory above this Scripts directory
set topdir  = `dirname $cwd`
# all input anat datasets are in this directory
cd $topdir/anat_orig
```

Shell variable **sub**

Script to Warp One Dataset – page 2

```
# create final output directory
mkdir -p $topdir/anat_warped
# create temporary directory to hold the work, copy anat there
mkdir -p temp_$sub
cp anat_$sub.nii.gz temp_$sub
cd temp_$sub
### process the anat dataset
@SSwarper anat_$sub.nii.gz $sub
# move the results to where they belong
\mv -f anatSS.${sub}.nii   anatQQ.${sub}.nii     \
        anatQQ.${sub}.aff12.1D anatQQ.${sub}_WARP.nii
$topdir/anat_warped
# delete the temporary directory
cd ..
\rm -rf temp_$sub
time
exit 0
```

Above Script is Submitted for Each Subject

```
#!/bin/tcsh
# This script submits the jobs for the nonlinear warping.
# Uses the 'swarm' command, part of the Linux cluster software SLURM.
unset noclobber
set site = Beijing
# subject ID list
set Slist = ( `cat $site.list.txt` )
# create a file, with 1 line for each case to run
set sname = junk.swarm.warper
if( -f $sname ) \rm $sname
touch $sname
foreach sub ( $Slist )
  echo "tcsh Script_1.warper.csh $sub" >> $sname
end
# run this file via swarm (16 threads per job)
# the 'nimh' partition is local to NIH.
swarm -f $sname -g 24 -t 16 --usecsh --time 2:59:00 \
          --partition nimh,norm --job-name Warper
```

Using Above Results

- Time series processing via afni_proc.py (*of course*)
- Use output files from @SSwarper to do the nonlinear warping
- Next pages show the afni_proc.py command for processing *one* subject
  - First part (not shown) of entire script is set up
    - Setting shell variables with values to control processing
- One copy of script is submitted for each subject, for each processing case
  - e.g., different HRF models **`"$stimresp"`**

afni_proc.py command – all of it

```
afni_proc.py -subj_id $subj                                    \
        -script proc.$subj    -scr_overwrite                   \
        -blocks despike tshift align tlrc volreg               \
                mask scale regress                             \
        -copy_anat $warpdir/anatSS.${subj}.nii                 \
           -anat_has_skull no                                  \
        -dsets $rest_dset                                      \
        -tcat_remove_first_trs 0                               \
        -align_opts_aea -giant_move                            \
            -cost lpc+ZZ                                       \
        -volreg_align_to MIN_OUTLIER                           \
        -volreg_align_e2a                                      \
        -volreg_tlrc_warp                                      \
        -tlrc_base $basedset                                   \
        -volreg_warp_dxyz 2.0                                  \
        -tlrc_NL_warp                                          \
        -tlrc_NL_warped_dsets                                  \
              $warpdir/anatQQ.${subj}.nii                      \
              $warpdir/anatQQ.${subj}.aff12.1D                 \
              $warpdir/anatQQ.${subj}_WARP.nii                 \
        -regress_anaticor_fast                                 \
        -regress_anaticor_fwhm 20                              \
        -regress_stim_times $stimfile                          \
        -regress_stim_labels $stimcase                         \
        -regress_basis "$stimresp"                             \
        -regress_censor_motion 0.2                             \
        -regress_censor_outliers 0.04                          \
        -regress_3dD_stop                                      \
        -regress_make_ideal_sum sum_ideal.1D                   \
        -regress_est_blur_errts                                \
        -regress_reml_exec                                     \
        -regress_run_clustsim no
```

Fragment from a larger script to run regression analysis on one subject (out of hundreds)

afni_proc.py command – part 1

```
afni_proc.py -subj_id $subj                                    \
    -script proc.$subj -scr_overwrite                          \
    -blocks despike tshift align tlrc volreg                   \
                    mask scale regress                         \
    -copy_anat $warpdir/anatSS.${subj}.nii                     \
       -anat_has_skull no                                      \
    -dsets $rest_dset                                          \
    -tcat_remove_first_trs 0                                   \
    -align_opts_aea -giant_move                                \
        -cost lpc+ZZ                                           \
    -volreg_align_to MIN_OUTLIER                               \
    -volreg_align_e2a                                          \
    -volreg_tlrc_warp                                          \
    -tlrc_base $basedset                                       \
    -volreg_warp_dxyz 2.0                                      \
```

afni_proc.py command – part 2

```
-tlrc_NL_warp                                    \
-tlrc_NL_warped_dsets                            \
      $warpdir/anatQQ.${subj}.nii                \
      $warpdir/anatQQ.${subj}.aff12.1D           \
      $warpdir/anatQQ.${subj}_WARP.nii           \
-regress_anaticor_fast                           \
-regress_anaticor_fwhm 20                        \
-regress_stim_times $stimfile                    \
-regress_stim_labels $stimcase                   \
-regress_basis "$stimresp"                       \
-regress_stim_types times                        \
-regress_censor_motion 0.2                       \
-regress_censor_outliers 0.04                    \
-regress_3dD_stop                                \
-regress_make_ideal_sum sum_ideal.1D             \
-regress_est_blur_errts                          \
-regress_reml_exec                               \
-regress_run_clustsim no
```


AND NOW BACK TO
OUR REGULARLY
SCHEDULED
PROGRAMMING