
4dfp Documentation

Release 0.1.0

Avi Snyder, Jon Koller, Haley Acevedo

Sep 19, 2019

1 Contents

3

This website is intended to provide up-to-date documentation on Avi Snyder's 4dfp suite of tools. Currently, it contains usage information for the tools, as well as explanations of the inputs, outputs, and processing steps for BOLD preprocessing scripts. It is a work in progress, with plans to add more worked examples and detailed documentation of additional scripts in the near future. In addition, we plan to pair it with a searchable community discussion site.

This work was performed by Dr. Avi Snyder, Haley Acevedo and Jon Koller and generously supported by the NIL, NIAC, McDonnell Center for Systems Neuroscience and the Psychiatry, Radiology, Neurology and Psychological and Brain Sciences Departments at WU.

1.1 4dfp format

The 4dfp (4-dimensional floating point) format was designed for functional neuroimaging. The four dimensions typically correspond to x, y, z, and time. Three dimensional structural images can be represented in 4dfp format by setting the depth of the fourth dimension to 1.

Note: NIfTI and 4dfp images are ALWAYS y-flipped to each other. Be sure to use 4dfp tools to convert back and forth, so that this is accounted for.

The voxel data are stored in the form of a binary image as one UNIX file. Consequently, 4dfp images may be directly loaded and viewed using IDL, matlab, fsleyes, etc. Information critical to interpreting the binary data (e.g., orientation, image dimensions, voxel dimensions) are stored in separate header file(s). The 4dfp UNIX file name convention is demonstrated below, where filename is any valid filename string:

```
<filename>.4dfp.img           # binary float voxel data
<filename>.4dfp.ifh           # interfile header (ASCII text)
<filename>.4dfp.hdr           # ANALYZE 7.5 header (binary)
<filename>.4dfp.img.rec       # creation history
```

All 4dfp based image analysis programs used at the Washington University School of Medicine Neuroimaging Laboratory (NIL) read/write interfile headers. The minimal 4dfp format is comprised of the binary image data (.img) and the interfile header (.ifh). All NIL image analysis programs maintain an additional rec file (.img.rec), which records the image creation history.

The voxel data are stored in the form of a binary image as one UNIX file. Consequently, 4dfp images may be directly loaded and viewed using IDL, matlab, fsleyes, etc. Information critical to interpreting the binary data (e.g., orientation, image dimensions, voxel dimensions) are stored in separate header file(s). The 4dfp UNIX file name convention is demonstrated below, where filename is any valid filename string:

```
<filename>.4dfp.img           # binary float voxel data
<filename>.4dfp.ifh           # interfile header (ASCII text)
```

(continues on next page)

(continued from previous page)

```
<filename>.4dfp.hdr          # ANALYZE 7.5 header (binary)
<filename>.4dfp.img.rec      # creation history
```

All 4dfp based image analysis programs used at the Washington University School of Medicine Neuroimaging Laboratory (NIL) read/write interfile headers. The minimal 4dfp format is comprised of the binary image data (.img) and the interfile header (.ifh). All NIL image analysis programs maintain an additional rec file (.img.rec), which records the image creation history.

1.1.1 Image data

The 4dfp file format was developed to manage human head images acquired with a Siemens MRI scanner. The imato4dfp utility converts Siemens slice-based image (.ima) files into 4dfp format. This is accomplished by extracting (without reordering) the stored pixel values, converting short int to float and writing the results to the 4dfp image file. Within each 3D volume the Siemens .ima files are read in order of decreasing image number.

Certain 4dfp conventions, especially regarding image orientation, reflect the Siemens Numaris operating system. For all acquisitions, including oblique and double oblique, Numaris determines the principal orientation. It is this orientation that appears in the interfile header and to which following table refers. The orientation-dependent axis flips required to display 4dfp image data in conventional radiologic orientation are tabulated below. It is assumed that the display is written left-to-right and bottom-to-top as in analyze_avw. It is also assumed that the image was acquired with the subject positioned in the scanner head-first and supine.

Acquired orientation	Flip axes	First voxel (after flipping)
2 (axial)	y	right occipital skull-base
3 (coronal)	y z	right skull-base occipital
4 (sagittal)	x y z	occipital skull-base right

The above enumerated axis flips may be effected in analyze_avw at 4dfp image load time. Alternatively, 4dfp format may be converted to ANALYZE 7.5 format using the 4dfptoanalyze utility which automatically performs the indicated axis flips as it converts the voxel value number format from float to short int. 4dfp image data loaded into analyze_avw as prescribed above will be consistently resliced by analyze_avw. That is, all Siemens acquisition orientations will be consistently displayed in all analyze orientations (transverse, coronal, sagittal).

Note: x, y, and z denote voxel indices as ordered in memory. That is, if the x, y, and z indices run from 0 to nx-1, 0 to ny-1, and 0 to nz-1, then coordinate (x,y,z) is stored relative to the first voxel of each frame at offset $x + nx*(y + ny*z)$.

1.1.2 Interfile header

The following is a listing of a 4dfp interfile header file (vm6c_b1.4dfp.ifh). The image data (vm6c_b1.4dfp.img) were acquired in one 128 frame fMRI run. Each frame has dimensions 64 x 64 x 18, The acquired voxels are 3 mm cubic.

```
version of keys           := 3.3
number format            := float
conversion program       := nifti_4dfp
name of data file        := T1w_acpc_dc.4dfp.ifh
number of bytes per pixel := 4
imagedata byte order     := littleendian
orientation              := 2
number of dimensions     := 4
```

(continues on next page)

(continued from previous page)

```

matrix size [1]           := 260
matrix size [2]           := 311
matrix size [3]           := 260
matrix size [4]           := 1
scaling factor (mm/pixel) [1] := 0.700000
scaling factor (mm/pixel) [2] := 0.700000
scaling factor (mm/pixel) [3] := 0.700000
mmpix                    := 0.700000      -0.700000      -0.700000
center                   := 92.0000      -91.7000      -110.0000

```

Various image analysis programs may make use of additional interfile header fields. The minimal set of fields required to interpret the voxel data is listed below:

```

number format            := float
number of bytes per pixel := 4
orientation              := 3
number of dimensions     := 4
scaling factor (mm/pixel) [1] := 3.000000
scaling factor (mm/pixel) [2] := 3.000000
scaling factor (mm/pixel) [3] := 3.000000
matrix size [1]          := 64
matrix size [2]          := 64
matrix size [3]          := 18
matrix size [4]          := 128

```

1.1.3 rec file

The rec file format was designed to capture the creation history of each particular 4dfp image. This is accomplished automatically provided that each UNIX executable which creates 4dfp output also produces a corresponding rec file. Rec files are ASCII text with the following format

```

rec <filename>.4dfp.img `date` `user`
UNIX command line which created <filename>.4dfp.img
rcs $Id$ (program revision code)
image/program specific processing information
...
rec file[s] corresponding to antecedent input 4dfp images
endrec `date` `user`

```

The critical feature of the rec file convention is inclusion of antecedent rec files at all stages of processing. It follows that rec files corresponding to averaged images may grow large. The key words “rec” (first field of first line) and “endrec” (first field of last line) guarantee secure parsing of the accumulated processing history. The following is a listing of the rec file corresponding to the above illustrated interfile header after being passed through `rmspike_4dfp` and `deband_4dfp`

```

rec vm6c_b1_rmsp_dbnd.4dfp.img Thu May 18 17:16:23 2000 avi
/data/petsun4/data1/solaris/deband_4dfp -n4 vm6c_b1_rmsp
$Id: deband_4dfp.c,v 1.8 1999/11/20 00:55:49 avi Exp $
Frame      1 slice multipliers: even=0.837060 odd=1.162940
Frame      2 slice multipliers: even=0.997099 odd=1.002901
Frame      3 slice multipliers: even=0.985484 odd=1.014516
Frame      4 slice multipliers: even=0.986583 odd=1.013417
Functional frame slice multipliers: even=0.986982 odd=1.013018
rec vm6c_b1_rmsp.4dfp.img Thu May 18 17:16:13 2000 avi

```

(continues on next page)

(continued from previous page)

```

/data/petsun4/data1/solaris/rmspike_4dfp -n4 -x33 vm6c_b1
$Header: /data/petsun4/src_solaris/rmspike_4dfp/RCS/rmspike_4dfp.c,v 2.6 1997/05/23_
↪00:49:24 yang Exp $
No spike found in vm6c_b1.4dfp.img
rec vm6c_b1.4dfp.img Thu May 18 17:15:18 2000 avi
/data/petsun4/data1/solaris/imato4dfp2 -fy /data/petsun23/vm6c/siem_im/bold1/5250 7 7_
↪vm6c_b1
$Id: imato4dfp2.c,v 1.12 2000/05/05 00:56:18 avi Exp $
patient_id:          vm6c
institution:         Washington University
manufacturer_model: MAGNETOM VISION
parameter_file_name: Initialized by sequence
sequence_file_name:  /usr/users/tec/nbea_uc_tg2.ekc
sequence_description: ep_fid  90      TR    135.2      TE    37.0/1
tilts:              Cor>Tra -12
4dfp_dimensions:    64          64          18          128
voxel_dimensions:   3.000000  3.000000  3.000000
scan_date:          22-FEB-1999
scan_time:          14:06:33-14:06:33
endrec Thu May 18 17:15:18 2000 avi
endrec
endrec Thu May 18 17:16:26 2000 avi

```

The *brec* (beautify rec file) utility parses rec files and writes to stdout a more easily readable version of the text. Here is the above rec file filtered through *brec*

```

1rec vm6c_b1_rmsp_dbnd.4dfp.img Thu May 18 17:16:23 2000 avi
1  /data/petsun4/data1/solaris/deband_4dfp -n4 vm6c_b1_rmsp
1  $Id: deband_4dfp.c,v 1.8 1999/11/20 00:55:49 avi Exp $
1  Frame          1 slice multipliers: even=0.837060 odd=1.162940
1  Frame          2 slice multipliers: even=0.997099 odd=1.002901
1  Frame          3 slice multipliers: even=0.985484 odd=1.014516
1  Frame          4 slice multipliers: even=0.986583 odd=1.013417
1  Functional frame slice multipliers: even=0.986982 odd=1.013018
2  rec vm6c_b1_rmsp.4dfp.img Thu May 18 17:16:13 2000 avi
2  /data/petsun4/data1/solaris/rmspike_4dfp -n4 -x33 vm6c_b1
2  $Header: /data/petsun4/src_solaris/rmspike_4dfp/RCS/rmspike_4dfp.c,v 2.6_
↪1997/05/23 00:49:24 yan
2  No spike found in vm6c_b1.4dfp.img
3  rec vm6c_b1.4dfp.img Thu May 18 17:15:18 2000 avi
3  /data/petsun4/data1/solaris/imato4dfp2 -fy /data/petsun23/vm6c/
↪siem_im/bold1/5250 7 7 vm6c
3  $Id: imato4dfp2.c,v 1.12 2000/05/05 00:56:18 avi Exp $
3  patient_id:          vm6c
3  institution:         Washington University
3  manufacturer_model:  MAGNETOM VISION
3  parameter_file_name: Initialized by sequence
3  sequence_file_name:  /usr/users/tec/nbea_uc_tg2.ekc
3  sequence_description: ep_fid  90      TR    135.2      TE    _
↪37.0/1
3  tilts:              Cor>Tra -12
3  4dfp_dimensions:    64          64          18          128
3  voxel_dimensions:   3.000000  3.000000  3.000000
3  scan_date:          22-FEB-1999
3  scan_time:          14:06:33-14:06:33
3  endrec Thu May 18 17:15:18 2000 avi

```

(continues on next page)

(continued from previous page)

```
2      endrec
lendrec Thu May 18 17:16:26 2000  avi
```

1.2 Dicom utilities

1.2.1 dcm_sort

sort dicom files by study series (used for flat directory structures)

Usage: `dcm_sort <dicom_directory>`

Examples:

```
dcm_sort /data/petsun52/data1/JHILL/04271737
dcm_sort /cdrom/botv/10251349 -p930589002 -c
```

Options

-d	verbose debug mode
-c	copy files (default symbolically link)
-t	toggle use of -t in call to <code>dcm_dump_file</code> (default ON)
-i	take files with integer filenames
-e<ext>	take files with specified extension
-r<str>	take files with filenames containing specified string (default is 7 digits)
-p<str>	take files only with dicom field 'PAT Patient Name' matching specified string

N.B.: `dcm_sort` removes existing single study subdirectories

N.B.: `dcm_sort` puts unclassifiable DICOMs into subdirectory `study0`

1.2.2 pseudo_dcm_sort.csh

sort dicom files by study series (used for nested directory structures)

Usage: `pseudo_dcm_sort.csh <dicom directory>`

Examples:

```
pseudo_dcm_sort.csh RAW
```

Options

-d	debug mode (set echo)
-s	DICOM files are within a subdirectory of numeric subdirectories (default directly in numeric subdirectory)
-e	identify DICOM files by specified extension (default extension = dcm)
-r	identify DICOM files by specified root (default root = 'MR*')
-i	take DICOM files with integer filenames
-t	toggle off use of -t in call to <code>dcm_dump_file</code> (default ON)

N.B.: dicom subdirectories must be numeric N.B.: default subdirectory of numeric subdirectory is 'DICOM'

1.3 DTI

Tip: For more information on script inputs, see *Params/Instructions files*.

1.3.1 generic_DWI_script_090219

generic DTI processing

Usage: generic_DWI_script_090219 params_file [instructions_file]

1.3.2 cross_DWI_imgreg_4dfp

atlas transform new data based on previous session results

Usage: cross_DWI_imgreg_4dfp <curr_dwi> <day1_dwi_path> <day1_dwi> <atlas_representative_target> [options]

Examples:

```
cross_DWI_imgreg_4dfp 6770_dwi /data/petsun24/data1/5575 5575_dwi [abspath/]711-2Y
```

N.B.: cross_DWI_imgreg_4dfp must be run in the current DWI directory

1.4 fcMRI oriented scripts

Tip: For more information on script inputs, see *Params/Instructions files*.

1.4.1 fcMRI_preproc_161012.csh

fcMRI preprocessing including nuisance variable regression

Attention: fcMRI_preproc_161012 assumes successful completion of BOLD preprocessing (*cross_bold_pp_161012.csh*).

Usage: fcMRI_preproc_<version>.csh <params file> [instructions file]

Examples:

```
fcMRI_preproc_161012.csh VB16168.params
```

Required parameters

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
target		atlas to be used for alignment
FSDir	<str>	freesurfer directory containing mri/aparc+aseg.mgz
srcdir	<str>	source directory path (contains run directories)
fbolds	<array>	list of bold run folders
TR_vol	<flt>	time per frame (s)
CSF_lcube	<int>	cube dimension (in voxels) used by <i>qntv_4dfp</i> for CSF (recommended: 3)
CSF_sd1t	<flt>	threshold CSF sd1 image (recommended: 25)
CSF_svdt	<flt>	limit regressor covariance condition number to $(1/\{\})^2$ for CSF (recommended: .2)
WM_lcube	<int>	cube dimension (in voxels) used by <i>qntv_4dfp</i> for WM (recommended: 5)
WM_svdt	<flt>	limit regressor covariance condition number to $(1/\{\})^2$ for WM (recommended: .15)

Optional parameters

Variable	Values	Default	Description
FCdir	<str>	FCmaps	output directory name
day1_patid	<str>		patient directory for first session (if patid is not patient's first session)
MB	0,1	0	skip slice timing correction and debanding
blur	<flt>	no blur	f_half for spatial blur
bpsp_params	<array>		options for <i>bandpass_4dfp</i> (only -E,M,F specified by default; no default bands)
conc	<str>		pre-existing conc file to use
fmtfile	<str>		format file
FDthresh	<flt>		frame displacement thresholds
FDtype	1,2	1	frame displacement calculation (1 = absolute value, 2 = squares)
anat_aveb	<flt>	0	run_dvar_4dfp preblur in mm (for small voxels, set to 10mm)
anat_avet	<flt>	(compute)	run_dvar_4dfp criterion
min_frames	<int>	numframes / 2	minimum number of remaining frames after scrubbing for participant to be included
CSF_excl_lim	<flt>	0.126	mask threshold for CSF
task_regressor	<str>		optional externally supplied task regressor
noGSR	1	0	suppress global signal (WB) regression

Processing steps

Step description	Function	Output
Generate FS masks if they don't already exist	<i>Generate_FS_Masks_AZ</i>	Atlas/ <patid>_orig_to_<target>_t4 <patid>_FSWB_on_<target>_333.4dfp.img <patid>_GM_on_<target>_333_comp_b60.4dfp.img <patid>_(WM,CS)_erode_on_<target>_333_clus.4dfp.img
Create conc file	<i>conc_4dfp</i>	<FCdir>/ \$conc or, <patid>[_faln_dbnd]_xr3d_uwrp_atl.conc
Compute frame censoring using FD (if \$FDthresh specified) and DVARs – skipped if \$fmtfile is specified	FD.awk, <i>run_dvar_4dfp</i>	movement/ <patid>[_faln_dbnd]_xr3d.FD atlas/ <patid>[_faln_dbnd]_xr3d.FD.format (if \$FDthresh) <patid>_func_vols(.vals, .dat, .crit, .xmgr, .format)
Create average censored image	<i>actmapf_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_ave.4dfp.img
Compute defined mask and apply it	<i>compute_defined_4dfp,</i> <i>maskimg_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_dfnd.4dfp.img atlas/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_dfndm.4dfp.img
Compute initial sd1 mean	<i>var_4dfp, qnt_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_sd1.4dfp.img
Make timeseries zero mean	<i>var_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_uout.conc
Make movement regressors for each bold run	<i>mat2dat, conc_4dfp,</i> <i>bandpass_4dfp, 4dfp-</i> <i>toascii</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_dat.conc <patid>[_faln_dbnd]_xr3d_dat_bpss.conc <patid>_<patid>_mov_regressors.dat
Temporal bandpass filter using \$bpss_params	<i>bandpass_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss.conc
Make whole brain regressors including the 1st derivative	<i>qnt_4dfp</i>	<FCdir>/ <patid>_WB_regressors_dt.dat
Make extra-axial CSF regressors (thresholded at \$CSF_excl_lim)	<i>maskimg_4dfp,</i> <i>qntv_4dfp</i>	<FCdir>/ <patid>_CSF_mask.4dfp.img <patid>_CSF_dies.4dfp.img <patid>_CSF_regressors.dat
Make venticle regressors	<i>maskimg_4dfp,</i> <i>qntv_4dfp</i>	<FCdir>/ <patid>_vent_mask.4dfp.img <patid>_vent_dies.4dfp.img <patid>_vent_regressors.dat
Make white matter regressors	<i>maskimg_4dfp,</i> <i>qntv_4dfp</i>	<FCdir>/ <patid>_WM_mask.4dfp.img <patid>_WM_dies.4dfp.img <patid>_WM_regressors.dat
Paste nuisance regressors (mov, CSF, vent, WM, task) together and run covariance analysis	<i>covariance</i> (-D250)	
Paste together WB and WB derivative, and covaried nuisance regressors		<FCdir>/ <patid>_nuisance_regressors.dat
Remove nuisance regressors out of volumetric time series	<i>glm_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_resid.conc <patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_coeff.4dfp.img
Compute sd1 mean for bandpass-filtered and nuisance regressed data	<i>var_4dfp, qnt_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_resid_sd1.4dfp (comparison to initial WB mean reported to stdout)
Spatial blur with f_half = \$blur if specified	<i>gauss_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_resid_g<blur>
Compute sd1 mean for blurred data if \$blur	<i>var_4dfp, qnt_4dfp</i>	<FCdir>/ <patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_resid_g<blur> (mean reported to stdout)

1.4.2 seed_correl_161012.csh

compute multi-volume correlation maps

Attention: `seed_correl_161012` assumes successful completion of BOLD preprocessing (`cross_bold_pp_161012.csh`) and fcMRI preprocessing (`fcMRI_preproc_161012.csh`).

Usage: `seed_correl_161012.csh` <parameters file> [instructions] [options]

Examples:

```
seed_correl_161012.csh VB16168.params
```

Options

-noblur	analyze unblurred version of preprocessed data
-A	use format in atlas subdirectory (default FCmaps directory)

Required variables

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
ROI_dir	<str>	directory containing ROI image(s)
skip	<int>	number of pre-steady state frames

ROI specification variables (required)

Variable	Values	Description
Option 1		
ROI_img	<str>	base filename for single ROI image
Option 2		
ROI_list	<str array>	list of base filenames for ROI images
Option 3		
ROI_listfile	<str>	list file (.lst) of base filenames for ROI image

Optional variables

Variable	Values	Default	Description
FC_dir	<str>	FCmaps	output directory name
MB	0,1	0	skip slice timing correction and debanding
blur	<flt>	no blur	f_half for spatial blur
bpss_params	<array>		options for <i>bandpass_4dfp</i> (only -E,M,F specified by default; no default bands)
conc	<str>		pre-existing conc file to use
fmtfile	<str>		format file

Processing Steps

Step description	Function	Output
Create (multi-volume) ROI image if \$ROIlistfile or \$ROIlist specified	<i>paste_4dfp</i>	
Calculate seed (ROI) regressors	<i>qntm_4dfp</i>	<FCdir>/ <patid>_seed_regressors.dat <patid>_seed_regressors.rec
Compute total correlations	<i>glm_4dfp</i> (-t)	<FCdir>/<patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_resid[_g<blu
Mask total correlation image by defined voxels	<i>maskimg_4dfp</i>	<FCdir>/<patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_resid[_g<blu
Fisher z transform total correlation	<i>rho2z_4dfp</i>	<FCdir>/<patid>[_faln_dbnd]_xr3d_uwrp_atl_bpss_resid[_g<blu
Compute zero-lag ROI-ROI correlation matrix (if number of ROIs <= 256)	<i>covariance</i> (-u, -o, -m0)	<FCdir>/<patid>_seed_regressors_CCR.dat

1.4.3 Generate_FS_Masks_AZS.csh

generate Freesurfer masks for whole brain, WM, GM, CSF

Usage: Generate_FS_Masks_AZS.csh <parameters file> [instructions]

Examples:

```
Generate_FS_Masks_AZS.csh FCS_039_A_1.params ../uwrp_process_Stroke_SMG_Subjects.  
↪params
```

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
FSdir	<str>	freesurfer directory containing mri/aparc+aseg.mgz
target		atlas to be used for alignment

1.5 fMRI oriented scripts

Tip: For more information on script inputs, see *Params/Instructions files*.

1.5.1 cross_bold_pp_161012.csh

generic EPI (BOLD) pre-processing

Usage: cross_bold_pp_161012.csh <params file> [instructions_file]

Examples:

```
cross_bold_pp_161012.csh VB16168.params
```

Required parameters

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
irun	<str array>	list of run folders
fstd	<int array>	list of scan numbers that map to run folders
mprs	<int array>	list of mprage scan numbers
inpath	<str>	top-level directory for raw DICOM data (i.e. directory containing study folders if \$sorted)
target		atlas to be used for alignment
go	0,1	if calls should be executed (if 0, statements will only be printed, not executed)
nx	<int>	number of voxels on the x-axis
ny	<int>	number of voxels on the y-axis
skip	<int>	number of pre-steady state frames
TR_vol	<flt>	time per frame (s)
TR_slc	<flt>	time per slice (s) (set to 0 to have it computed)
epidir	0,1	direction of EPI slices (0 = inferior to superior, 1 = superior to inferior)
economy	<int>	level of removal for intermediate files created during execution (see table below)
epi2atl	0,1,2	if EPI to atlas transform is required (0 = no transform, 1 = transform to 333 space, 2 = skip to resampling step)
FDthresh	<flt>	frame displacement thresholds
normode	0,1	if per-frame volume intensity should be modified
dwll	<flt>	EPI dwell time (echo spacing) (ms)
ped	x,x-,y,y-,z,z-	EPI phase encoding direction
rsam_cmnd	<str>	script to use for resampling (recommended: <i>one_step_resample.csh</i>)

Economy value	Files to be removed
> 2	original bold run images
> 3	frame-aligned images (_faln)
> 4	cross-realigned avg images (_r3d_avg or xr3d_BC_avg if \$BiasField) (only if \$epi2atl == 0)

Field map correction parameters (required)

Variable	Values	Description
Option 1 (<i>Measured field map</i> - gradient echo)		
gre	<int array>	gradient echo measured field map scan numbers (magnitude image should be first, followed by phase image)
delta	<flt>	difference between field map echo times (ms)
TE_vol	<int>	echo time (ms)
Option 2 (<i>Measured field map</i> - spin echo)		
sefm	<int array>	spin echo measured field maps
TE_vol	<int>	echo time (ms)
Option 3 (<i>Mean field map</i>)		
FMmean		mean field map image
Option 4 (<i>basis_opt field map</i>)		
FMmean		mean field map image
FMBases		??

Optional parameters

Tip: Although `tse` and `pdt2` are optional, you should specify one or the other if you have them in order to get a better registration to atlas.

Variable	Value	Default	Description
<code>tse</code>	<int array>		list of tse scan numbers
<code>pdt2</code>	<int array>		list of proton-density weighted scan numbers
<code>day1_patid</code>	<str>		patient directory for first session (if <code>patid</code> is not patient's first session)
<code>day1_path</code>	<str>		path to day1 atlas directory (required if <code>day1_patid</code> is set)
<code>scrdir</code>	<str>		scratch directory to be
<code>sorted</code>	0,1	0	if <code>dcm</code> sort has already been run
<code>MB</code>	0,1	0	skip slice timing correction and debanding
<code>sx</code>	<int>	1	unpacked x-dimension squeeze factor (<i>unpack_4dfp</i>)
<code>sy</code>	<int>	1	unpacked y-dimension squeeze factor (<i>unpack_4dfp</i>)
<code>E4dfp</code>	0,1	0	if 4dfp files already exist (skips <i>dcm_to_4dfp</i>)
<code>use_anat_ave</code>	0,1	0	use <code>_anat_ave</code> epi image (default is <code>_func_vols_ave</code>)
<code>min_frames</code>	<int>	240	minimum number of remaining frames after scrubbing for participant to be included
<code>interleave</code>	-S		sequential slice acquisition (<i>frame_align_4dfp</i>)
<code>seqstr</code>	<str>		specify [MB] slice sequence (counting from 1) as a comma-separated (no spaces) integer string (if non-standard interleaving)
<code>MBfac</code>	<int>	1	multiband factor
<code>lomotil</code>	1-6		lowpass filter for specified motion parameter (<i>mat2dat</i> -1), used for filtered FD (set to 2 for y direction)
<code>BiasField</code>	0,1	0	perform bias field correction
<code>FDtype</code>	1,2	1	frame displacement calculation (1 = absolute value, 2 = squares)
<code>anat_aveb</code>	<flt>		<code>run_dvar_4dfp</code> preblur in mm (for small voxels, set to 10mm)
<code>anat_avec</code>	<flt>	(compute)	<code>run_dvar_4dfp</code> criterion (set excessively high to skip DVARS censoring)
<code>cross_day_nostretch</code>	0,1	0	disable stretch for cross-day transform
<code>Gad</code>	0,1	0	if gadolinium contrast was used

Additional optional parameters

Warning: Only specify the following variables if the action is desired. They will happen if you specify them at all (even if you set them to 0).

Variable	Description
<code>goto_UNWARP</code>	Immediately go to unwarp step
<code>epi_zflip</code>	flip z when unpacking (<i>unpack_4dfp</i>)
<code>Siemens_interleave</code>	Siemens interleave order (<i>frame_align_4dfp</i>)
<code>nounpack</code>	skips unpacking step

Processing steps

Step description	Function	Output
Convert bold run DICOM data to 4dfp format	<i>dcm_to_4dfp</i>	
Convert mosaic format to volume (if not \$nounpack)	<i>unpack_4dfp</i>	bold<irun>/ <patid>_b<irun>.4dfp.img
Correct slice timing and odd/even slice intensities (if not \$MB)	<i>frame_align_4dfp</i> , <i>deband_4dfp</i>	bold<irun>/ <patid>_b<irun>_faln.4dfp.img <patid>_b<irun>_faln_dbnd.4dfp.img
Motion correction via rigid body transform of each volume to reference frame	<i>cross_realign3d_4dfp</i>	bold<irun>/ <patid>_b<irun>[_faln_dbnd]_xr3d.4dfp.img <patid>_b<irun>[_faln_dbnd]_r3d_avg.4dfp.img <patid>_b<irun>[_faln_dbnd]_xr3d.mat
Bias field correction (if \$BiasField)	FSL bet, FSL fast, extend_fast_4dfp	bold<irun>/ <patid>_b<irun>[_faln_dbnd]_xr3d_BC_avg.4dfp.im atlas/ <patid>[_faln_dbnd]_xr3d_avg_brain.nii.gz <patid>[_faln_dbnd]_xr3d_avg_brain_restore(.4dfp.img, .nii.gz)
Compute and apply within-run mode 1000 normalization	<i>normalize_4dfp</i> , <i>scale_4dfp</i>	bold<irun>/ <patid>_b<irun>[_faln_dbnd]_xr3d[_BC]_norm.4dfp
Extract/format movement data from cross-realign output	<i>mat2dat</i>	movement/ <patid>_b<irun>[_faln_dbnd]_xr3d(.dat, .ddat, .rdat)
Extract EPI first frame (anatomy) image and create functional volume conc file	<i>paste_4dfp</i> , <i>conc_4dfp</i>	atlas/ <patid>_anat_ave.4dfp.img <patid>_func_vols.conc
Compute high movement frames using FD (if \$FDthresh specified) and DVARS (stops here if \$min_frames criteria not met)	FD.awk (using .ddat movement file), <i>run_dvar_4dfp</i>	movement/ <patid>[_faln_dbnd]_xr3d.FD atlas/ <patid>[_faln_dbnd]_xr3d.FD.format (if \$FDthresh) <patid>_func_vols(.vals, .dat, .crit, .xmgr, .format)
Make func_vols_ave image with high movement frames removed	<i>actmapf_4dfp</i>	atlas/ <patid>_func_vols_ave.4dfp.img
Compute cross-session BOLD atlas transform if \$day1_patid specified (then skips to applying EPI transform step)	<i>cross_day_imgreg_4dfp</i> , <i>t4_mul</i>	atlas/ <patid>_anat_ave_to_<target>_t4 (and other intermediate t4 files)
Convert MPRAGE DICOM data to 4dfp format (if not \$E4dfp)	<i>dcm_to_4dfp</i>	atlas/ <patid>_mpr#.4dfp.img
Convert MPRAGE images to transverse orientation (if not already)	<i>C2T_4dfp</i> or <i>S2T_4dfp</i>	atlas/ <patid>_mpr#T.4dfp.img
Compute MPRAGE average	<i>avgmpr_4dfp</i>	atlas/ <patid>_mpr_n#.4dfp.img
Create transverse t2w image (if \$tse or \$pdt)	<i>collate_slice_4dfp</i> if \$tse, <i>extract_frame_4dfp</i> (second frame) if \$pdt, <i>C2T_4dfp</i> or <i>S2T_4dfp</i>	atlas/ <patid>_t2w[T].4dfp.img
Compute EPI to atlas transform	<i>epi2mpr2atl2_4dfp</i> if neither \$tse nor \$pdt, otherwise <i>epi2t2w2mpr2atl2_4dfp</i>	atlas/ <patid>_anat_ave_to_<target>_t4 (and other intermediate t4 files)
Make atlas transformed EPI average image and t2w in 111, 222, and 333 atlas space	<i>t4img_4dfp</i>	atlas/ <patid>_(anatfunc_vols)_ave_on_<target>_111.4dfp.img <patid>_(anatfunc_vols)_ave_on_<target>_222.4dfp.img <patid>_(anatfunc_vols)_ave_on_<target>_333.4dfp.img
Compute field mapping correction	<i>fmri_unwarp_170616.tc</i>	unwarp/ <patid>_(anatfunc_vols)_ave_uwarp.4dfp.img
Compute and apply unwarped epi to atlas transform	<i>imgreg_4dfp</i> , <i>t4_mul</i> , <i>t4img_4dfp</i>	unwarp/ <patid>_(anatfunc_vols)_ave_uwarp_on_<target>_111.4 <patid>_(anatfunc_vols)_ave_uwarp_on_<target>_222.4dfp.img <patid>_(anatfunc_vols)_ave_uwarp_on_<target>_333.4dfp.img
One-step resample unwarped images	\$rsam_cmnd	bold<irun>/ <patid>[_faln_dbnd]_xr3d_uwarp_atl.4dfp.img

1.5.2 fmri_unwarp_170616.tcsh

distortion correction wrapper script for fMRI preprocessing

Measured field map

Usage: fmri_unwarp_170616.tcsh -map <patid> <epi> <mag> <phase> <dwell> <te> <ped> <delta>

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
epi	<4dfp img>	EPI anat image (_anat_ave or _func_vols_ave)
mag	<nifti img>	magnitude field map image
phase	<nifti img>	phase field map image
dwell	<flt>	EPI dwell time (echo spacing) (ms)
te	<int>	echo time (ms)
ped	x,x-,y,y-,z,z-	EPI phase encoding direction
delta	<flt>	difference between field map echo times (ms) (required only for gradient-echo field map)

Mean field map

Usage: fmri_unwarp_170616.tcsh -mean <epi> <FMmean> <epi_to_atl_t4> <dwell> <ped>

Variable	Values	Description
epi	<4dfp img>	EPI anat image (_anat_ave or _func_vols_ave)
FMmean		mean field map image
epi_to_atl_t4	<t4 file>	EPI to atlas t4 file
dwell	<flt>	EPI dwell time (echo spacing) (ms)
ped	x,x-,y,y-,z,z-	EPI phase encoding direction

basis_opt field map

Usage: fmri_unwarp_170616.tcsh -basis <epi> <t2w> <FMmean> <FMbases> <epi_to_t2w_t4> <epi_to_atl_t4> <dwell> <ped> <nbasis> [t2w brain mask]

Variable	Values	Description
epi	<4dfp img>	EPI anat image (_anat_ave or _func_vols_ave)
t2w	<4dfp img>	structural 4dfp image (can be t2w or mpr)
FMmean		mean field map image
epi_to_t2w_t4	<t4 file>	EPI to T2-weighted t4 file
epi_to_atl_t4	<t4 file>	EPI to atlas t4 file
dwell	<flt>	EPI dwell time (echo spacing) (ms)
ped	x,x-,y,y-,z,z-	EPI phase encoding direction
nbasis	<int>	??

N.B.: with option -basis, basis_opt optimizes the <dwell> value (aka, echo spacing) by default

1.5.3 sefm_pp_AZS.csh

merge AP/PA into one image and run topup to derive field map

Usage: sefm_pp_AZS.csh <params file> [instructions file]

Examples:

```
sefm_pp_AZS.csh PSQ0001_s1.params ../PSQ_study.params
```

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
sefm	<int array>	spin echo measured field maps
sorted	1	if dcm sort has already been run (must be sorted)

1.5.4 one_step_resample.csh

one step resampling with support for bias field correction

Usage: one_step_resample.csh <parameters file> [instructions]

Examples:

```
one_step_resample.csh VB16168.params
```

Params variables

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
day1_patid	<str>	patient directory for first session (if patid is not patient's first session)
day1_path	<str>	path to day1 atlas directory (required if day1_patid is set)
irun	<str array>	list of run folders

Instructions variables

Variable	Values	Description
use_anat_ave	0,1	use _anat_ave epi image (default is _func_vols_ave)
outrés	111,222,333	output resolution (default = 333)
target		atlas to be used for alignment
MB	0,1	skip slice timing correction and debanding
BiasField	0,1	perform bias field correction

1.5.5 epi2mpr2atlv_4dfp

EPI → T1W → atlas

Usage: epi2mpr2atlv_4dfp <epi_anat> <mpr_anat> [useold] [atlas target [711-2? OR -T<target including path>] [-S<atlas space>] [noinit]

Examples:

```
epi2mpr2atlv_4dfp stem9_anat_ave stem9_654-3 useold 711-2C
epi2mpr2atlv_4dfp stem9_anat_ave stem9_654-3 useold -T/data/cninds01/atlas/NP765 -
↪ S711-2B
```

Options

useold	inhibits re-computation of all t4 files
noinit	inhibits reinitialization of epi->mpr t4 files
-T<str>	atlas target (specified string should include path) (default is 711-2B)
-S	specifies the atlas space (requires -T) (currently only 711-2B is supported)

N.B.: Any image argument may include a path, e.g., /data/petmr1/data7/stem/96_06_14_stem9/stem9_654-3

N.B.: All named images must be in either ANALYZE or 4dfp format. ANALYZE will be converted to 4dfp

1.5.6 epi2t2w2mpr2atlv_4dfp

EPI → T2W → T1W → atlas (8 parameter cross-modal; for “low” res fMRI)

Usage: epi2t2w2mpr2atlv_4dfp <epi_anat> <t2w_anat> <mpr_anat> [useold] [atlas_target]

Examples:

```
epi2t2w2mpr2atlv_4dfp stem9_anat_ave stem9_643-2 stem9_654-3 useold 711-2Y
```

Step description	Function	Inputs	Output
Compute MPR to atlas transform	mpr2atl_4dfp		`\${mpr}_to_\${target}_t4
Blur images	<i>gauss_4dfp</i>	t_half = 1.1	`\${mpr}_g11 `\${t2w}_g11 `\${epi}_g11
Compute MPR brain mask	msktgen_4dfp		`\${mpr}_mskt
Compute (iteratively) T2w to MPR transform	<i>imgreg_4dfp</i> , using blurred (*_g11) images as source/target	w/ unmasked mpr (source_mask = none) modes[1] = 1024 + 3 w/ masked mpr (source_mask = `\${mpr}_mskt) modes[2] = 1024 + 3 modes[3] = 2048 + 3 + 8 + 64 modes[4] = 8192 + 2048 + 3 + 8 + 64	`\${t2w}_to_\${mpr}_t4
Compute EPI brain mask	<i>gauss_4dfp</i> , <i>imgmax_4dfp</i> , <i>img2msk_4dfp</i> , <i>masking_4dfp</i>	f_half = .3 threshold = .2 * max(blurred epi)	`\${epi}_msk
Compute EPI (iteratively) to T2w transform	<i>imgreg_4dfp</i> , using blurred (*_g11) images as source/target	w/ original epi (source_msk = none) modes[1] = 1024 + 3 w/ masked epi (source_msk=\${epi}_msk) modes[2] = 1024 + 3 modes[3] = 3072 + 3 + 8 + 64 modes[4] = 8192 + 2048 + 3 + 8 + 64	`\${epi}_to_\${t2w}_t4
Combine transforms	<i>t4_mul</i>	`\${t2w}_to_\${mpr}_t4, `\${mpr}_to_\${target}_t4 `\${epi}_to_\${t2w}_t4, `\${t2w}_to_\${target}_t4 `\${epi}_to_\${t2w}_t4, `\${t2w}_to_\${mpr}_t4	`\${t2w}_to_\${target}_t4 `\${epi}_to_\${target}_t4 `\${epi}_to_\${mpr}_t4

N.B.: Any argument may include a path, e.g., /data/petmr1/data7/stem/96_06_14_stem9/stem9_654-3

N.B.: All named images must be in either ANALYZE or 4dfp format. ANALYZE will be converted to 4dfp

N.B.: ‘useold’ instructs `epi2t2w2mpr2atlv_4dfp` to use existing t4 files

N.B.: The default `atlas_target` is 711-2B

1.5.7 epi2t2w2mpr2atl1_4dfp

EPI → T2W → T1W → atlas (9 parameter cross-modal; for “high” res fMRI)

Usage: `epi2t2w2mpr2atl1_4dfp <epi_anat> <t2w_anat> <mpr_anat> [useold] [711-2? OR -T<Target including path>] [-S<atlas space>]`

Examples:

```
epi2t2w2mpr2atl1_4dfp stem9_anat_ave stem9_643-2 stem9_654-3 711-2B
epi2t2w2mpr2atl1_4dfp stem9_anat_ave stem9_654-3 useold -T/data/cninds01/atlas/NP765 -
→S711-2B
```

Step description	Function	Inputs	Output
Compute MPR to atlas transform	<code>mpr2atl_4dfp</code>		<code>#{mpr}_to_#{target}_t4</code>
Compute MPR brain mask	<code>msktgen_4dfp</code>		<code>#{mpr}_mskt</code>
Compute (iteratively) T2w to MPR transform	<code>imgreg_4dfp</code>	w/ original mpr (source_mask = none) modes[1] = 1024 + 3 w/ masked mpr (source_mask = <code>#{mpr}_mskt</code>) modes[2] = 1024 + 3 modes[3] = 2048 + 3 + 8 + 64 modes[4] = 8192 + 2048 + 3 + 8	<code>#{t2w}_to_#{mpr}_t4</code>
Compute EPI brain mask	<code>gauss_4dfp</code> , <code>imgmax_4dfp</code> , <code>img2msk_4dfp</code> , <code>masking_4dfp</code>	<code>f_half = .3</code> threshold = <code>.2 * max(blurred epi)</code>	<code>#{epi}_msk</code>
Compute EPI (iteratively) to T2w transform	<code>imgreg_4dfp</code>	w/ original epi (source_msk = none) modes[1] = 1024 + 3 w/ masked epi (source_msk= <code>#{epi}_msk</code>) modes[2] = 1024 + 3 modes[3] = 3072 + 3 + 8 modes[4] = 8192 + 2048 + 3 + 8	<code>#{epi}_to_#{t2w}_t4</code>
Combine transforms	<code>t4_mul</code>	<code>#{t2w}_to_#{mpr}_t4</code> , <code>#{mpr}_to_#{target}_t4</code> <code>#{epi}_to_#{t2w}_t4</code> , <code>#{t2w}_to_#{target}_t4</code> <code>#{epi}_to_#{t2w}_t4</code> , <code>#{t2w}_to_#{mpr}_t4</code>	<code>#{t2w}_to_#{target}_t4</code> <code>#{epi}_to_#{target}_t4</code> <code>#{epi}_to_#{mpr}_t4</code>

N.B.: Any image argument may include a path, e.g., `/data/petmr1/data7/stem/96_06_14_stem9/stem9_654-3`

N.B.: All named images must be in 4dfp format

N.B.: -S specifies the atlas space. The only currently supported atlas space is 711-2B

1.5.8 epi2t2w2mpr2atl2_4dfp

EPI → T2W → T1W → atlas (6 parameter cross-modal; best for distorted fMRI)

Usage: `epi2t2w2mpr2atl2_4dfp <epi_anat> <t2w_anat> <mpr_anat> [useold] [711-2? OR -T<Target including path>] [-S<atlas space>]`

Examples:

```
epi2t2w2mpr2atl2_4dfp stem9_anat_ave stem9_643-2 stem9_654-3 711-2B
epi2t2w2mpr2atl2_4dfp stem9_anat_ave stem9_654-3 useold -T/data/cninds01/atlas/NP765 -
↳S711-2B
```

Processing steps

Step description	Function	Inputs	Output
Compute MPR to atlas transform	<code>mpr2atl_4dfp</code>		<code>\${mpr}_to_\${target}_t4</code>
Compute T2w to MPR transform	<code>imgreg_4dfp</code>	modes[1] = 4096 + 3 modes[2] = 1024 + 3 modes[3] = 2048 + 3 modes[4] = 2048 + 3 [+ 8 + 64 if thin t2w] modes[5] = modes[4] + 8192	<code>\${t2w}_to_\${mpr}_t4</code>
Compute EPI brain mask	<code>gauss_4dfp</code> , <code>imgmax_4dfp</code> , <code>img2msk_4dfp</code> , <code>masking_4dfp</code>	f_half = .3 threshold = .2 * max(blurred epi)	<code>\${epi}_msk</code>
Compute (iteratively) EPI to T2w transform	<code>imgreg_4dfp</code>	w/ original epi (source_msk = none) modes[1] = 4096 + 3 modes[2] = 1024 + 3 w/ masked epi (source_msk = <code>\${epi}_msk</code>) modes[3] = 1024 + 3 modes[4] = 3072 + 3 modes[5] = 8192 + 2048 + 3	<code>\${epi}_to_\${t2w}_t4</code>
Combine transforms	<code>t4_mul</code>	<code>\${t2w}_to_\${mpr}_t4</code> , <code>\${mpr}_to_\${target}_t4</code> , <code>\${epi}_to_\${t2w}_t4</code> , <code>\${t2w}_to_\${target}_t4</code> , <code>\${epi}_to_\${t2w}_t4</code> , <code>\${t2w}_to_\${mpr}_t4</code>	<code>\${t2w}_to_\${target}_t4</code> <code>\${epi}_to_\${target}_t4</code> <code>\${epi}_to_\${mpr}_t4</code>

N.B.: Any image argument may include a path, e.g., /data/petmr1/data7/stem/96_06_14_stem9/stem9_654-3

N.B.: All named images must be in 4dfp format

N.B.: -S specifies the atlas space. The only currently supported atlas space is 711-2B

1.5.9 cross_day_imgreg_4dfp

link first session atlas transform to subsequent sessions via EPI “anat_ave”

Usage: `cross_day_imgreg_4dfp <curr_patid> <day1_atlas_path> <day1_patid> <atlas_representative_target> [options]`

Examples:

```
cross_day_imgreg_4dfp tpj0202 /data/petsun24/data1/tpj0201/atlas tpj0201 711-2Y
cross_day_imgreg_4dfp tpj0202 /data/petsun24/data1/tpj0201/atlas tpj0201 -T/data/
↳cninds01/data2/ATLAS/ALLEGRA_Y_111
```

Options

-a<str>	specify image filename trailer (default = "anat_ave")
-nostretch	disable stretch in transform
-setecho	set echo
-S<str>	specify atlas space (default=711-2B)

N.B.: `cross_day_imgreg_4dfp` must be run in the current atlas directory

N.B.: <atlas_representative_target> may be of form 711-2? OR -T/path/image

1.5.10 compute_run_sd1.csh

run `var_4dfp -sn4` on all bold directories (*xr3d_norm and *xr3d_atl) and makes movies

Usage: `compute_run_sd1.csh <patid>`

Examples:

```
compute_run_sd1.csh VB15792
```

1.5.11 cross_day_imgreg_4dfp

compute cross-session BOLD atlas transform

Usage: `cross_day_imgreg_4dfp <curr_patid> <day1_atlas_path> <day1_patid> <atlas_representative_target> [options]`

Examples:

```
cross_day_imgreg_4dfp tpj0202 /data/petsun24/data1/tpj0201/atlas tpj0201 711-2Y
cross_day_imgreg_4dfp tpj0202 /data/petsun24/data1/tpj0201/atlas tpj0201 -T/data/
↪cninds01/data2/ATLAS/ALLEGRA_Y_111
```

Options

-a<str>	specify image filename trailer (default = "anat_ave")
-nostretch	disable stretch in transform
-setecho	set echo
-S<str>	specify atlas space (default=711-2B)

N.B.: `cross_day_imgreg_4dfp` must be run in the current atlas directory

N.B.: <atlas_representative_target> may be of form 711-2? OR -T/path/image

1.5.12 run_dvar_4dfp

compute format (identify frames with too much motion) (*dvar_4dfp* wrapper)

Usage: `run_dvar_4dfp <(conc) concfile> [options]`

Options

-d	debug mode
-v	verbose mode
-p<str>	specify printer on which to plot generated .dat.ps file
-P<str>	print previously generated results on specified printer (run on SunOS)
-x<flt>	set frame rejection threshold (default = mode + 2.5*(left s.d.) over non-skipped frames)

N.B.: run_dvar_4dfp is a wrapper for dvar_4dfp

N.B.: options -b -m -n -t are passed to dvar_4dfp

N.B.: option -s is always passed to dvar_4dfp

1.5.13 conc_4dfp

create conc file

Usage: conc_4dfp <(conc) outroot> <(4dfp) 1> <(4dfp) 2> ...

Examples:

```
conc_4dfp vb13157_faln_dbnd_xr3d_at1 vb13157_b?_faln_dbnd_xr3d_at1.4dfp.img
```

Options

-w	supress inclusion of current working directory in listed file path
-l<str>	read input 4dfp list

N.B.: output conc file always has extension “conc”

N.B.: only files in or below the current working directory can be correctly addressed

1.5.14 conc_mv

update conc file 4dfp image pointers

Usage: conc_mv <conc file> <from> <to>

Examples:

```
conc_mv TC26851_rmstp_faln_dbnd_xr3d_at1.conc /data/nil-bluearc/raichle/gusnard/np751_↵  
↵auto_evolve/AVI_TEST
```

Options

-v	verbose mode
-t	practice mode (<conc file> not changed)

1.5.15 conc2format

compute conc-specific format using a fixed number of pre-steady state frames

Examples:

```
conc2format vb13157_faln_dbnd_xr3d_atl.conc 4
```

Options

-v	verbose mode
-X	label first frame of each run 'X'

1.5.16 RFX2.csh

random effects analysis (1 or 2 groups)

Usage: RFX2.csh <list_group1> <Nimage_group1> [<list_group2> <Nimage_group2>]

Options

-d	debug mode
-R	suppress creation of large rec files (bootstrap mode)

N.B.: <list_group[12]> name 4dfp images on which to run the t-test

N.B.: <Nimage_group[12]> are 4dfp 'n' images (number of subjects for which each voxel is defined)

N.B.: If one group is entered a t-test will be run on this group against the null hypothesis of 0

N.B.: If two groups are entered a t-test will be run comparing the two groups and the computed statistic is Welch's approximate t' (Eqn. 8.11, p. 129 in Zar.)

1.6 Miscellaneous scripts

1.6.1 freesurfer2mpr_4dfp

transform freesurfer generated images back to atlas space

Usage: freesurfer2mpr_4dfp <(4dfp) mpr> <(4dfp) orig> [options]

Examples:

```
freesurfer2mpr_4dfp vc1234_654-3[.4dfp.img] vc1234_orig
freesurfer2mpr_4dfp vc1234_654-3 vc1234_orig -T711-2V -alh.ribbon.mgz -arh.ribbon_
↪apply
```

Options

-skew	general affine orig->mpr registration (default 6 parameter rigid body)
-T<target>	specify atlas representative target
-a<segimg>	add named (4dfp format) freesurfer segemntation result to "apply" list
apply	proceed directly to transform (4dfp format) segmentations
force	force atlas transformation of segmentation results even if it already exists
setecho	set echo

N.B.: <(4dfp) orig> is the freesurfer-resampled 256x256x256 coronal mpr N.B.: the default "apply" list includes (4dfp format) images named *parc* and *aseg*

Tip: You must convert the freesurfer-created mgz (i.e. orig, aparc+aseg) images to 4dfp before running this script. For more details, see *Creating a Freesurfer ROI Mask*.

1.6.2 split_ROIs

split peak_4dfp ROI image into multiple mask images

Usage: split_ROIs <4dfp ROI file> [start_ROI_number] [end_ROI_number] [options]

Examples:

```
split_ROIs sum_condition_time_anova_ROI[.4dfp[.img]] 0 82
```

Options

[<startend>_ROI_number]	ROI numbers count from 0 (fidl convention) (defaults are 0, nROI-1)
-x	flip ROI to contralateral hemisphere
-0	name output mask file by ROI number (default name as in ifh)
-d<int>	specify difference between ROI number and voxel value (default 2)

N.B.: split_ROIs output files are put into the subdirectory ./single_ROIs

1.6.3 brec

parse rec file by procedure depth

Usage: brec <my_file[.rec]> [-depth_limit]

1.7 Registration scripts

1.7.1 mpr2atl_4dfp

single T1W → atlas

Usage: mpr2atl_4dfp <mpr_anat> [options]

Examples:

```
mpr2atl_4dfp vc1234_654-3[.4dfp.img]
mpr2atl_4dfp vc1234_654-3[.4dfp.img] -T/data/petsun23/data1/atlas/NP345_111[.4dfp.
→img] -S711-2B
```

Options

711-2<C O Y K L G H V F>	specify 711-2? series atlas representative target image
-T<target including path>	specify arbitrary atlas representative target image
-S<atlas space>*	specify atlas space (default=711-2B space)
crossmodal	use cross-modal mpr->target registration
useold	suppress recomputation of existing t4 file
redo	suppress initialization of existing t4 file
setecho	set echo

N.B.: <mpr_anat> may include a path, e.g., /data/petmr1/data7/stem9/scout/654-3 N.B.: <mpr_anat> must be in either ANALYZE short int or 4dfp format; ANALYZE will be converted to 4dfp

1.7.2 mpr2atl1_4dfp

T1W → atlas

Usage: mpr2atl1_4dfp <mpr_anat> [options]

Examples:

```
mpr2atl1_4dfp vc1234_654-3 [.4dfp.img]
mpr2atl1_4dfp vc1234_654-3 [.4dfp.img] -T/data/petsun23/data1/atlas/NP345 [.4dfp.img]
```

Options

-T<target including path>	specify arbitrary atlas representative target image
crossmodal	use cross-modal mpr->target registration
useold	suppress recomputation of existing t4 file
redo	suppress t4 file initialization
setecho	set echo

1.7.3 avgmpr_4dfp

multiple T1W → atlas

Usage: avgmpr_4dfp <img1> <img2> ... <avgout> [useold] [711-2<B-Z> OR -T<Target including path>]

Examples:

```
avgmpr_4dfp va2345_mpr1 va2345_mpr2 va2345_mpr3 va2345_mpr4 va2345_mpr_n4
avgmpr_4dfp va2345_mpr1 va2345_mpr2 va2345_mpr3 va2345_mpr4 none
```

Options

useold	suppresses unnecessary recomputation of atlas transformation, e.g., <img1>_to_711-2B_t4
--------	---

N.B.: Each named image must be in 4dfp format and acquired in the same subject. Mixed orientations are allowed. Any component image filename may include a unix path.

N.B.: If <avgout> = “none”, t4 and lst files will be generated but averaged images will not.

1.7.4 t2w2mpr_4dfp

T2W → T1W → atlas¹

Usage: t2w2mpr_4dfp <4dfp mprage> <4dfp t2w> [options]

Examples:

```
t2w2mpr_4dfp vc6383_130-4 vc6383_130-5
```

¹ Assumes pre-existing atlas-transform t4 file

Options

-T<target>	specify atlas target (<target> may include absolute path)
nostretch	disable stretch
setecho	set echo
debug	debug mode

N.B.: `t2w2mpr_4dfp` assumes that `<4dfp mprage>` is in the current working directory and that its atlas transform, e.g., `vc6383_130-4_to_711-2V_t4` exists and is in the current working directory

1.7.5 epi2t1w_4dfp

EPI → T1W¹

Usage: `epi2t1w_4dfp <4dfp epi> <4dfp t1w> <tarstr>`

Examples:

```
epi2t1w_4dfp 070630_4TT00280_t1w 070630_4TT00280_anat_ave -T/data/cninds01/data2/  
↪atlas/TRIO_Y_NDC
```

N.B.: `epi2t1w_4dfp` assumes that the `<4dfp t1w>` atlas transform, e.g., `070630_4TT00280_t1w_to_TRIO_Y_NDC_t4` exists and is in the current working directory

N.B.: `<tarstr>` is either `'711-2?'` or `'-T/targetpath/target'`

1.7.6 make_mprage_avg_4dfp

compute T1W anatomical average for group (list directed)¹

Usage: `make_mprage_avg_4dfp <study_id> <t4file_list>`

Examples:

```
make_mprage_avg_4dfp NP659_all NP659_mpr_t4.lst
```

N.B.: the output average will be named `<study_id>_mpr_avg`

N.B.: `make_mprage_avg_4dfp` assumes that the MP-RAGE 4dfp image files are in the same directories together their atlas transform `t4files`

N.B.: `<t4file_list>` should list the `t4files` including path (e.g.: `vc12605c/PROCESSED/vc12605c_949-3_to_711-2Y_t4`)

Inputs

`t4file_list` `ls vc????c/PROCESSED/*t4 | awk '$1 !~/anat/' >! <t4file_list>`

1.7.7 msktgen_4dfp

create tailored mask (by inversion of atlas transform)¹

Usage: `msktgen_4dfp <(4dfp) image> [threshold] -T<target including path> -S<atlas space>`

Examples:

```
msktgen_4dfp 4859-5_mpr
msktgen_4dfp 4859-5_mpr -T/data/petsun29/data1/atlas/NP345_111[.4dfp.img] -S711-2B
```

Options

threshold	specify threshold for mask (default is 200) - use a higher threshold for a tighter mask and vice versa
-S<atlas space>	specify atlas space (default=711-2B space)
-T<target>	specify atlas target

N.B.: `msktgen_4dfp` uses the first legitimate atlas transform t4 file it sees in the current working directory, i.e., one of `<image>_to_711-2*_t4` or one of `<image>_to_<target>_t4`

1.7.8 cross_mpr_imgreg_4dfp

compute cross-session T1W atlas transform¹

Usage: `cross_mpr_imgreg_4dfp <session1_abspath> <session2_abspath> <target>`

Examples:: `cross_mpr_imgreg_4dfp /data/disk1/P44W_16800_L1 /data/disk2/P44W_16800_L2`
`711-2L cross_mpr_imgreg_4dfp /data/disk1/P44W_16800_L1 /data/disk2/P44W_16800_L2`
`/bmr01/01/nmrgrp/avi/P44W_C_111`

N.B.: `<target>` may be of the form '711-2[B-Z]' OR '-T[mypath/]mytarget'

N.B.: `cross_mpr_imgreg_4dfp` assumes that each session patid is `<sessionpath>:t`

1.7.9 newatl_init_4dfp

initialize creation of a cohort-specific atlas-representative target image¹

Attention: After successful execution, execute `newatl_refine_4dfp`

Usage: `newatl_init_4dfp <t4list> <newatl>`

Examples:

```
newatl_init_4dfp symph-mpr_to_711-2B_t4.lst 711-2S
```

Options

`-m` | mask all input images (each input image must be paired with a 4dfp mask named `<input_image>_mask`)

N.B.: `<t4list>` is a text file listing the absolute addresses of extant atlas transform t4files corresponding to a set of structural images

N.B.: `<newatl>` specifies the name of the new atlas representative target image

N.B.: `<t4list>` itself and the t4files named in it may exist in other directories

N.B.: all images (`*.4dfp.img` and `*.4dfp.ifh`) referred to in `<t4list>` must exist either in their original directory or in the current working directory - `newatl_init_4dfp` will copy these images into the cwd as necessary

1.7.10 newat1_refine_4dfp

refine cohort-specific atlas-representative target image¹

Attention: Execute after successful completion of newat1_init_4dfp

Usage: newat1_refine_4dfp <newat1>

Examples:

```
newat1_refine_4dfp 711-2S
```

Options

-b	suppress gauss 1.1 pre-blur of component images
-m	mask all input images ((each input image must be paired with a 4dfp mask named <input_image>_mask)
-T<str>	set reference target to specified image (default = /data/petsun43/data1/atlas/711-2B)

N.B.: <newat1> specifies the name of the new atlas representative target image

N.B.: all images (*.4dfp.img and *.4dfp.ifh) referred to in <newat1>.lst must exist in the working directory

1.8 Deprecated scripts

1.8.1 cross_bold_pp

generic EPI (BOLD) pre-processing

Usage: cross_bold_pp_<version>.csh <params file> [instructions_file]

Examples:

```
cross_bold_pp_161012.csh VB16168.params  
generic_cross_bold_pp_090115.csh VB16168.params
```

cross_bold_pp_130702.csh

Required parameters

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
irun	<str array>	list of run folders
fstd	<int array>	list of scan numbers that map to run folders
mprs	<int array>	list of mprage scan numbers
target		atlas to be used for alignment
go	0,1	if calls should be executed (if 0, statements will only be printed, not executed)
nx	<int>	number of voxels on the x-axis
ny	<int>	number of voxels on the y-axis
skip	<int>	number of pre-steady state frames
TR_vol	<flt>	time per frame (s)
TR_slc	<flt>	time per slice (s) (set to 0 to have it computed)
epidir	0,1	direction of EPI slices (0 = inferior to superior, 1 = superior to inferior)
economy	<int>	level of removal for intermediate files created during execution (see table below)
epi2atl	0,1,2	if EPI to atlas transform is required (0 = no transform, 1 = transform to 333 space, 2 = skip to resampling step)
normode	0,1	if per-frame volume intensity should be modified
day1_patid	<str>	patient directory for first session (if patid is not patient's first session)
day1_path	<str>	path to day1 atlas directory (required if day1_patid is set)
uwrp_cmnd	<str>	script to use for unwarping
rsam_cmnd	<str>	script to use for resampling

Economy value	Files to be removed
> 2	original bold run images
> 3	frame-aligned images (<code>_faln</code>)
> 4	cross-realigned avg images (<code>_r3d_avg</code> or <code>xr3d_BC_avg</code> if <code>\$BiasField</code>) (only if <code>\$epi2atl == 0</code>)

Field map correction variables (required)

Variable	Values	Description
Option 1 (<i>Measured field map</i> - gradient echo)		
gre	<int array>	gradient echo measured field map scan numbers (magnitude image should be first, followed by phase image)
delta	<flt>	difference between field map echo times (ms)
TE_vol	<int>	echo time (ms)
Option 2 (<i>basis_opt field map</i>)		
FMmean		mean field map image
FMBases		??

Optional parameters

Tip: Although `tse` and `pdt2` are optional, you should specify one or the other if you have them in order to get a better registration to atlas.

Variable	Values	Default	Description
tse	<int array>		list of tse scan numbers
pdt2	<int array>		list of proton-density weighted scan numbers
scrdir	<str>		scratch directory to be
sorted	0,1	0	if dcm sort has already been run
MB	0,1	0	skip slice timing correction and debanding
sx	<int>	1	unpacked x-dimension squeeze factor (<i>unpack_4dfp</i>)
sy	<int>	1	unpacked y-dimension squeeze factor (<i>unpack_4dfp</i>)
E4dfp	0,1	0	if 4dfp files already exist (skips <i>dcm_to_4dfp</i>)
use_anat_ave	0,1	0	use _anat_ave epi image (default is _func_vols_ave)
min_frames	<int>	240	minimum number of remaining frames after scrubbing for participant to be included
interleave	-S	interleave	sequential slice acquisition (<i>frame_align_4dfp</i>)
MBfac	<int>	1	multiband factor
anat_aveb	<flt>	0	run_dvar_4dfp preblur in mm (for small voxels, set to 10mm)
anat_avet	<flt>	(compute)	run_dvar_4dfp criterion (set excessively high to skip DVARS censoring)
cross_day_nostretch	0,1	0	disable stretch for cross-day transform
Gad	0,1	0	if gadolinium contrast was used

Additional optional parameters

Warning: Only specify the following variables if the action is desired. They will happen if you specify them at all (even if you set them to 0).

Variable	Description
goto_UNWARP	immediately go to unwarp step
epi_zflip	flip z when unpacking (<i>unpack_4dfp</i>)
Siemens_interleave	enables Siemens interleave order (<i>frame_align_4dfp</i>)
nounpack	skips unpacking step

Processing steps

- Convert bold run dicoms to 4dfp format (*dcm_to_4dfp*)
- Covert mosaic format to volume – if not \$nounpack (*unpack_4dfp*)
- Correct slice timing and odd/even slice intensities – if not \$MB (*frame_align_4dfp*, *deband_4dfp*)
- Motion correction via rigid body transform of each volume to reference frame (*cross_realign3d_4dfp*)
- Compute and apply mode 1000 normalization (*normalize_4dfp*, *scale_4dfp*)
- Extract/format movement data from on cross_realign3d_4dfp output (*mat2dat*)
- Extract EPI first frame (anatomy) image (*paste_4dfp*)
- Make func_vols_ave image with high movement frames removed (DVARS) (*actmapf_4dfp*)
- Compute cross-session BOLD atlas transform – if \$day1_patid specified for current patid (*cross_day_imgreg_4dfp*)

- Convert MPRAGE dicoms to 4dfp format (*dcm_to_4dfp*)
- Compute MPRAGE atlas transforms (*mpr2atl_4dfp* with first mpr if \$Gad, otherwise *avgmpr_4dfp*)
- Compute EPI atlas transform

(Show/Hide Details)

- Make atlas transformed EPI anat_ave and t2w in 111, 222, and 333 atlas space (*t4img_4dfp*)
- Compute field mapping correction (\$uwrp_cmnd)
- Compute and apply unwarped epi to atlas transform (*imgreg_4dfp*, *t4_mul*, *t4img_4dfp*)
- Resample unwarped images (\$rsam_cmnd)
- Make average atlas-aligned, unwarped image (*actmapf_4dfp*)

cross_bold_pp_121215.csh**Required parameters**

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
target		atlas to be used for alignment
irun	<str array>	list of run folders
fstd	<int array>	list of scan numbers that map to run folders
mprs	<int array>	list of mprage scan numbers
go	0,1	if calls should be executed (if 0, statements will only be printed, not executed)
nx	<int>	number of voxels on the x-axis
ny	<int>	number of voxels on the y-axis
skip	<int>	number of pre-steady state frames
TR_vol	<flt>	time per frame (s)
TR_slc	<flt>	time per slice (s) (set to 0 to have it computed)
epidir	0,1	direction of EPI slices (0 = inferior to superior, 1 = superior to inferior)
economy	<int>	level of removal for intermediate files created during execution (see table below)
epi2atl	0,1,2	if EPI to atlas transform is required (0 = no transform, 1 = transform to 333 space, 2 = skip to resampling step)
normode	0,1	if per-frame volume intensity should be modified

Economy value	Files to be removed
> 2	original bold images
> 3	frame-aligned images (<i>_faln</i>)
> 4	debanded images (<i>_faln_dbnd</i>) (only if \$epi2atl == 0)
> 6	normalized images (<i>_norm</i>)

Optional parameters

Tip: Although `tse` and `pd2` are optional, you should specify one or the other if you have them in order to get a better registration to atlas.

Variable	Values	Default	Description
<code>tse</code>	<int array>		list of tse scan numbers
<code>pd2</code>	<int array>		list of proton-density weighted scan numbers
<code>t1w</code>	<int array>		list of t1w scan numbers
<code>scrdir</code>	<str>		scratch directory to be
<code>to_MNI152</code>	0,1	0	transform to MNI152 atlas space
<code>day1_patid</code>	<str>		patient directory for first session (if patid is not patient's first session)
<code>day1_path</code>	<str>		path to day1 atlas directory (required if day1_patid is set)
<code>sorted</code>	0,1	0	if dcm sort has already been run

Additional optional parameters

Warning: Only specify the following variables if the action is desired. They will happen if you specify them at all (even if you set them to 0).

Variable	Description
<code>Siemens_interleave</code>	enables Siemens interleave order (<i>frame_align_4dfp</i>)

Processing steps

- Convert bold run dicoms to 4dfp format (*dcm_to_4dfp*)
- Covert mosaic format to volume (*unpack_4dfp*)
- Correct slice timing and odd/even slice intensities (*frame_align_4dfp*, *deband_4dfp*)
- Motion correction via rigid body transform of each volume to reference frame (*cross_realign3d_4dfp*)
- Compute and apply mode 1000 normalization (*normalize_4dfp*, *scale_4dfp*)
- Extract/format movement data from on *cross_realign3d_4dfp* output (*mat2dat*)
- Extract EPI first frame (anatomy) image (*paste_4dfp*)
- Move anatomy image (*anat_ave*) to atlas directory
- Compute cross-session BOLD atlas transform if `$day1_patid` specified for current patid (*cross_day_imgreg_4dfp*)
- Convert MPRAGE dicoms to 4dfp format (*dcm_to_4dfp*)
- Compute MPRAGE atlas transforms (*avgmpr_4dfp*)
- Compute EPI atlas transform

(Show/Hide Details)

- Make atlas transformed EPI anat_ave in 111, 222, and 333 (711-2B or MNI152 if \$to_MNI152) atlas space (*t4img_4dfp*)
- Make cross-realigned atlas-transformed resampled BOLD 4dfp stacks (*t4_xr3d_4dfp*)

generic_cross_bold_pp_090115

Required parameters

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
irun	<str array>	list of run folders
fstd	<int array>	list of scan numbers that map to run folders
mprs	<int array>	list of mprage scan numbers
target		atlas to be used for alignment
go	0,1	if calls should be executed (if 0, statements will only be printed, not executed)
nx	<int>	number of voxels on the x-axis
ny	<int>	number of voxels on the y-axis
skip	<int>	number of pre-steady state frames
TR_vol	<flt>	time per frame (s)
TR_slc	<flt>	time per slice (s) (set to 0 to have it computed)
epidir	0,1	direction of EPI slices (0 = inferior to superior, 1 = superior to inferior)
economy	<int>	level of removal for intermediate files created during execution (see table below)
epi2atl	0,1,2	if EPI to atlas transform is required (0 = no transform, 1 = transform to 333 space, 2 = skip to resampling step)
normode	0,1	if per-frame volume intensity should be modified

Economy value	Files to be removed
> 2	original bold images
> 3	frame-aligned images (<i>_faln</i>)
> 4	debanded images (<i>_faln_dbnd</i>) (only if \$epi2atl == 0)
> 6	normalized images (<i>_norm</i>)

Optional parameters

Tip: Although *tse* and *pdt2* are optional, you should specify one or the other if you have them in order to get a better registration to atlas.

Variable	Values	Default	Description
tse	<int array>		list of tse scan numbers
pdt2	<int array>		list of proton-density weighted scan numbers
t1w	<int array>		list of t1w scan numbers
scrdir	<str>		scratch directory to be
sorted	0,1	0	if dcm sort has already been run
MB	0,1	0	skip slice timing correction and debanding
sx	<int>	1	unpacked x-dimension squeeze factor (<i>unpack_4dfp</i>)
sy	<int>	1	unpacked y-dimension squeeze factor (<i>unpack_4dfp</i>)
E4dfp	0,1	0	if 4dfp files already exist (skips <i>dcm_to_4dfp</i>)
interleave	-S		sequential slice acquisition (<i>frame_align_4dfp</i>)
MBfac	<int>	1	multiband factor
day1_patid	<str>		patient directory for first session (if patid is not patient's first session)
day1_path	<str>		path to day1 atlas directory (required if day1_patid is set)
cross_day_nostretch	0,1	0	disable stretch for cross-day transform
Gad	0,1	0	if gadolinium contrast was used

Additional optional parameters

Warning: Only specify the following variables if the action is desired. They will happen if you specify them at all (even if you set them to 0).

Variable	Description
epi_zflip	flip z when unpacking (<i>unpack_4dfp</i>)
Siemens_interleave	enables Siemens interleave order (<i>frame_align_4dfp</i>)
nounpack	skips unpacking step

Processing steps

- Convert bold run dicoms to 4dfp format (*dcm_to_4dfp*)
- Covert mosaic format to volume – if not \$nounpack (*unpack_4dfp*)
- Correct slice timing and odd/even slice intensities – if not \$MB (*frame_align_4dfp*, *deband_4dfp*)
- Motion correction via rigid body transform of each volume to reference frame (*cross_realign3d_4dfp*)
- Compute and apply mode 1000 normalization (*normalize_4dfp*, *scale_4dfp*)
- Extract/format movement data from on cross_realign3d_4dfp output (*mat2dat*)
- Extract EPI first frame (anatomy) image (*paste_4dfp*)
- Move anatomy image (anat_ave) to atlas directory
- Compute cross-session BOLD atlas transform if \$day1_patid specified for current patid (*cross_day_imgreg_4dfp*)
- Convert MPRAGE dicoms to 4dfp format (*dcm_to_4dfp*)

- Compute MPRAGE atlas transforms (*mpr2atl1_4dfp* with first mpr if \$Gad, otherwise *avgmpr_4dfp*)
- Compute EPI to atlas transform
- Compute EPI atlas transform

(Show/Hide Details)

- Make atlas transformed EPI anat_ave in 111, 222, and 333 atlas space (*t4img_4dfp*)
- Make cross-realigned atlas-transformed resampled BOLD 4dfp stacks (*t4_xr3d_4dfp*)

1.8.2 fcMRI_preproc

fcMRI preprocessing including nuisance variable regression.

Attention: fcMRI_preproc assumes successful completion of BOLD preprocessing (*cross_bold_pp*).

Usage: fcMRI_preproc_<version>.csh <params file> [instructions file]

Examples:

```
fcMRI_preproc_161012.csh VB16168.params
fcMRI_preproc_090115.csh VB16168.params
```

fcMRI_preproc_140413.csh

Required parameters

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
srcdir	<str>	source directory path (contains run directories)
workdir	<str>	working directory path
TR_vol	<flt>	time per frame (s)
skip	<int>	number of pre-steady state frames
fbolds	<array>	list of bold run folders

Optional parameters

Variable	Values	Default	Description
FCdir	<str>	FCmap	output directory name
anat_aveb	<flt>	10	run_dvar_4dfp preblur in mm (for small voxels, set to 10mm)
anat_avet	<flt>	7	run_dvar_4dfp criterion
FWHM	<int>	6	full-width half maximum for spatial blur
MB	0,1	0	skip slice timing correction and debanding
conc	<str>		pre-existing conc file to use
task_regressor	<str>		optional externally supplied task regressor
noGSR	1	0	suppress global signal (WB) regression

Processing steps

- Create conc file (*conc_4dfp*) and move it to FCdir

- Compute defined mask and apply it (*compute_defined_4dfp*, *maskimg_4dfp*)
- Compute frame censoring (DVARs) (*run_dvar_4dfp*)
- Compute initial sd1 mean (*var_4dfp*, *qnt_4dfp*)
- Make movement regressors for each bold run (*mat2dat*)
- Make whole brain regressors including the 1st derivative (*qnt_4dfp*)
- Make ventricle and bilateral white matter regressors and their derivatives (*qnt_4dfp*)
- Paste nuisance regressors together (including task_regressor if supplied)
- Remove nuisance regressors out of volumetric time series (*glm_4dfp*)
- Temporal bandpass filter with bh = .1 and oh = 2 (*bandpass_4dfp*)
- Spatial blur with f_half = 4.413/\$FWHM (*gauss_4dfp*)

fcMRI_preproc_130715.csh

Params file variables

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
srcdir	<str>	source directory path (contains run directories)
fbolds	<array>	list of bold run folders

Instructions file variables

Variable	Values	Description
FCdir	<str>	output directory name
FSdir	<str>	freesurfer directory containing mri/aparc+aseg.mgz
MB	0,1	skip slice timing correction and debanding
conc	<str>	pre-existing conc file to use
task_regressor	<str>	optional externally supplied task regressor
noGSR	1	suppress global signal (WB) regression
anat_aveb	<flt>	run_dvar_4dfp preblur in mm (for small voxels, set to 10mm)
anat_avet	<flt>	run_dvar_4dfp criterion
CSF_1cube	<int>	cube dimension (in voxels) used by <i>qntv_4dfp</i> for CSF (recommended: 3)
CSF_sd1t	<flt>	threshold CSF sd1 image (recommended: 25)
CSF_svdt	<flt>	limit regressor covariance condition number to (1/{})^2 for CSF (recommended: .2)
WM_1cube	<int>	cube dimension (in voxels) used by <i>qntv_4dfp</i> for WM (recommended: 5)
WM_svdt	<flt>	limit regressor covariance condition number to (1/{})^2 for WM (recommended: .15)
fmtfile	<str>	format file
bps_params	<array>	options for <i>bandpass_4dfp</i> (only -E,M,F specified by default; no default bands)
blur	<flt>	f_half for spatial blur

Processing steps

- Generate FS masks (results in ../atlas) (*Generate_FS_Masks_AZS.csh*)
- Create conc file (*conc_4dfp*) and move it to FCdir
- Compute frame censoring (DVARs) (*run_dvar_4dfp*) and create avg censored image – only if no \$fmtfile specified

- Compute defined mask and apply it (*compute_defined_4dfp*, *maskimg_4dfp*)
- Compute initial sd1 mean (*var_4dfp*, *qnt_4dfp*)
- Make timeseries zero mean (*var_4dfp*)
- Make movement regressors for each bold run (*mat2dat*)
- Temporal bandpass filter using \$bpss_params (*bandpass_4dfp*)
- Make whole brain regressors including the 1st derivative (*qnt_4dfp*)
- Make extra-axial CSF regressors (*maskimg_4dfp*)
- Make venticle movement_regressors (*qntv_4dfp*)
- Make white matter regressors (*qntv_4dfp*)
- Paste nuisance regressors together (including task_regressor if supplied)
- Pass final set of nuisance regressors (omitting WB and WB derivative) through *covariance* -D250
- Remove nuisance regressors out of volumetric time series (*glm_4dfp*)
- Spatial blur with f_half = \$blur if specified (*gauss_4dfp*)

fcMRI_preproc_090115H.csh

Hallquist compliant version of *fcMRI_preproc_090115.csh*

Params file variables

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
srcdir	<str>	source directory path (contains run directories)
workdir	<str>	working directory path
fcbolds	<array>	list of bold run folders

Instructions file variables

Variable	Values	Description
FCdir	<str>	output directory name
MB	0,1	skip slice timing correction and debanding
conc	<str>	pre-existing conc file to use
task_regressor	<str>	optional externally supplied task regressor
noGSR	1	suppress global signal (WB) regression
noWM	1	supress WM regression
movement_regressors	raw, bpss, none	

Processing steps

- Create conc file (*conc_4dfp*) and move it to FCdir
- Compute defined mask and apply it (*compute_defined_4dfp*, *maskimg_4dfp*)
- Compute initial sd1 mean (*var_4dfp*, *qnt_4dfp*)
- Spatial blur with f_half = .735452 (*gauss_4dfp*)
- Temporal bandpass filter with bh = .1 and oh = 2 (*bandpass_4dfp*)

- Make movement regressors for each bold run (*mat2dat*) (if \$movement_regressors is not “none”)
- Make whole brain, ventricle, and bilateral white matter regressors (*qnt_4dfp*)
- Paste nuisance regressors together (including task_regressor if supplied)
- Pass final set of nuisance regressors (omitting WB and WB derivative) through *covariance* -D500
- Remove nuisance regressors out of volumetric time series (*glm_4dfp*)

fcMRI_preproc_090115.csh

Params file variables

Variable	Values	Description
patid	<str>	unique identifier for current subject/session
srcdir	<str>	source directory path (contains run directories)
workdir	<str>	working directory path
fbolds	<array>	list of bold run folders

Instructions file variables

Variable	Values	Description
FCdir	<str>	output directory name
MB	0,1	skip slice timing correction and debanding
conc	<str>	pre-existing conc file to use
task_regressor	<str>	optional externally supplied task regressor
noGSR	1	suppress global signal (WB) regression

Processing steps

- Create conc file (*conc_4dfp*) and move it to FCdir
- Compute defined mask and apply it (*compute_defined_4dfp*, *maskimg_4dfp*)
- Compute initial sd1 mean (*var_4dfp*, *qnt_4dfp*)
- Spatial blur with f_half = .735452 (*gauss_4dfp*)
- Temporal bandpass filter with bh = .1 and oh = 2 (*bandpass_4dfp*)
- Make movement regressors for each bold run (*mat2dat*)
- Make whole brain, ventricle, and bilateral white matter regressors (*qnt_4dfp*)
- Paste nuisance regressors together (including task_regressor if supplied)
- Remove nuisance regressors out of volumetric time series (*glm_4dfp*)

1.8.3 seed_correl

compute multi-volume correlation maps

Attention: *seed_correl* assumes successful completion of BOLD preprocessing (*cross_bold_pp*) and fcMRI preprocessing (*fcMRI_preproc*).

Usage: *seed_correl*<version>.csh <parameters file> [instructions] [options]

`seed_correl_140413.csh`

`seed_correl_130715.csh`

`seed_correl_090115.csh`

1.9 Processing BOLD Data

Scenario: You've collected some BOLD data and you're interested in functional connectivity.

This example will walk you through the typical functional connectivity pipeline in the 4dfp suite - generic BOLD pre-processing, fcMRI pre-processing, and seed-based correlation.

1.9.1 System requirements

These scripts assume that the current shell is `csh`. To check/change your shell you can use the following commands:

```
# output the current shell
echo $0

# switch to csh shell
csh
```

They also expect `REFDIR` to be set as an environment variable, so you will need to add it to your login script:

```
setenv REFDIR /data/petsun43/data1/atlas
```

Additionally, the scripts rely on a couple external programs that will need to be on your system path.

First, check if `mri_convert` and `mcverter` are on your path:

```
which mri_convert
which mcverter
```

If either program is not found, you will need to add the following lines to your login script.

Listing 1: `~/lin.cshrc`

```
set path = ( $path \
             $FREESURFER_HOME/bin \
             /data/nil-bluearc/hershey/unix/software/MRConvert/MRConvert-2.1.0/usr/
             ↪bin \
             )
```

1.9.2 Preparing DICOM data

If you haven't already downloaded your data, see [Downloading data from CNDA](#).

Once you have DICOM data downloaded and transferred to your project directory, you will start by sorting your DICOM data. How to run this will depend on the DICOM directory structure. In the following examples we'll use `NEWT002_s1` for an example MR session, and assume that the DICOM data is under the `SCANS/` directory:

```
$ cd /path/to/project
$ cd NEWT002_s1
$ ls
SCANS
$ ls SCANS
```

If SCANS/ contains a flat list of DICOMs, you will use *dcm_sort*:

```
$ dcm_sort SCANS
```

If SCANS/ contains numbered directories, you will use *pseudo_dcm_sort.csh*:

```
$ pseudo_dcm_sort.csh SCANS
```

This will create study folders for each of the scans downloaded from CNDA, as well as a SCANS.studies.txt file that contains the mapping of study number to series description

```
$ ls
SCANS          study10      study21      study25      study29
SCANS.studies.txt  study14      study23      study27

$ cat SCANS.studies.txt
10  tf13d1_16ns  ABCD_T1w_MPR_vNav          176
14  spc_314ns   ABCD_T2w_SPC_vNav          176
21  epse2d1_90  SpinEchoFieldMap_AP_2p4mm_64s1  3
23  epse2d1_90  SpinEchoFieldMap_PA_2p4mm_64s1  3
25  epfid2d1_90 fMRI_AP_2p4mm_MB4_tr1230_te33  250
27  epfid2d1_90 fMRI_AP_2p4mm_MB4_tr1230_te33  250
29  epfid2d1_90 fMRI_AP_2p4mm_MB4_tr1230_te33  250
```

1.9.3 Generic BOLD pre-processing

Now that we have our DICOM data sorted, we are ready to begin BOLD pre-processing. In the 4dfp suite, this is done via *cross_bold_pp_161012.csh*.

In order to run cross bold, we first need to set up some input files. If you look at the usage for cross bold, it has one required argument and one optional. As mentioned in *Params/Instructions files*, the convention is to use both, putting subject-specific parameters in the params file and study-specific parameters in the instructions file. When creating these files, you'll want to have the list of variables handy. These can be found in the *cross_bold_pp_161012.csh* docs.

The instructions file contains customizations for the processing pipeline in addition to information about the scan sequence. To obtain the scan parameters, you can use *dcm_dump_file*. Since we are looking to process BOLD data, be sure to grab a DICOM from one of the BOLD study folders:

```
$ dcm_dump_file -t study25/NEWT002_s1.MR.head_Hershey.25.173.20161130.131330.19u1n9g.
↪dcm
```

This will print out tags from the DICOM header, including echo time and repetition time. An excerpt is shown here:

```
0018 0023      2 //          ACQ MR Acquisition Type //2D
0018 0024     12 //          ACQ Sequence Name//epfid2d1_90
0018 0025      2 //          ACQ Angio Flag//N
0018 0050     16 //          ACQ Slice Thickness//2.4000000953674
0018 0080      4 //          ACQ Repetition Time//1230
0018 0081      2 //          ACQ Echo Time//33
```

(continues on next page)

(continued from previous page)

```

0018 0083      2 //          ACQ Number of Averages//1
0018 0084     10 //          ACQ Imaging Frequency//123.246868
0018 0085      2 //          ACQ Imaged Nucleus//1H
0018 0086      2 //          ACQ Echo Number//1
0018 0087      2 //          ACQ Magnetic Field Strength//3
0018 0088     16 //          ACQ Spacing Between Slices//2.4000000349655
0018 0089      2 //ACQ Number of Phase Encoding Steps//90

```

Attention: Be sure to pay attention to units. The DICOM header stores times in milliseconds and some cross_bold variables are in seconds.

Some variables don't match a specific tag in the DICOM header and need to be calculated.

- `nx` and `ny`

You will need to grab the 'Img Rows' (0028,0010), 'Img Columns' (0028,0011) and 'NumberOfImagesInMosaic' (0019,100a) tags.

```

$ dcm_dump_file -t study25/NEWT002_s1.MR.head_Hershey.25.173.20161130.
↳131330.19u1n9g.dcm | grep '0028 0010' | awk '{print $8}'
720 # imgRows

$ dcm_dump_file -t study25/NEWT002_s1.MR.head_Hershey.25.173.20161130.
↳131330.19u1n9g.dcm | grep '0028 0011' | awk '{print $8}'
720 # imgColumns

$ dcm_dump_file -t study25/NEWT002_s1.MR.head_Hershey.25.173.20161130.
↳131330.19u1n9g.dcm | grep '0019 100a' | awk '{print $7}'
64 # numImgs

```

With these numbers, you can calculate `nx` and `ny` with the following formulas:

$$nx = imgRows / ceil(sqrt(numImgs))$$

$$ny = imgColumns / ceil(sqrt(numImgs))$$

- `dwell`

Warning: This formula was corrected on 3/26/19. If you used this section previously, you should double-check the value in your instructions file to verify it was calculated correctly.

You will need to grab the 'BandwidthPerPixelPhaseEncode' (0019,1028) tag and `nx` (or `ny`) calculated above.

```

$ strings study25/NEWT002_s1.MR.head_Hershey.25.173.20161130.131330.
↳19u1n9g.dcm | grep BandwidthPer -A 1
BandwidthPerPixelPhaseEncode
18.83200000

```

You can then calculate `dwell` using the following formula, using `nx` for 'MatrixPhase':

$$dwell = 1000 / (BandwidthPerPixelPhaseEncode * MatrixPhase)$$

Tip: For Siemens 3T fMRI, dwell times should be in the range 0.4 - 0.6 ms.

- delta

If you are using a gradient-echo field map (which the current example does not), you will need to calculate delta. To do so, you will need to grab the values of the 'Echo Time' (0018,0081) field from your magnitude field map image.

```
% dcm_dump_file -t /path/to/magnitude/fm/image | grep "0018 0081"
0018 0081      4 //                               ACQ Echo Time//7.38
0018 0081      4 //                               ACQ Echo Time//4.92
```

To get delta, compute the difference of the echo time values.

Tip: For Siemens GRE field map sequences, delta is typically 2.46 ms.

- seqstr

The slice acquisition sequence in multiband fMRI does not follow the old "Siemens_interleave" rule. In this case, the slice sequence depends on the number of slices and the multiband factor to ensure there is no adjacent slice excitation. Siemens now provides an exact listing of slice times in each fMRI DICOM header in the 'MosaicRefAcqTimes' (0019,1029) tag.

In order to correct slice timing for multiband sequences, the slice sequence needs to be identified and passed to *frame_align_4dfp* via the *seqstr* parameter.

AFNI has a function *dicom_hdr* that you can use to extract the slice timing from the header:

```
$ dicom_hdr -slice_times SCANS/25/DICOM/NEWT002_s1.MR.head_Hershey.25.1.
↪20161130.131330.adfigp.dcm
-- Siemens timing (64 entries): 0.0 530.0 1057.5 377.5 907.5 227.5 755.0 ↪
↪75.0 605.0 1135.0 452.5 982.5 302.5 832.5 150.0 680.0 0.0 530.0 1057.5 ↪
↪377.5 907.5 227.5 755.0 75.0 605.0 1135.0 452.5 982.5 302.5 832.5 150.0 ↪
↪680.0 0.0 530.0 1057.5 377.5 907.5 227.5 755.0 75.0 605.0 1135.0 452.5 ↪
↪982.5 302.5 832.5 150.0 680.0 0.0 530.0 1057.5 377.5 907.5 227.5 755.0 ↪
↪75.0 605.0 1135.0 452.5 982.5 302.5 832.5 150.0 680.0
```

We can get the number of bands by counting how many times a slice time is repeated:

```
..code-block::bash

# replace <first_slice_time> before use
$ dicom_hdr -slice_times study25/NEWT002_s1.MR.head_Hershey.25.1.20161130.
↪131330.adfigp.dcm | grep -wo <first_slice_time> | wc -l
4
```

Based on these outputs, we can see that there are 64 slices and a multiband factor of 4. This gives us 16 slices per band. With this information, we can now calculate the slice order for a single band:

```
# replace <num_slice_per_band> before use
$ dicom_hdr -slice_times SCANS/25/DICOM/NEWT002_s1.MR.head_Hershey.25.1.
↪20161130.131330.adfigp.dcm | cut -d ":" -f2 | tr " " "\n" | tail -n
↪<num_slice_per_band> | gawk '{print NR, $1}' | sort -n -k 2,2 | gawk '
↪{printf("%d,", $1);}'
1,8,15,6,13,4,11,2,9,16,7,14,5,12,3,10,
```

Alternatively, you can run `strings` on the header:

```
$ strings SCANS/25/DICOM/NEWT002_s1.MR.head_Hershey.25.1.20161130.131330.
↪adfigp.dcm | grep 'MosaicRefAcqTimes' -A 66
MosaicRefAcqTimes
sGRADSPEC.asGPADData[0].sEddyCompensationX.af1T
0.00000000
530.00000000
1057.50000000
377.50000000
907.50000000
227.50000001
755.00000000
75.00000001
605.00000001
1135.00000001
452.50000001
982.50000001
302.49999999
832.50000002
149.99999999
679.99999999
...
```

You can then copy the slice timing of one band into a file (i.e. `temp.dat`), and run the following:

```
$ cat temp.dat | gawk '{print NR, $1}' | sort -n -k 2,2 | gawk '{printf("
↪%d, ", $1);}'
1, 8, 15, 6, 13, 4, 11, 2, 9, 16, 7, 14, 5, 12, 3, 10,
```

Now that we know how to source information for the instructions file, we'll go ahead and put one together. In this example, we will assume nothing besides `dcm_sort` has already been run on the data and we won't skip any processing steps.

Since we've chosen to set up our instructions file to define study-level params, we'll store it in the project directory.

```
$ cd /path/to/project
$ gedit NEWT_study.params
```

Listing 2: `NEWT_study.params`

```
set inpath = /path/to/project/${patid}
set target = $REFDIR/TRIO_KY_NDC
set go = 1
set sorted = 1
set economy = 0
set epi2atl = 1
set normode = 0

set nx = 90
set ny = 90

set skip = 0

set FDthresh = 0.2
set FDtype = 1
set anat_aveb = 10 # use 10mm preblur (voxel size < 3mm)
```

(continues on next page)

(continued from previous page)

```

set TR_vol = 1.23
set TR_slc = 0 # use default (TR_vol/nslices)
set epidir = 0
set MBfac = 4
set seqstr = 1,8,15,6,13,4,11,2,9,16,7,14,5,12,3,10 # non-standard interleaving
set lomotil = 2 # filter FD in phase-encoding direction

set TE_vol = 33
set dwell = .59
set ped = y-
set rsam_cmd = one_step_resample.csh

```

Our params file, on the other hand, needs to be specified per subject as it contains a mapping to a subject's specific scan numbers. The file outputted by `dcm_sort`, `SCANS.studies.txt`, is a good reference to have handy when creating a subject's params file.

```

$ cd NEWT002_s1
$ cat SCANS.studies.txt
$ gedit NEWT002_s1.params

```

Listing 3: NEWT002_s1.params

```

set patid = NEWT002_s1
set mprs = ( 10 )
set tse = ( 14 )
set irun = ( 1 2 3 )
set fstd = ( 25 27 29 )
set sefm = ( 21 23 )

```

Since our subjects have a T2 image and spin-echo field maps, we specified `tse` and `sefm`, respectively. However, which parameters are specified here will depend on the data you have available. For EPI to atlas registration, you should specify either `tse`, `pdt2`, or neither. For field map correction, you should specify either `sefm` or `gre`.

Now, we run cross bold:

```

$ cross_bold_pp_161012.csh NEWT002_s1.params ../NEWT_study.params

```

Afterwards, you'll have the following subject anf bold directory structures:

```

$ ls
atlas      NEWT002_s1_fmri_unwarp_170616_se.log  SCANS.studies.txt  study23
bold1     NEWT002_s1_one_step_resample.log     sefm                study25
bold2     NEWT002_s1.params                    study10             study27
bold3     NEWT002_s1_xr3d.lst                  study14             study29
movement  SCANS                                study21             unwarp

$ ls bold1
NEWT002_s1_b1.4dfp.hdr          NEWT002_s1_b1_faln_dbnd_r3d_avg_norm.
↪4dfp.ifh
NEWT002_s1_b1.4dfp.ifh        NEWT002_s1_b1_faln_dbnd_r3d_avg_norm.
↪4dfp.img
NEWT002_s1_b1.4dfp.img        NEWT002_s1_b1_faln_dbnd_r3d_avg_norm.
↪4dfp.img.rec
NEWT002_s1_b1.4dfp.img.rec     NEWT002_s1_b1_faln_dbnd_xr3d.mat
NEWT002_s1_b1_faln.4dfp.ifh   NEWT002_s1_b1_faln_dbnd_xr3d_norm.4dfp.
↪hdr

```

(continues on next page)

(continued from previous page)

NEWT002_s1_b1_faln.4dfp.img ↪ ifh	NEWT002_s1_b1_faln_dbnd_xr3d_norm.4dfp.
NEWT002_s1_b1_faln.4dfp.img.rec ↪ img	NEWT002_s1_b1_faln_dbnd_xr3d_norm.4dfp.
NEWT002_s1_b1_faln_dbnd.4dfp.hdr ↪ img.rec	NEWT002_s1_b1_faln_dbnd_xr3d_norm.4dfp.
NEWT002_s1_b1_faln_dbnd.4dfp.ifh	NEWT002_s1_b1_faln_dbnd_xr3d_norm.ddat
NEWT002_s1_b1_faln_dbnd.4dfp.img ↪ 4dfp.hdr	NEWT002_s1_b1_faln_dbnd_xr3d_norm_dsd0.
NEWT002_s1_b1_faln_dbnd.4dfp.img.rec ↪ 4dfp.ifh	NEWT002_s1_b1_faln_dbnd_xr3d_norm_dsd0.
NEWT002_s1_b1_faln_dbnd.dat ↪ 4dfp.img	NEWT002_s1_b1_faln_dbnd_xr3d_norm_dsd0.
NEWT002_s1_b1_faln_dbnd_r3d_avg.4dfp.ifh ↪ 4dfp.img.rec	NEWT002_s1_b1_faln_dbnd_xr3d_norm_dsd0.
NEWT002_s1_b1_faln_dbnd_r3d_avg.4dfp.img ↪ 4dfp.hdr	NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl.
NEWT002_s1_b1_faln_dbnd_r3d_avg.4dfp.img.rec ↪ 4dfp.ifh	NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl.
NEWT002_s1_b1_faln_dbnd_r3d_avg.hist ↪ 4dfp.img	NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl.
NEWT002_s1_b1_faln_dbnd_r3d_avg_norm.4dfp.hdr ↪ 4dfp.img.rec	NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl.

Tip: A lot of files get generated per run and the folders can get cluttered. If you don't intend to use the intermediate files, you should set the economy flag to 5 to remove some of them.

1.9.4 fcMRI pre-processing

After running bold pre-processing, you'll want to run functional connectivity specific processing. However, before we can run *fcMRI_preproc_161012.csh*, there is a prerequisite step of running Freesurfer to generate masks for the subjects which will be used to calculate the nuisance regressors.

If you don't already have a SUBJECTS_DIR for your project, go ahead and make one:

```
$ mkdir /path/to/project/freesurfer
$ setenv SUBJECTS_DIR /path/to/project/freesurfer
```

Next we'll need to get a DICOM from our T1w image to use as our input file for Freesurfer:

```
$ cd /path/to/project/NEWT002_s1
$ cat SCANS.studies.txt | grep T1w
10   tf13d1_16ns   ABCD_T1w_MPR_vNav           176

$ ls SCANS/10/DICOM/*10.1.*
../SCANS/10/DICOM/NEWT002_s1.MR.head_Hershey.10.1.20161130.131330.11drvvd.dcm
```

With this information at hand, we can now launch the Freesurfer job

```
$ at now
at> setenv SUBJECTS_DIR /path/to/project/freesurfer
at> recon-all -all -s NEWT002_s1 -i /path/to/project/NEWT002_s1/SCANS/10/DICOM/
↪NEWT002_s1.MR.head_Hershey.10.1.20161130.131330.11drvvd.dcm
at> <ctrl-d>
```

Same as before, `fcMRI_preproc` accepts a `params` and `instructions` file. If you look at the variable specification for `fcMRI_preproc_161012.csh`, you'll see that it shares some variables with `cross_bold_pp_161012.csh` - we'll leave those the same and simply add in the `fcMRI`-specific ones:

```
$ gedit /path/to/project/NEWT_study.params
```

Listing 4: `NEWT_study.params`

```
# BOLD variables
set inpath = /path/to/project/${patid}
set target = $REFDIR/TRIO_KY_NDC
set go = 1
set sorted = 1
set economy = 0
set epi2atl = 1
set normode = 0

set nx = 90
set ny = 90

set skip = 0

set FDthresh = 0.2
set FDtype = 1
set anat_aveb = 10 # use 10mm preblur (voxel size < 3mm)

set TR_vol = 1.23
set TR_slc = 0 # use default (TR_vol/nslices)
set epidir = 0
set MBfac = 4
set seqstr = 1,8,15,6,13,4,11,2,9,16,7,14,5,12,3,10 # non-standard interleaving
set lomotil = 2 # filter FD in phase-encoding direction

set TE_vol = 33
set dwell = .59
set ped = y-
set rsam_cmnd = one_step_resample.csh

# fcMRI pre-processing
set srcdir = $cwd
set FSdir = /path/to/project/freesurfer/${patid}
set fcbolds = ( ${irun} )
set CSF_lcub = 3
set CSF_sdt = 25
set CSF_svd = .2
set WM_lcub = 5
set WM_svd = .15
set bpss_params = ( -bh0.1 -oh2 )
set blur = .73542
```

No changes are needed to the session `params` file, so now we can run the script:

```
$ fcMRI_preproc_161012.csh NEWT002_s1.params ../NEWT_study.params
```

Afterwards, we will have the following new files:

```
# per run
```

(continues on next page)

(continued from previous page)

```
% ls -tr bold1/*atl_*
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_dsd0.4dfp.img
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_dsd0.4dfp.ifh
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_dsd0.4dfp.hdr
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_dsd0.4dfp.img.rec
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_uout.4dfp.img
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_uout.4dfp.ifh
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_uout.4dfp.hdr
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_uout.4dfp.img.rec
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss.4dfp.img
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss.4dfp.ifh
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss.4dfp.hdr
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss.4dfp.img.rec
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid.4dfp.img
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid.4dfp.ifh
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid.4dfp.hdr
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid.4dfp.img.rec
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid_g7.4dfp.img
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid_g7.4dfp.ifh
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid_g7.4dfp.hdr
NEWT002_s1_b1_faln_dbnd_xr3d_uwrp_atl_bpss_resid_g7.4dfp.img.rec
```

1.9.5 Seed-based correlation

After preprocessing, we can now generate a seed-to-seed correlation matrix for our subject.

If you look at the docs for `seed_correl_161012.csh`, you'll see that we only need to add which regions to analyze (ROIs) to our instructions file.

Here we'll use the canonical ROI list from `$REFDIR` as our input. You can use a different list of ROIS (i.e. BigBrain264, BigBrain305), but there are a few things to be aware of:

- ROIlistfile should contain only a single column

The single column should contain just the ROI file names. If you have additional columns (i.e. listing the coordinates), `paste_4dfp` will misinterpret them and cause the script to error. You can use the following command to create a file with just the first column:

```
cat $ROIlistfile | awk '{print $1}' > ${ROIlistfile}_1col.txt
```

- The correlation matrix will **not** get generated if you have more than 256 ROIs

`covariance` used to only support up to 256 ROIs, so `seed_correl` checks for this and skips the correlation matrix step. While `covariance` has been updated to support more ROIs, `seed_correl` has not. If you are using an ROI list with greater than 256 ROIs, you will run the following commands (after you run `seed_correl`) to get the correlation matrix (and remove intermediate files):

```
# from $FCdir
covariance -uom0 <patid>[_faln_dbnd]_xr3d_uwrp_atl.format <patid>_seed_
↪regressors.dat
/bin/rm *_ROI*_CCR.dat
```

Listing 5: NEWT_study.params

```
# BOLD variables
set inpath = /path/to/project/${patid}
```

(continues on next page)

(continued from previous page)

```

set target = $REFDIR/TRIO_KY_NDC
set go = 1
set sorted = 1
set economy = 0
set epi2atl = 1
set normode = 0

set nx = 90
set ny = 90

set skip = 0

set FDthresh = 0.2
set FDtype = 1
set anat_aveb = 10 # use 10mm preblur (voxel size < 3mm)

set TR_vol = 1.23
set TR_slc = 0 # use default (TR_vol/nslices)
set epidir = 0
set MBfac = 4
set seqstr = 1,8,15,6,13,4,11,2,9,16,7,14,5,12,3,10 # non-standard interleaving
set lomotil = 2 # filter FD in phase-encoding direction

set TE_vol = 33
set dwell = .59
set ped = y-
set rsam_cmnd = one_step_resample.csh

# fMRI pre-processing
set srcdir = $cwd
set FSdir = /path/to/project/freesurfer/${patid}
set fcbolds = ( ${irun} )
set CSF_lcube = 3
set CSF_sdlc = 25
set CSF_svdv = .2
set WM_lcube = 5
set WM_svdv = .15
set bpss_params = ( -bh0.1 -oh2 )
set blur = .73542

# seed_correl ROIs
set ROI_dir = ${REFDIR}/CanonicalROIsNP705
set ROIlistfile = ${REFDIR}/CanonicalROIsNP705/CanonicalROIsNP705.lst

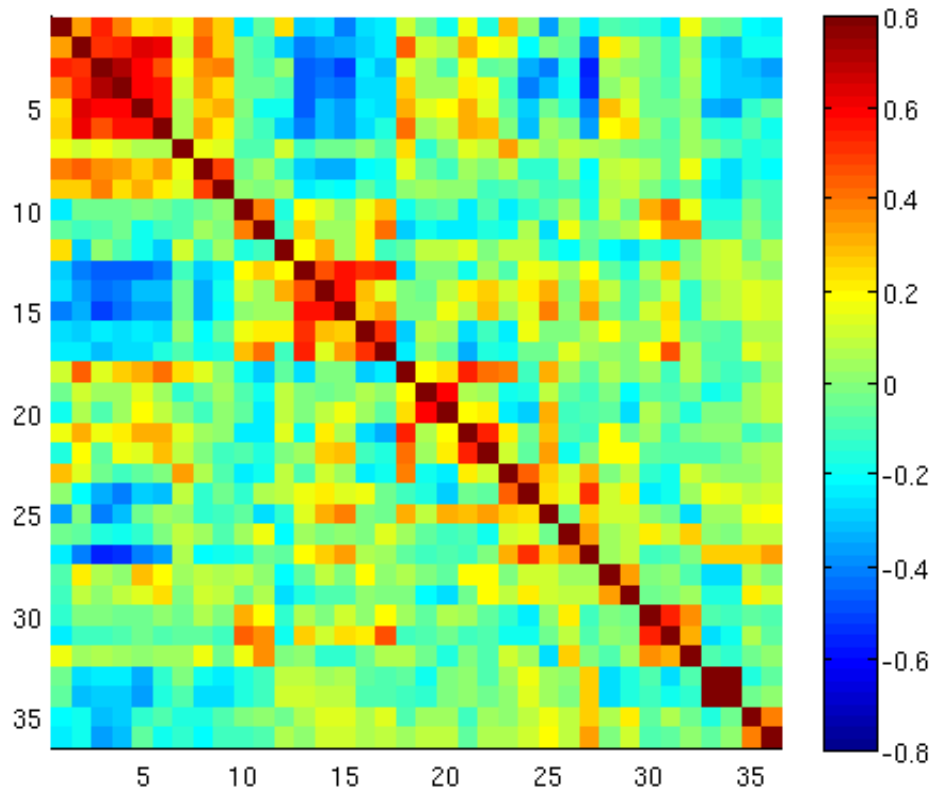
```

Now we can go ahead and run it:

```
$ seed_correl_161012.csh NEWT002_s1.params ../NEWT_study.params
```

This produces a correlation matrix, `$(FCdir)/${patid}_seed_regressors_CCR.dat`.

You can display the matrix with any plotting tool (i.e. `imagesc` in `matlab`, `matplotlib.pyplot.imshow` in `python`).



1.10 Downloading data from CNDA

There are 3 ways to download DICOM data from CNDA.

1.10.1 Manual download of a single subject

You can download DICOM data for a single subject by navigating to a particular subject session and selecting the 'Download' option from the actions menu.



- Home
- New ▾
- Upload ▾
- Tools ▾
- Help ▾

PROJECT: NP919 > SUBJECT: NT710 > NT710

MR Session: NT710

Details	Projects
Accession #: CND_A_E23916 Date Added: 05/05/2011 10:09:17 (jmk1) Date: 05/05/2011 Time: 09:51:28 Operator: jmk Scanner: bay3 Trio SIEMENS TrioTim Acquisition Site: Washington University	Subject: NT710 Gender: Handedness: Age: --

Launch Uploader

- Projects
 - Recent
 - Favorite
 - My projects
 - Other projects
- Stored Searches
- Data

Actions

- Edit
- Download
- View
- View with ImageJ
- Email
- Build
- Add Radiology Read
- Add Manual QC
- Manage Files
- View Images

Scans

Scan	Type	Series Desc	Usability	Files	Note
1	localizer	localizer	questionable	1.7 MB in 3 files	
2	AAScout	AAScout	questionable	23.0 MB in 256 files	
3	T2 weighted	T2 weighted	usable	14.2 MB in 66 files	
4	ep2d_bold_rest	ep2d_bold_rest	usable	49.0 MB in 132 files	
5	MoCoSeries	MoCoSeries	questionable	49.1 MB in 132 files	
6	t1_mpr_ns_sag_ipat	t1_mpr_ns_sag_ipat	usable	56.6 MB in 322 files	
7	ep2d_bold_rest	ep2d_bold_rest	usable	49.0 MB in 132 files	
8	MoCoSeries	MoCoSeries	questionable	49.1 MB in 132 files	
9	t1_mpr_ns_sag_ipat	t1_mpr_ns_sag_ipat	usable	56.6 MB in 322 files	
10	ep2d_bold_rest	ep2d_bold_rest	questionable	3.0 MB in 8 files	
11	MoCoSeries	MoCoSeries	questionable	3.0 MB in 8 files	
12	ep2d_bold_rest	ep2d_bold_rest	usable	49.0 MB in 132 files	
13	MoCoSeries	MoCoSeries	questionable	49.1 MB in 132 files	
99	PhoenixZIPReport	PhoenixZIPReport	questionable	510.6 KB in 9 files	
				Total: 453.0 MB in 1786 files	

This will bring you to a page where you can select individual scans to download for a session.



Home New ▾ Upload ▾ Tools ▾ Help ▾

[PROJECT: NP919](#) > [SUBJECT: NT710](#) > [NT710](#)

Session: NT710

Download Type:

Select All

SCANS

- 1 (localizer)
- 2 (AAScout)
- 3 (T2 weighted)
- 4 (ep2d_bold_rest)
- 5 (MoCoSeries)
- 6 (t1_mpr_ns_sag_ipat)
- 7 (ep2d_bold_rest)
- 8 (MoCoSeries)
- 9 (t1_mpr_ns_sag_ipat)
- 10 (ep2d_bold_rest)
- 11 (MoCoSeries)
- 12 (ep2d_bold_rest)
- 13 (MoCoSeries)
- 99 (Phoenix.ZIPReport)

ADDITIONAL RESOURCES NU_FSLDDMM

MISC FILES



1.10.2 Manual bulk download

Warning: This Java applet will not work in Chrome; run from IE or other browser.

You can bulk download DICOM data for a project by navigating to the project page and selecting ‘Download Images’ from the actions menu.

Launch Uploader

Projects
 + Recent
 + Favorite
 My projects
 Other projects
 + Stored Searches
 + Data

Dopamine buffering capacity measured by pHMRI as a novel biomarker of disease progression in PD

Details Access Manage Pipelines

ID: NP1101 Aka: MPDP
 Keywords: Parkinson,levodopa,pHMRI,hysteresis,PKPD,ASL
 PI: Black, Kevin

Edit Details Delete Manage Custom Variables

Actions

- Add
- Upload Images
- View Prearchive
- Scan Type Cleanup
- Add to Favorites
- Download XML
- Download Images**
- Manage Files

Subjects

Add Tab << < prev 1 next > >>

200 1 of 1 Pgs (9 Rows) Reload Options

Subject	M/F	Hand	DOB	MR Sessions
MPD101				1
MPD102				1
MPD103				1
MPD104				1
MPD105				2
MPD106				1
MPD107				1
MPD109				1
MPD110				2

This will bring you to a page where you can select scan types to download for multiple subjects. The defaults in step 3 should be sufficient.

NP1101

Imaging Data Download

1 Confirm Sessions for download

Toggle All (11)

- MPD101_s1
- MPD102_s1
- MPD103_s1
- MPD104_s1
- MPD105_s1
- MPD105_s2
- MPD106_s1
- MPD107_s1
- MPD109_s1
- MPD110_s1
- MPD110_s2

2 Select image data to download

Scan Formats

- SNAPSHOTS
- DICOM

Toggle All

Scan Types

- AAHScout (20)
- AAHScout_MPR_cor (20)
- AAHScout_MPR_sag (20)
- AAHScout_MPR_tra (20)
- FastIR (1)
- Localizer (17)
- Localizer_aligned (20)
- M0 (1)
- pcal_3D_tgse_3mm_SEG2by2 (77)
- Perfusion_Weighted (76)
- PhoenixZIPReport (11)
- T1 (1)
- T1w_MPR_vNav_4e (18)
- T1w_MPR_vNav_4e RMS (18)
- T1w_setter (18)
- T2w_setter (18)
- T2w_SPC_vNav (18)

3 Select download format

Option 1. Direct download

You will be prompted to enter a local directory to copy the data to. Then, all of the requested imaging data will be directly downloaded via a java applet into that directory.

Option 2. Catalog XML (.xcat)

Download an xml representation of the files. This xml can then be used by specific applications to download the data at the user's command.

- Include project in file paths
- Include subject in file paths
- Simplify downloaded archive structure

4 Submit Data Request

Submit

Upon page submission, the pertinent files will be reviewed and organized for download. Depending on the number of files, this may take several minutes to process.

The next screen will show a pop-up asking for permission to run the Java applet, choose 'Run'.

The screenshot shows the CNDA web interface. At the top left is the CNDA logo. To its right is a search bar with 'Search' and 'Advanced' buttons. Below the logo is a navigation menu with 'Home', 'New', 'Upload', 'Tools', and 'Help'. On the left side, there is a 'Launch Uploader' button and a tree view containing 'Projects' (with sub-items: Recent, Favorite, My projects, Other projects), 'Stored Searches', and 'Data'. The main content area is titled 'Download Images' and features a large Java logo. A security warning dialog box is overlaid on the right, with the title 'Do you want to run this application?'. The dialog contains the following information:

- Name: XNAT
- Publisher: Washington University
- Location: https://cnda.wustl.edu

 Below this information, it states: 'This application will run with unrestricted access which may put your computer and personal information at risk. Run this application only if you trust the location and publisher above.' There is a checkbox for 'Do not show this again for apps from the publisher and location above'. At the bottom of the dialog are 'More Information' (with an info icon), 'Run', and 'Cancel' buttons.

You will then choose a local folder to store the downloaded data before selecting 'Start' to begin the download.

This screenshot shows the same CNDA web interface as above, but the security dialog is closed. The 'Download Images' section now contains two input fields:

- 'Select local folder:' with a text input field and a 'Browse...' button.
- 'Overall progress:' with a text input field and a 'Start' button.

 The rest of the interface, including the navigation menu and the left sidebar, remains the same.

1.10.3 API download

Instructions for using the XNAT REST API can be found on the [XNAT wiki](#)

1.11 Creating a Freesurfer ROI Mask

This example will walk you through how to transform your Freesurfer segmentation into atlas space and extract specific anatomical regions.

1.11.1 Transforming a Freesurfer segmentation to atlas space

If you ran recon-all using a native-space T1 image and did not specify a custom atlas, you will need to transform your Freesurfer segmentation to your atlas space. The command to do this is *freesurfer2mpr_4dfp*.

Note: If you have run *fcMRI_preproc_161012.csh* on your subjects, you can skip this step as it has already been done.

As shown in the usage information for the script, we will need to supply the Freesurfer `orig` and `aparc+aseg` files in 4dfp format. Since Freesurfer outputs images in mgz format, we will first need to convert the required images to 4dfp. There is no direct conversion from mgz to 4dfp, so we will use nifti as an intermediate format.

```
$ cd /path/to/project
$ cd NEWT002_s1/atlas

$ mri_convert /path/to/fsdir/NEWT002_s1/mri/orig.mgz NEWT002_s1_orig.nii --out_
↪orientation RAS
$ nifti_4dfp -4 NEWT002_s1_orig.nii NEWT002_s1_orig.4dfp.img

$ mri_convert /path/to/fsdir/NEWT002_s1/mri/aparc+aseg.mgz NEWT002_s1_aparc+aseg.nii -
↪-out_orientation RAS
$ nifti_4dfp -4 NEWT002_s1_aparc+aseg.nii NEWT002_s1_aparc+aseg.4dfp.img
```

The script relies on an existing transform from the subject's MPRAGE to the target atlas. Because you don't supply the script with a t4 file, it looks in the atlas directory for a t4 file of the form `<(4dfp) mpr>_to_<target>_t4`, where `<(4dfp) mpr>` is the MPRAGE supplied as the first argument (sans the 4dfp extension) and `<target>` is the value supplied to the `-T` flag.

From the subject's atlas directory, we can now run the script - making sure to use the MPRAGE and atlas that match the existing t4.

```
$ ls *mpr*to*_t4
NEWT002_s1_mpr1_to_TRIO_KY_NDC_t4

$ freesurfer2mpr_4dfp NEWT002_s1_mpr1.4dfp.img NEWT002_s1_orig.4dfp.img -TTRIO_KY_NDC
```

When it's done, you should have the atlas-aligned `aparc+aseg` in 111, 222, and 333 space in your atlas folder.

```
$ ls *aparc+aseg_on*.4dfp.img
NEWT002_s1_aparc+aseg_on_TRIO_KY_NDC_111.4dfp.img
NEWT002_s1_aparc+aseg_on_TRIO_KY_NDC_222.4dfp.img
NEWT002_s1_aparc+aseg_on_TRIO_KY_NDC_333.4dfp.img
```

1.11.2 Creating ROI masks

Once you have the atlas-aligned segmentation image, you can create region masks by masking the image using the Freesurfer-assigned region value (which can be found in the [Freesurfer lookup table](#)).

To extract a particular region, we would use *zero_ltgt_4dfp* to zero out all the voxels that are not equal to that region's value. If we wanted the left Amygdala, we would use the following:

```
$ zero_ltgt_4dfp 18to18 NEWT002_s1_aparc+aseg_on_TRIO_KY_NDC_333 NEWT002_s1_lAmygdala
```

If we wanted a binary mask of both the left and right Amygdala, then we would need to follow the steps above again, this time using the value for the right Amygdala. Then, we would need to combine them together using the add operation of *imgopr_4dfp*. Finally, we would use *masking_4dfp* with the `-v1` flag to binarize the values.

```
$ zero_ltgt_4dfp 54to54 NEWT002_s1_aparc+aseg_on_TRIO_KY_NDC_333 NEWT002_s1_rAmygdala
$ imgopr_4dfp -aNEWT002_s1_amygdala NEWT002_s1_lAmygdala NEWT002_s1_rAmygdala
$ masking_4dfp -v1 NEWT002_s1_amygdala NEWT002_s1_amygdala NEWT002_s1_amygdala_msk
```

1.12 Dicom utilities

1.12.1 dcm_dump_file

dump dicom header info to stdout

Usage: `dcm_dump_file [-b] [-g] [-l] [-m mult] [-t] [-v] [-w flag] [-z] file [file ...]`

Options

<code>-b</code>	Input files are stored in big-endian byte order
<code>-e</code>	Exit on file open error. Do not process other files
<code>-g</code>	Remove group length elements
<code>-l</code>	Use (retired) length-to-end attribute for object length
<code>-m mult</code>	Change VM limit from 0 to mult
<code>-t</code>	Part 10 file
<code>-v</code>	Place DCM facility in verbose mode
<code>-w</code>	Set open options; flag can be REPEAT
<code>-z</code>	Perform format conversion (verification) on data in files

1.13 DTI

1.13.1 dwi_xalign3d_4dfp

motion compensation for dwi data (single run)

Usage: `dwi_xalign3d_4dfp <(4dfp) dwi> <(4dfp) mask>`

Examples:

```
dwi_xalign3d_4dfp hbo08a_dwi1 hbo08a_dwi1_mskt -s -g2-4 -g5,13,18,23
```

Options

-p	planar (2D; disable cross-slice) alignment
-w	enable wrap addressing
-s	enable cross DWI voxel size adjust (principal axis stretch)
-a	compute group arithmetic mean volume (default geometric mean)
-n	zero negative values in output image
-I<int>	specify volume number of I0 counting from 1 (default 1)
-f<flt>	specify pre-blur filter half freq (1/mm) (default none)
-d<flt>	specify sampling interval in mm (default=5.0000)
-i<flt>	specify displacement search radius in mm (default=3.0000)
-j<flt>	specify parameter search object radius in mm (default=40.0000)
-c<int>	specify number of within-group cycles (default=3)
-g<int>[-<int>][,<int>[-<int>]][,..]	program alignment group

N.B.: <(4dfp) mask> may be “none” N.B.: I0 should not be named in any programmed alignment group

1.13.2 dwi_cross_xalign3d_4dfp

cross-run motion compensation and averaging of dwi data

Usage: dwi_cross_xalign3d_4dfp <(4dfp) dwi1> <(4dfp) dwi2> <(4dfp) dwi3> ... <(4dfp) dwi_out>

Examples:

```
dwi_cross_xalign3d_4dfp -sgmjo_sub2-dwi1_mskt jo_sub2-dwi1 jo_sub2-dwi2 jo_sub2-dwi_
→all
dwi_cross_xalign3d_4dfp -sgmjo_sub2-dwi1_mskt -ljo_sub2_dwi.lst jo_sub2-dwi_all
```

Options

-p	planar (2D; disable cross-slice) alignment
-w	enable wrap addressing
-s	enable cross DWI voxel size adjust (principal axis stretch)
-n	zero negative values in output image
-z<x y z><flt>	zoom output x y or z dimension by specified factor
-g	use group geometric mean (*_geom) volumes for cross-run registration
-a	append successive runs in output (default average)
-m<(4dfp) mask>	specify first volume mask
-I<int>	specify volume number of I0 counting from 1 (default 1)
-f<flt>	specify pre-blur filter half freq (1/mm) (default none)
-d<flt>	specify sampling interval in mm (default=5.0000)
-i<flt>	specify displacement search radius in mm (default=3.0000)
-j<flt>	specify parameter search object radius in mm (default=40.0000)
-l<lst>	read input file names from specified file (use before naming output)
-@<bll>	output big or little endian (default input endian)

N.B.: option -I (non-default I0 volume) must be matched according to use of option -g

1.13.3 diff_4dfp

diffusion tensor computation given dwi

Usage: diff_4dfp <prm_file> <file_4dfp> <opt mask_file> <opt CO_file>

Examples:

```
diff_4dfp tp7_params.dat /data/emotion/data3/track_sub3/track_sub3_DTI_avg
```

Options

Computational	
-N	compute D using nonlinear Levenberg-Marquardt (default log linear LS)
-Z	use nonlinear approach to repair bad voxels from log linear LS
-c	estimate non-mobile diffusion term (applies only to Levenberg-Marquardt algorithm)
-s<int>	compute only selected slice for debugging
-v<int>	compute only selected voxel for debugging
-B<flt>	ignore bad encoding at specified threshold (units = s.d.) (default=3.0)
-b<flt>	ignore encodings with noisy background at specified threshold (default=3.0)
-S<flt>	subtract a fraction of S0 image from data (def=0.1), not compatible with B,b
-C	subtract a CO fraction from data using an imported file, not compatible with B,b
-G<int> Correct tbi data, =1 CCIR remove encode 1, =2 SLCH remove encode 10,22	
Masking	
-m	Use external mask included as third input file
-M	compute threshold mask without holes
-t<flt>	specify mask threshold as fraction of I0 mode (default=0.1000)
-h<flt>	specify minimum I0 mode (default=100.00)
-n<int> specify number of I0 histogram smoothings (default=4)	
Output	
-a<str>	append specified trailer to output fileroots
-p	print out pixel numbers for debugging
-D	output D tensor
-F	output FA (fractional anisotropy)
-E	output eigenvalues
-V[int]	output [specified number of (default=1)] eigenvectors (principal first)
-P	output prolaticity
-R	output single residue volume for model
-r	output squared residue values for all encodings in a separate file
-o	output extra full LM output files (applies only to LM algorithm) (implies -N)
-d	debug mode, provide extra volume of output as needed
-@<blt>	output big or little endian (default input endian)

N.B.: the first data volume must have high SNR from b=0 or low b value N.B.: optional output volumes are appended to MD and RA N.B.: output order: MD,RA,(Dxx,Dyy,Dzz,Dxy,Dxz,Dyz),(FA),(E123,RD),(CO),(Res),(Evecs),(Prol) N.B.: -b and -B are independent but can both be applied N.B.: -b requires -m and mask dimensions must match image dimensions N.B.: -B, -b parameter useful range is 1.5 to 3 N.B.: eigenvalue ordering is = Eval1 < Eval2 < Eval3 N.B.: -c produces an fractional constant output $CO = C/(C+S0)$

1.13.4 diffRGB_4dfp

dwi → RGB map

Usage: diffRGB_4dfp <prm_file> <file_4dfp>

Examples:

```
diffRGB_4dfp -t0.5 -qc1.7 tp7_params.dat /data/DTI_avg
```

Options

-q	scale intensity by sqrt(Asig) instead of Asig
-G	change color coding to bgr (default rgb)
-c<flt>	specify the intensity scale value (default=1.0000)
-t<flt>	specify mask threshold as fraction of I0 mode (default=0.1000)
-T<str>	specify t4 file used to transform DWI data
-h<flt>	specify minimum I0 mode (default=100.00)
-n<int>	specify number of I0 histogram smoothings (default=4)
-D	input <file_4dfp> is 8 volume diff_4dfp -D output (Dbar, Asigma, D tensor)
-@<bl>	output big or little endian (default input endian)

N.B.: <prm_file> is ignored with -D option

1.13.5 whisker_4dfp

dwi → whiskers (visualized in Matlab)

Usage: whisker_4dfp <prm_file> <file_4dfp>

Examples:

```
whisker_4dfp tp7_params.dat -dz3 /data/emotion/data3/track_sub3/track_sub3_DTI_avg
```

Options

-h<flt>	specify minimum I0 mode (default=100.00)
-n<int>	specify number of I0 histogram smoothings (default=4)
-t<flt>	specify mask threshold as fraction of I0 mode (default=0.1000)
-T<str>	specify t4 file used to transform DWI data
-E	additionally output eigenvalues
-3	output 3 eigenvectors scaled by eigenvalue
-d<xlylz><int>	specify quiver spacing in pixels (default=1)

N.B.: default output is first two eigenvectors scaled by Asigma

1.14 Evaluate and ROI-oriented programs

1.14.1 peak_4dfp

locate and consolidate maxima to generate ROI

Usage: peak_4dfp <file_4dfp>

Examples:

```
peak_4dfp grand_average_222[.4dfp.img] -s10
```

Options

-s<flt>	preblur with hard sphere kernel of specified radius (invokes hsphere_4dfp)
-n<int>	limit initial pos and neg peak list lengths (default=1000)
-c<flt>[to<flt>]	specify sign inverted curvature thresholds (default none)
-v<flt>[to<flt>]	specify peak value thresholds (default none)
-d<flt>	consolidate extremum pairs closer than specified distance
-o<flt>	output a fidl compatible 4dfp format ROI file with regions of specified radius
-m<str>	apply named mask file to output ROIs
-N<int>	specify output ROI minimum voxel count (default = 1)
-a<str>	append specified string to ROI output filename
-q	quiet mode (suppress rec file listing)
-F	force preblur image creation even if hsphere_4dfp result exists (no effect without -s<flt>)
-@<bll>	output big or little endian (default input endian)

N.B.: operations controlled by options -s, -n, -c, -v, -d, -o, -m, -N are applied serially in listed order N.B.: all distances are in mm

1.14.2 read_4dfp

report value of image at specified real coordinate

Usage: read_4dfp <flt x0> <flt y0> <flt z0> <4dfp imgroot> [options]

Examples:

```
read_4dfp 33.1 -56.2 18. grand_average_222[.4dfp.img]
```

Options

-v	verbose mode
----	--------------

1.14.3 imgmax_4dfp

report maximum and minimum values

Usage: imgmax_4dfp <my_image[.4dfp.img]>

Options

-m	report min as well as max
-e	report max/min values in scientific notation
-r	report root sum of squares
-v	verbose (time series) mode

1.14.4 img_hist_4dfp

construct voxel value histogram; evaluate moments

Usage: img_hist_4dfp <(4dfp) image>

Options

-b<int>	specify number of bins (default = 100)
-f<int>	select volume (counting from 1) of 4dfp stack (default analyze all volumes)
-t<flt>	specify image intensity threshold
-r<flt>[to<flt>]	specify histogram range
-m<(4dfp) mask>	mask input using (non-zero voxels of) specified mask (only first frame used)
-h	create <image>.hist file suitable for plotting, e.g., with xmgr
-p	create <image>.dat file suitable for input to numerical procedures
-x	create <image>.xtile percentile listing
-C	create output files in \$cwd (default parallel to <(4dfp) image>)
-u	normalize output .hist and .dat distributions to unit area
-s	report moments

N.B.: option -f causes selected volume to be reported in filename of -{hpx} created files

1.14.5 qnt_4dfp

report mean value within 3D ROI

Usage: qnt_4dfp <(4dfp)|(conc) image> <(4dfp) mask>

Examples:

```
qnt_4dfp -t23.2 va1234_mpr mask
```

Options

-s	time series mode
-d	include backwards differences (differentiated signal) in output (requires -f or -F, implies -s)
-D	count only defined (finite, non 0.0, non-NaN, non 1.e-37) <image> voxels
-A	apply threshold test to absolute value of <mask>
-W	interpret <mask> as spatial weights (negative values allowed) (disables mask threshold testing)
-v<flt>[to<flt>]	count only <image> voxels within specified range
-f<str>	specify frames to count format, e.g., "4x120+4x76+"
-F<str>	read frames-to-count format from specified file
-p<flt>	specify mask threshold as percent of <mask> max
-t<flt>	specify absolute <mask> threshold (default = 0.0)
-c<flt>	scale output mean values by specified constant (default = 1.0)

N.B.: only the first frame of <mask> is used N.B.: <image> and <mask> may be the same N.B.: conc files must have extension "conc"

1.14.6 qntm_4dfp

evaluate multiple volumes in multiple ROIs

Usage: qntm_4dfp <(4dfp)|(conc) image> <(4dfp) ROI>

Examples:

```
qntm_4dfp TC30274_rmshp_faln_dbnd_xr3d_at1.conc iter10_roi_-02_-37_+27m_ROI
```


Options

-Z	count zero voxels in <image> as defined
-V	force code_by_volume even if the number of volumes is 1
-N	create ROIs/voxel image
-o<str>	write output to specified text file (default stdout)
-h	suppress printing output header

N.B.: conc files must have extension “conc” N.B.: only defined voxels (not 0.0 and not NaN and not 1.e-37 and finite) are counted N.B.: <(4dfp) ROI> may either a value-coded single volume ROI image or a multi-volume mask N.B.: <(4dfp) ROI> coded values are integerized N.B.: qntm_4dfp ignores <(4dfp) ROI> ifh center and mmpix fields

1.14.7 qntv_4dfp

evaluate multiple volumes in ROI subdivided into cubes

Usage: qntv_4dfp <(4dfp)|(conc) image> <(4dfp) ROI>

Examples:

```
qntv_4dfp TC30274_rmisp_faln_dbnd_xr3d_atl.conc iter10_roi_-02_-37_+27m_ROI
```

Options

-H	include header info in output
-V	print defined voxel counts per die
-D	create die image (voxels >= ncrit)
-K	create die (voxel) coordinate listing
-Z	count zero voxels in <image> as defined
-O<int>	select output type (see below)
-f<str>	specify frames-to-count format (default count all frames)
-F<str>	read frames-to-count format from specified file (supersedes option -f)
-l<int>	specify length of die in voxels (default 1)
-n<int>	specify minimum die voxel count (default 1)
-t<flt>	specify svd output tolerance - ratio of least to greatest eigenvalue (default 1e-06)
-o<str>	write output to specified text file (default stdout)

-O<int> options

1	timeseries directly extracted from dice
2	timeseries extracted from dice with mean removed
3	die timeseries passed through svd multiplied by eigenvalue
4	die timeseries passed through svd (unit variance)

N.B.: conc files must have extension “conc” N.B.: only defined voxels (not 0.0 and not NaN and not 1.e-37 and finite) are counted N.B.: qntv_4dfp ignores <(4dfp) ROI> ifh center and mmpix fields N.B.: to obtain a GLM condition number = X specify $\sqrt{1/X}$ as tol with option -t

1.14.8 qntw_4dfp

evaluate multiple volumes using weighted ROI

Usage: qntw_4dfp <(4dfp)|(conc) image> <(4dfp) ROI>

Examples:

```
qntw_4dfp TC30274_rmsp_faln_dbnd_xr3d_at1.conc iter10_roi_-02_-37_+27m_ROI
```

Options

-L<int>	specify ROI weight L-norm (default = 0)
-o<str>	write output to specified text file (default stdout)
-Z	count zero voxels in <image> as defined
-H	include header info in output

N.B.: conc files must have extension “conc” N.B.: <(4dfp) ROI> is interpreted as a multi-volume voxel-wise set of weights N.B.: only defined voxels (not 0.0 and not NaN and not 1.e-37 and finite) are counted N.B.: qntw_4dfp ignores <(4dfp) ROI> ifh center and mmppix fields

1.14.9 var_4dfp

evaluate variance or s.d. about mean over timeseries

Usage: var_4dfp <(4dfp|conc) input>

Examples:

```
var_4dfp -sn3 -c10 test_b1_rmsp_dbnd
```

Options

-d	debug mode
-m	remove mean volume from stack
-s	compute s.d. about mean
-G	compute mean ignoring run boundaries (default within runs)
-v	compute variance about mean (default operation)
-z	output logical and of all defined voxels
-n<int>	specify number of pre-functional frames per run (default = 0)
-f<str>	specify frames to count format, e.g., “4x120+4x76+” (overrides -n)
-F<str>	read format from specified file
-c<flt>	scale output image values by specified factor
-N	output undefined voxels as NaN
-Z	output undefined voxels as 0
-E	output undefined voxels as 1.e-37 (default)
-@<bl>	output big or little endian (default input endian)

N.B.: input conc files must have extension “conc” N.B.: identically zero input voxels are counted as defined N.B.: options {-v -s -m -z} are mutually exclusive N.B.: absent -G voxelwise mean is individually computed over each run in conc N.B.: -f option overrides -n

1.14.10 dvar_4dfp

evaluate variance or s.d. about mean over differentiated timeseries

Usage: dvar_4dfp [options] <stack_4dfp>

Examples:

```
dvar_4dfp -n3 test_b1_rmsp_dbnd -mtest_anat_ave_mskt
```

Options

-m<(4dfp) mask>	use specified 4dfp mask
-n<int>	specify number of pre-functional (anatomy) frames
-t<flt>	specify maskfile threshold (default = 0.0)
-b<flt>	specify preblur FWHM in mm (default none)
-s	output sqrt(dvar) (default dvar)
-@<bll>	output big or little endian (default input endian)

1.14.11 burn_sphere_4dfp

“burn in” sphere at specified real coordinates

Usage: burn_sphere_4dfp <flt x0> <flt y0> <flt z0> <4dfp imgroot> <4dfp outroot> [options]

Examples:

```
burn_sphere_4dfp 33.1 -56.2 18. grand_average_222[.4dfp.img] -v2 -o7.5
burn_sphere_4dfp 33.1 -56.2 18. 222 -v2 -o7.5
```

Options

-a	superimpose sphere on image (default duplicate input format with zero background)
-s	sum overlapping spheres (default overwrite)
-v<flt>	specify burn in value (default=1.0000)
-o<flt>	specify sphere radius in mm (default=6.0000) (radius of 0 creates single pixel burn)
-l<lst>	read sphere coordinates from specified list (command line coords ignored)
-@<bll>	output big or little endian (default input endian)

N.B.: without -a only the input ifh (or standard atlas string) is required N.B.: specifying <4dfp imgroot> as “333[.n]” “222” or “111” generates standard atlas space output N.B.: if the 4dfp image does not exist the default output endianness is CPU endian

1.14.12 ROI_resolve_4dfp

resolve a set of possibly overlapping ROIs into a disjoint set

Usage: ROI_resolve_4dfp <(4dfp) ROI1> <(4dfp) ROI2> <(4dfp) ROI3> ...

Options

-l<lst>	read input file names from specified list file
-@<bll>	output big or little endian (default CPU endian)

N.B.: output 4dfp fileroots are same as inputs with appended “z”

1.14.13 imgsurf_4dfp

move ROI coordinates to nearest surface

Usage: imgsurf_4dfp <(4dfp) image> <point_list>

N.B.: <point_list> lists loci in atlas coordinates (X Y Z) in mm

1.14.14 spatial_corr_4dfp

compute image similarity as correlation over space

Usage: spatial_corr_4dfp <image_x> <mask_x> <image_y> <mask_y> [output_text_file]

Options

-C	compute covariance (default correlation)
-M	suppress removal of patial means
-c<flt>	scale output covariance matrix values by specified factor

N.B.: image dimensions must match N.B.: spatial_corr_4dfp counts only defined (not NaN or 1.e-37 or 0.0) voxels

1.14.15 spatial_cov_multivol_4dfp

compute volume-pair covariance over space

Usage: spatial_cov_multivol_4dfp <(4dfp) image> <(4dfp) mask>

Options

-Z	compute covariance with respect to zero (default wrt image mean)
-p<int>	generate specified number of PCs (default none)
-c<flt>	scale text output covariance matrix values by specified factor

N.B.: spatial_cov_multivol_4dfp counts only defined (not NaN or 1.e-37) voxels N.B.: zero voxels are counted as defined in <(4dfp) image> in cov computation N.B.: zero voxels are counted as undefined in <(4dfp) image> in <(4dfp) mask> N.B.: all zero or all undefined <(4dfp) image> volumes are ignored

1.15 Filter in space

1.15.1 gauss_4dfp

spatial frequency domain

Usage: gauss_4dfp <4dfplconc input> f_half [outroot]

Examples:

```
gauss_4dfp pt349_study9to9 0.1
gauss_4dfp p1234ho5 0.7 p1234ho5_g7
```

Options

outroot	output file name (default = <inroot>_g<10*f_half>) (only for 4dfp input)
-@<bll>	output big or little endian (default input endian)
-w	(wrap) suppress x and y padding
-d	differentiate

N.B.: f_half is half frequency in 1/cm N.B.: $\text{FWHM} * f_{\text{half}} = (2 \ln 2 / \pi) = 0.4412712$ N.B.: conc files must have extension "conc"

1.15.2 imgblur_4dfp

spatial domain

Usage: imgblur_4dfp [options] <image_file> <FWHM_in_mm>

Examples:

```
imgblur_4dfp -yz vc345 5.5
```

Options

-x	selective x blur
-y	selective y blur
-z	selective z blur
-@<bll>	output big or little endian (default input endian)

N.B.: default blur is 3D isotropic

1.15.3 hsphere_4dfp

convolve image with hard sphere kernel

Usage: hsphere_4dfp <file_4dfp> <flt>

Examples:

```
hsphere_4dfp np1234_zmap_222 5
```

N.B.: second argument specifies hard sphere radius in mm

1.16 Filter in time

1.16.1 bandpass_4dfp

Independent specification of low and high ends; remove linear trends; remove DC

Usage: bandpass_4dfp <(4dfplconc) input> <TR_vol>

Examples:

```
bandpass_4dfp qst1_b1_rmsp_dbnd_xr3d[.4dfp.img] 2.36 -b10.01 -o11 -bh0.15 -oh2
```

Options

-b[llh]<flt>	specify low end or high end half frequency in hz
-o[llh]<int>	specify low end or high end Butterworth filter order
-n<int>	specify number of pre-functional frames (default = 0)
-I<int>	specify interpolation mode (0 = none; 1 = linear; 2 = cubic spline) (default = 1)
-f<string>	specify frames-to-count format (overrides option -n)
-F<string>	read frames-to-count format from specified file (overrides options -n and -f)
-t<str>	change output filename trailer (default=" _bps")
-a	retain DC (constant) component
-r	retain linear trend
-E	code undefined voxels as 1.e-37
-M	disable undefined mask computation
-B	compute gain using correct Butterworth formula (default squared Butterworth gain)
-@<bl>	output big or little endian (default input endian)

N.B.: undefined values are zero, NaN, or 1.e-37 N.B.: input conc files must have extension "conc" N.B.: omitting low end order specification disables high pass component N.B.: omitting high end order specification disables low pass component

1.17 fMRI oriented programs

1.17.1 compute_defined_4dfp

generate mask of voxels defined over all frames

Usage: compute_defined_4dfp <4dfplconc input>

Options

-z	count zero voxels as undefined (default defined)
-f<str>	specify frames-to-count format (default count all)
-F<str>	read frames-to-count format from specified file

1.17.2 cs2ap_4dfp

Converts cosine and sine amplitude images to amplitude and phase. Primarily used for phase-encoded retinotopy.

Usage: cs2ap_4dfp <(4dfp) cos_img> <(4dfp) sin_img> <(4dfp) outroot>

Options

-t<flt>	specify amplitude threshold for phase map (default = 0.0000)
-w<flt>	specify pre-blur FWHM in mm (default = 0.0000)
-@<bl>	output big or little endian (default input endian)

1.17.3 normalize_4dfp

scale to achieve mode 1000

Usage: normalize_4dfp <(4dfp) image>

Examples:

```
normalize_4dfp -n3 my_run_4dfp
normalize_4dfp -n3 -v2 my_run_4dfp
```

Options

-n<int>	specify number of pre-functional frames
-v0	no frame to frame intensity stabilization
-v1	volume based frame to frame intensity stabilization (default)
-v2	slice based frame to frame intensity stabilization
-s	disable mode=1000 normalization
-z	subtract mean volume from functional frames
-h	create <image>.hist file suitable for plotting, e.g., with xmgr
-a<str>	specify trailer (default="norm")
-m<str>	read specified 4dfp mask (default blur & threshold input image)
-@<bll>	output big or little endian (default input endian)

1.17.4 deband_4dfp

correct systematic odd vs. even slice intensity banding

Usage: deband_4dfp <(4dfp) image>

Examples:

```
deband_4dfp -n3 mybold
deband_4dfp -F"3x125+" mybold
```

Options

-e	deband by exponential gradient model (default flat model)
-g	deband by linear gradient model (default flat model)
-n<int>	specify number of pre-functional frames
-F<str>	specify complete functional/non-functional format
-@<bll>	output big or little endian (default input endian)

1.17.5 rmspike_4dfp

remove artifact due to k-space DC offset¹

Usage: rmspike_4dfp <file_4dfp>

Examples:

```
rmspike_4dfp -n3 -x33 test_b1.4dfp.img
rmspike_4dfp -x33 -F"45(1x6+)" test_b1
```

Options

¹ Only needed for older sequences

-n<int>	specify number of anatomy frames
-x<int>	restrict search to specified column
-y<int>	restrict search to specified row
-F<str>	specify whole run functional/non-functional format
-@<bl>	output big or little endian (default input endian)

1.17.6 cross_realign3d_4dfp

motion correct fMRI timeseries within and across runs

Usage: cross_realign3d_4dfp -l<4dfp_list_file> or: cross_realign3d_4dfp <run1_4dfp> <run2_4dfp> ...

Examples:

```
cross_realign3d_4dfp run1_4dfp run2_4dfp run3_4dfp
cross_realign3d_4dfp -sqwv -lruns_4dfp.lst
cross_realign3d_4dfp -pwqsf -n3 -lruns_4dfp.lst
```

Options

-d	debug mode
-@<bl>	output big or little endian (default CPU endian)
-f	force recomputing even if output files exist
-g	enable linear intensity gradient compensation
-c	use cross-modal registration always
-l<str>	specify list file of 4dfp filenames
-m<str>	specify 4dfp mask to be applied to all runs (default compute)
-n<int>	specify number of pre-functional frames
-b<flt>	specify pre-blur in reciprocal mm (default=0.06)
-p	2D (planar) realignment (default 3D)
-q	minimize status reporting
-r<int>	specify non-default reference frame
-s	enable stretch
-v[01]	disable/enable per frame intensity normalization (default disabled)
-w	enable wrap addressing
-Z	output undefined voxels as 0.0 (default 1.0e-37)
-R	disable resampling

1.17.7 t4_xr3d_4dfp

motion correct and resample in atlas space in one step

Usage: t4_xr3d_4dfp [options] <t4file> <input_4dfp_stack>

Examples:

```
t4_xr3d_4dfp -aat1 anat_ave_to_711-2B_t4 b1_rmsp_dbnd
```

Options

-a<str>	specify outfile name trailer (default = "xr3d")
-c<flt>	scale output by specified factor
-N	output undefined voxels as NaN
-Z	output undefined voxels as 0
-E	output undefined voxels as 1.e-37 (default)
-v[0 1]	set per frame intensity equalization mode (default = OFF)
-@<bl>	output big or little endian (default input endian)
-f	fast (linear interpolation resample instead of 3D cubic spline)
-e	echo mat file to stdout frame by frame (verbose mode)
-O111	output in 111 space
-O222	output in 222 space
-O333.n	output in 333.n space (y shifted up by n pixels)
-O<str>	output image dimensions according to <str>.4dfp.ifh

N.B.: default output format = 333.0

1.17.8 mat2dat

convert cross_realign3d_4dfp mat files to spread sheet format Usage: mat2dat <mat_file>

Examples:

```
mat2dat atten5_b1_rms4_dbnd_xr3d[.mat]
```

Options

-I	save trajectory as 4dfp
-R	save trajectory relative to run mean (remove accumulated movememt)
-D	save differentiated trajectory
-L	write local (in \$cwd) (default write parallel to <mat_file>)
-n<int>	specify number of pre steady state frames (default=0)
-l<int>	lowpass filter (< 0.1 Hz) specified motion parameter (counting from 1)
TR_vol=<flt>	specify TR_vol in sec (required only with option -l)
-r<flt>	specify head radius in mm for total motion computation (default=50mm)
-f<str>	specify frames to count format, e.g., "4x120+4x76"

N.B.: -f option overrides -n

1.17.9 frame_align_4dfp

correct asynchronous slice acquisition

Usage: frame_align_4dfp <(4dfp) input> <frames_to_skip> [options]

Examples:

```
frame_align_4dfp bold_run.4dfp.img 4 -TR_vol 2.5 -TR_slc .136 -d 1
frame_align_4dfp bold_run.4dfp.img 4 -TR_vol 2.5 -TR_slc .136 -seqstr 1,8,5,2,9,6,3,
↪10,7,4
```

Options

-N	enable interleaved order 2,4,6,...,1,3,5,... for even total slice counts
-S	specify sequential slice acquisition (default interleaved)
-d <0 1>	specify slice acquisition direction (0:Inf->Sup; 1:Sup->Inf) (default=0)
-m <int>	specify multi-band factor) (default=1)
-seqstr <str>	specify [MB] slice sequence (counting from 1) as a comma-separated (no spaces) integer string
-TR_vol <flt>	specify frame TR in sec (default=2.36)
-TR_slc <flt>	specify slice TR in sec (default=TR_vol/nslice)

N.B.: space between option and value

1.17.10 interp_4dfp

correct asynchronous slice acquisition and resample in time

Usage: `interp_4dfp <(4dfp) image> <TR_vol_in> <TR_slice_in> <TR_vol_out>`

Examples:

```
interp_4dfp bold_run[.4dfp[.img]] 2.25 .136 2.5
```

Options

-d<0 1>	specify slice acquisition direction (0:Inf->Sup; 1:Sup->Inf) (default=1)
-@<bl>	output big or little endian (default input endian)

N.B.: if <TR_slice_in> is input as 0 slices are spaced evenly on TR_vol

1.17.11 jitter

optimally distribute n events on m frames

Usage: `jitter <(int) nevent> <(int) nframe> <(flt) tr_vol>`

Examples:

```
jitter 20 100 2.0 -s4
```

Options

-r<int>	specify randomization seed (default=0)
-s<int>	add specified number of skip frames to output event series (default=0)
-g<flt>	specify max interval in sec (t_max; default=30.00) (ignored when -F specified)
-m<flt>	specify min interval in sec (t_min; default=tr_vol)
-o<str>	output named fdl-type event file
-v	verbose mode
-F	use flat distribution of delay intervals (default Poisson process)

N.B.: nevent must be at least 3 N.B: first event is ALWAYS on frame skip; last event is ALWAYS on frame skip + nframe, duration = Inf; fMRI run should include additional frames at end

1.18 “Format” string manipulation

1.18.1 condense

generate maximally compact format string

Usage: condense <format_str>

Examples:

```
condense "4x86+4x86+4x86+4x86+4x86+4x86+4x86+4x86+4x86+"
# output: 9(4x86+)
```

Options

-v	verbose mode
-f<str>	read input format string from specified file (default command line)

1.18.2 format2lst

expand format string

Usage: format2lst <formatfmtfile>

Examples:

```
format2lst "2x3-2+1-2+2-2+1-1+2-1+1-1+1-1+2-1+1-1+1-2+2-1+1-1+2-2+1-2+2-" -e
# output: xx---+--+---+--+---+--+---+--+---+--+---+--+---+--+---
```

Options

-w	convert {'x' '+' '-'} to {0.0 1.0 -1.0}
-e	expand on single line

1.19 GLM and related operations

1.19.1 glm_4dfp

multivariate voxelwise regression/correlation

Usage: glm_4dfp <formatfmtfile> <profile> <4dfplconc input>

Examples:

```
glm_4dfp "4x124+" doubletask.txt b1_rmisp_dbnd_xr3d_norm
```

Options

-Z	supress automatic removal of mean from input regressors
-C<str>	read partial beta coefficients from specified 4dfp image (default compute)
-o[<i>str</i>]	save partial beta images with specified trailer (default = "coeff")
-R	compute partial beta images as percent modulation
-b[<i>str</i>]	save total beta images with specified trailer (default = "tbeta")
-p[<i>str</i>]	save partial corr images with specified trailer (default = "pcorr")
-t[<i>str</i>]	save total corr images with specified trailer (default = "tcorr")
-r[<i>str</i>]	save residual timeseries with specified trailer (default = "resid")
-@<bl>	output big or little endian (default input endian)

N.B.: conc files must have extension "conc" N.B.: <profile> lists temporal profiles (ASCII npts x ncol; '#' introduces comments) N.B.: <profile> line limits are 81920 chars and 8192 fields N.B.: absent -C, options -o and -r require design matrix inversion; dimension limit 256

1.19.2 actmapf_4dfp

voxelwise evaluate timeseries inner product against reference waveform

Usage: actmapf_4dfp <formatlfmtfile> <4dfplconc input>

Examples:

```
actmapf_4dfp -zu "3x3(11+4x15-)" b1_rmosp_dbnd_xr3d_norm
actmapf_4dfp -aanatomy -c10 -u "+" ball_dbnd_xr3d.conc
actmapf_4dfp -zu "4x124+" b1_rmosp_dbnd_xr3d -weights.txt
```

Options

-a<str>	specify 4dfp output root trailer (default = "actmap")
-c<flt>	scale output by specified factor
-u	scale weights to unit variance
-z	adjust weights to zero sum
-R	compute relative modulation (default absolute)
-w<weight file>	read (text) weights from specified filename
-@<bl>	output big or little endian (default input endian)

N.B.: conc files must have extension "conc" N.B.: when using weight files 'x' frames in format are not counted N.B.: relative modulation images are zeroed where mean intensity < 0.5*whole_image_mode

1.19.3 GC_4dfp

Granger causality mapping Usage: GC_4dfp <format> <4dfplconc input> <order>

Examples:

```
GC_4dfp "4(4x190+)" VB20579_rmosp_faln_dbnd_xr3d_at1.conc 2
```

Options

-w<str>	specify timecourse profile file (one or more columns)
-i<int>	use only specified column (counting from 1) of timecourse profile
-a<str>	append specified string to map output
-g	write lagged covariance (gamma) 4dfp stack
-D	write difference of directed influences ($F_{x \rightarrow y} - F_{y \rightarrow x}$) map
-Z	write Geweke “N(0,2)” measure (difference of square roots) map
-F	write $F_{x,y}$, $F_{x \rightarrow y}$, $F_{y \rightarrow x}$, $F_{x,y}$ map stack
-@<bl>	output big or little endian (default input endian)

N.B.: conc files must have extension “conc” N.B.: effective frame count is determined by <format> N.B.: ‘x’ frames in format are not counted

1.19.4 GC_dat

Granger causality on ASCII column data

Usage: GC_dat <format> <input_datafile> <order>

Examples:

```
GC_dat "4x106+" ROI_timeseries.dat 2
```

Options

-d	debug mode
-v	verbose mode
-u	normalize all input timeseries to unit variance
-x<int>	specify dimensionality of x process (default = 1)
-m	create text listing of AR model
-w	write residual after full AR modeling
-P	format residual output suitable for plotting (xxy)

1.19.5 covariance

covariance, correlation, coherence, etc. on ASCII column data

Usage: covariance <formatlfmtfile> <profile>

Examples:

```
covariance "4x124+" doubletask.txt
```

Options

-q	quiet mode
-t	optionally remove trend (ramp) from input timeseries
-u	optionally normalize all input timeseries to unit variance
-o	output lagged CCV dat files (CCR with -u)
-a	output lagged ACV dat file (ACR with -u)
-r	output Bartlett smoothed cross spectra (spectral density with -u)
-p	output Bartlett smoothed auto spectra (spectral density with -u)
-e	compute eigenvectors of lag 0 CCV
-L	read ROI labels from <profile> (default ignore '#' commented lines)
-T<int>	additionally smooth spectra with Tukey window of specified width (in frames)
-d<flt>	specify frame TR in sec for Fourier analysis (default = 1.0000)
-m<int>	specify CCV function maximum lag in frames (default = 32)
-D<flt>	SVD lag 0 CCV and output new profile with cndnum < specified value (implies -e)
-g<str>	regress timeseries in named file out of <profile>

N.B.: all input timeseries are made zero mean as a first step N.B.: region names can be specified on the first line of <profile> with '#' in the first column

1.20 Image algebra

1.20.1 sqrt_4dfp

\sqrt{A}

Examples:

```
sqrt_4dfp vce20_mpr
```

Options

-@<bll>	output big or little endian (default input endian)
-E	output undefined voxels as 1.0e-37 (default 0.0)

N.B.: default output filename = <image>_sqrt

1.20.2 scale_4dfp

$m*A + b$

Usage: scale_4dfp <image_4dfp> <scale_factor> [options]

Examples:

```
scale_4dfp b2_xfrm_avg 12  
scale_4dfp b2_xfrm_avg 12 -b5 -ax12+5
```

Options

-E	preserve 1.0e-37 values (fidl NaN convention)
-a<str>	append trailer to output file name
-b<flt>	add specified constant to each voxel
-@<bll>	output big or little endian (default input endian)

N.B.: <image_4dfp> is overwritten unless the trailer option is used N.B.: <scale_factor> must be specified for proper operation

1.20.3 imgopr_4dfp

A+B, A-B, A*B, A/B, various special operations

Usage: imgopr_4dfp -<operation><(4dfp) outroot> <(4dfp) image1> <(4dfp) image2> ...

Operations

a	add
s	subtract (image1 - image2)
p	product
r	ratio (image1 / image2)
e	mean (expectation)
v	variance
g	geometric mean
n	count defined (see -u option) voxels
x	voxelwise maximum
y	voxelwise minimum
G	report serial number (counting from 1) of image with greatest value
P	unsplit multiple ROIs into fidl compatible ROI file

Options

-u	count only defined (not NaN or 1.e-37 or 0.0) voxels
-R	suppress creation of rec file
-N	output undefined voxels as NaN
-Z	output undefined voxels as 0
-E	output undefined voxels as 1.E-37 (default)
-c<flt>	multiply output by specified scaling factor
-l<lst>	read input file names from specified list file
-@<bll>	output big or little endian (default first input endian)

N.B.: image dimensions must match except for binary operations {aspr} in which a 1 volume second image may be paired with a multi-volume first image

1.21 Image segmentation and gain field correction

1.21.1 partitiond_gfc_4dfp

intensity inhomogeneity correction assuming 3D parabolic gain field

Usage: partitiond_gfc_4dfp <imgroot>

Examples:

```
partitiond_gfc_4dfp vc1440_mpr_n4_111_t88.4dfp
```

Options

-g	freeze initial gain field
-n	force negative definite quadratic gain field
-v	verbose mode
-p<flt>	pre-blur by specified FWHM in mm
-b<flt>	specify bandwidth in intensity units (default=200.0)
-e<flt>	specify drms convergence criterion (default=0.000200)
-i<flt>	specify sigma (default=1.000000)
-l<int>	specify iteration limit (default=8)
-m<flt>	specify gfc computation region count (default=24)
-s<flt>	specify space constant in mm (default=4.000000)
-z<flt>	specify background threshold (default=180.0)
-M<flt>	specify maximum correction factor
-r<flt>[to<flt>]	specify gfc range (default=0.0to10000.0)

1.22 Interconvert image formats

1.22.1 dcm_to_4dfp

DICOM → 4dfp

Usage: dcm_to_4dfp [-b base] [-d gggg eeee] [-f] [-g] [-u] file(s)

Slice Spacing Options: [-c] [-t <flt> or S or T]

Slice Position Options: [-X] [-Y] [-Z]

Examples:

```
dcm_to_4dfp *
dcm_to_4dfp -b ID101 -f -g -u *IMA
dcm_to_4dfp -d 0008 0030 -t 4.98 -g *.dcm
dcm_to_4dfp -b P0089 -t T -g mydir/*
```

Options

[-b base]	Output base filename follows the -b
[-c]	Slice Spacing: By Image Position (0020 0032)
[-d gggg eeee]	Divide series by group and element number (Default: ID series time (0008 0031))
[-f]	Directories will be created, and dicom files will be moved
[-g]	Add image name, XYZ relative position, and number to rec file
[-q]	Slice Spacing: Do not compute by Image Position
[-r]	Rescale: Use the rescale slope and intercept fields
[-t <flt>]	Slice Spacing: Use input value.[-t <flt>]
[-t T]	Slice Spacing: Use Slice Thickness 0018 0050.[-t T]
[-t S]	Slice Spacing: Use Slice Spacing 0018 0088 [-t S]
[-u]	Output files named using sequence tag 0018 0024 plus number

4dfp Coordinant System is determined by Image Position (0020 0032). Multivolume and BOLD images are ordered by REL Image Number (0020 0013). [-X] Sagittal: image positions will be ordered low to high [-Y] Coronal: image positions will be high to low [-Z] Transverse: image positions will be high to low [-@ <bl>] output big or little endian (default CPU endian)

N.B.: -t S is the default slice spacing N.B.: Default slice position is transverse ordered by REL Image Number (0020 0013)

1.22.2 endian_4dfp

report status and interconvert big ↔ little endian

Usage: endian_4dfp <(4dfp) image>

Options

-@<blc>	make <(4dfp) image> big, little or CPU endian
-t	perform var(log(fabs(.))) test

N.B.: <(4dfp) image> may be overwritten N.B.: absent option -@ endian_4dfp only reports state of <(4dfp) image>

1.22.3 4dfptoanalyze

4dfp → analyze 7.5

Usage: 4dfptoanalyze <(4dfp) filename>

Options

-c<flt>	scale output values by specified factor
-8	output 8 bit unsigned char
-SPM99	include origin and scale in hdr (http://wideman-one.com/gw/brain/analyze/format.doc)
-@<bl>	output big or little endian (default CPU endian)

1.22.4 analyzeto4dfp

analyze 7.5 (int or char) → 4dfp

Usage: analyzeto4dfp <analyze_image>

Options

-s	apply SPM2 ROIScaleFactor
-x	flip first axis
-y	flip second axis
-z	flip third axis
-@<bl>	toutput big or little endian (default CPU endian)
-O<int>	supply orientation code (in range [0-5])

N.B.: to convert SPM2 use options -x and -s

1.22.5 ifh2hdr

create analyze 7.5 header

Usage: ifh2hdr <(4dfp) file>

examples:

```
ifh2hdr vc654_mpr_atl -r-500to1500
```

Options

-r<flt>[to<flt>]	set range
------------------	-----------

1.22.6 hdr2txt

dump analyze 7.5 header info

Usage: hdr2txt <analyze_image>

Examples:

```
hdr2txt brain_asig[.hdr]
```

1.22.7 index2atl

convert atlas indices (ASCII text) to mm (e.g. atlas coordinates)

Usage: index2atl <(4dfp) ifhroot> <index_list_file>

Examples:

```
index2atl -af time_BOXzstat_333_t88.4dfp.ifh time_BOXzstat_333_t88_index.lst
```

Options

-f	input indices use FORTRAN convention (first index=1) (default first index=0)
-a	indices were read under orientation-specific 4dfp<->analyze flips
-o<str>	output coordinates to specified file

N.B.: <(4dfp) ifhroot> corresponds to the 4dfp image from which the indices were read

1.22.8 asciito4dfp

convert text columns to 4dfp format timeseries

Usage: asciito4dfp <text file> <(4dfp) out>

Options

-@<bl>	output big or little endian (default CPU endian)
--------	--

N.B.: columns in <text file> map to voxels in <(4dfp) out> N.B.: '#' in <text file> introduce comments N.B.: <text file> lines beginning with '#' are included in <(4dfp) out>.img.rec

1.22.9 mpetto4dfp

convert microPET images 4dfp

Usage: mpetto4dfp <microPET_data>

Examples:

```
mpetto4dfp m1042-cft1_v1
```

Options

-x	flip x
-y	flip y
-z	flip z
-w	create frame duration listing for use with actmapf_4dfp -w
-c<flt>	scale all voxel values by specified factor
-o<str>	name 4dfp output using specified string (default same as input)
-@<bll>	output big or little endian (default input endian)

1.22.10 vto4dfp

Varian fid/proctpar → 4dfp

Usage: vto4dfp <varian file path>

Examples:

```
vto4dfp /home/usr/shimonyj/vto4dfp/hard_010703 -odwi_010703
```

Options

-v	verbose mode
-D	suppress subtraction of k-space DC offset
-I	perform Fourier interpolation; output voxel count will be quadrupled
-F	phase reverse odd echos in multi-echo data
-o<str>	specify 4dfp outroot (default="fid")
-c<flt>	intensity scale output (mag) image by specified constant
-m<flt>	scale voxel dimensions by specified constant
-@<bll>	output big or little endian (default CPU endian)

N.B.: vto4dfp expects <varian file path> to contain files “fid” and “proctpar”

1.22.11 nifti_4dfp

interconvert nii ↔ 4dfp

Usage: nifti_4dfp -<4ln> <infile> <outfile> [options]

Examples:

```
nifti_4dfp -n time_BOXzstat_333_t88.4dfp.ifh time_BOXzstat_333_t88.nii
```

Options

-T <t4 file>	specify a t4 file to use converting TO NIfTI from 4dfp
-n	convert TO NIfTI from 4dfp
-4	convert TO 4dfp from NIfTI
-N	suppress saving of mppix and center fields in output ifh
-@<val>	specify endianness for output, b or B for big, l or L for little

N.B.: exactly one of -4 or -n must be specified N.B.: “.4dfp.ifh” or “.nii” are appended to filenames specified without extension N.B.: option -N has effect only on converting nii->4dfp N.B.: option -T has effect only on converting 4dfp->nii

1.22.12 niftigz_4dfp

interconvert nii.gz ↔ 4dfp (*nifti_4dfp* wrapper)

Usage: niftigz_4dfp -<ln> <infile> <outfile> [options]

Examples:

```
niftigz_4dfp -4 VB18896_mpr_n1_333_t88.nii.gz VB18896_mpr_n1_333_t88
```

Options (for more options, see *nifti_4dfp*)

-v	verbose mode
-s<int>	skip specified number of frames at run start on 4dfp->NIfTI conversion

N.B.: niftigz_4dfp always gzips NIfTI output but unzipped NIfTI input is left unchanged

1.23 Rearrange voxels in space or time

1.23.1 collate_slice_4dfp

collate interleaved datasets

Usage: collate_slice_4dfp <4dfp img1> <4dfp img2> ... <4dfp imgn> <4dfp imgout>

Options

-v	verbose mode
-@<bl>	output big or little endian (default CPU endian)

1.23.2 paste_4dfp

append or average selected frames from multiple files (list directed)

Usage: paste_4dfp <inlist> <outfile>

inlist is a file containing rows of 1 to 3 columns: filename, starting frame (counting from 1), number of frames.

Note:

- If a starting frame (column 2) is not specified, the first frame will be used.
- Column 3 is only applied during append mode (-a). The period value (-p) will be used otherwise.
- In append mode, column 3 has priority over the -p flag. The period value (-p) will only be used for rows that do not specify column 3.

Options

-a	append successive epochs (default average)
-p<int>	specify period in frames (default=1)
-@<bl>	output big or little endian (default initial input endian)

1.23.3 extract_frame_4dfp

extract single frame from stack (*paste_4dfp* wrapper)

Usage: extract_frame_4dfp <(4dfp) stack> <(int) frame>

Examples:

```
extract_frame_4dfp CDR.5to1+ 3
```

Options

-o<str>	specify output 4dfp fileroot (default = <stack>_frame<frame>)
---------	---

1.23.4 chop_4dfp

extract contiguous frames from stack (*paste_4dfp* wrapper)

usage: chop_4dfp <(4dfp) stack> <(int) frame0> <(int) frame1>

Examples:

```
chop_4dfp vb12345_b5_dbnd_xr3d[.4dfp[.img]] 4 68
```

Options

-o<str>	specify output 4dfp fileroot (default = <stack>_frames<frame0>to<frame1>)
---------	---

1.23.5 crop_4dfp

crop or roll (correct image wrap)

Usage: crop_4dfp <(4dfp) inroot> [(4dfp) outroot]

Options

-<xlyz><int>[to<int>]	specify x y z crop/expand limits (1-indexed)
-s<xlyz><int>	scroll specified axis by specified number of pixels (after cropping/expanding)
-f	interpret specifications under 4dfp<->analyze flips
-Z	zero voxels instead of physically cropping
-@<bl>	output big or little endian (default input endian)

N.B.: if upper crop limit exceeds input dimension undefined voxels will be set to 1.e-37 N.B.: default (4dfp) output root is <(4dfp) inroot>”_crop”

1.23.6 reindex_4dfp

xy, slicevolume

Usage: reindex_4dfp <(4dfp) input> <index1> <index2> [options]

Examples:

```
reindex_4dfp my4Dstack 3 4
```

Options

-v	verbose mode
-o<str>	specify 4dfp output root (default = <input>_r<index1><index2>)
-@<bl>	output big or little endian (default input endian)

N.B.: reindex_4dfp swaps specified indices N.B.: <index1> and <index2> must be unequal integers in the range 1-4 except as follows: <index1> == 4 and <index2> == 0: right rotate indices (first index <- last index) <index1> == 0 and <index2> == 4: left rotate indices (last index <- first index)

1.23.7 unpack_4dfp

mosaic → volume

Usage: unpack_4dfp <(4dfp) input> <(4dfp) output>

Examples:

```
unpack_4dfp 030211_EL_b_1 030211_EL_b1
```

Options

-V	read frame count from input ifh slice count
-R	multiply output x and y voxsiz by pack factor
-z	flipz (unpack slices in reverse order)
-y	flipy
-nx<int>	specify unpacked nx (default=64)
-ny<int>	specify unpacked ny (default=64)
-sx<int>	squeeze unpacked x dimension by specified factor
-sy<int>	squeeze unpacked y dimension by specified factor
-@<bl>	output big or little endian (default input endian)

1.23.8 multipack_4dfp

volume → mosaic

1.23.9 flip_4dfp

flip x, y, z

Usage: flip_4dfp <(4dfp) image> [(4dfp) output]

Examples:

```
flip_4dfp -yz vc345 vc345_flipyz
```

Options

-x	flip x
-y	flip y
-z	flip z
-@<bl>	output big or little endian (default input endian)

N.B.: default output fileroot = <image>_flip[xyz]

1.23.10 split_4dfp

split assembled volumes

1.23.11 T2S_4dfp

transverse → sagittal

Usage: T2S_4dfp <(4dfp) imgroot> [(4dfp) outroot]

Examples:: T2S_4dfp vm6c_mpr T2S_4dfp vm6c_mpr vm6c_mprS

Options

-@<bl>	output big or little endian (default input endian)
--------	--

N.B.: default output root = <imgroot>”S”

1.23.12 S2T_4dfp

sagittal → transverse

Usage: S2T_4dfp <(4dfp) imgroot> [(4dfp) outroot]

Examples:

```
S2T_4dfp vm6c_mpr
S2T_4dfp vm6c_mpr vm6c_mprT
```

Options

-@<bl>	output big or little endian (default input endian)
--------	--

N.B.: default output root = <imgroot>”T”

1.23.13 C2T_4dfp

coronal → transverse

Usage: C2T_4dfp <(4dfp) image> [(4dfp) outroot]

Examples:

```
C2T_4dfp vm6c_b1
C2T_4dfp vm6c_b1 vm6c_b1T
```

Options

-@<bl>	output big or little endian (default input endian)
--------	--

N.B.: default output root = <imgroot>”T”

1.23.14 T2C_4dfp

transverse → coronal

Usage: T2C_4dfp <(4dfp) imgroot> [(4dfp) outroot]

Examples:

```
T2C_4dfp vc12345_b1
T2C_4dfp vc12345_b1 vc12345_b1C
```

Options

-@<bl>	output big or little endian (default input endian)
--------	--

N.B.: default output root = <imgroot>”C”

1.24 Register in space (and other t4 oriented programs)

1.24.1 imgreg_4dfp

compute transform (various modes)

Usage:

```
imgreg_4dfp target_imag target_mask source_imag source_mask t4file mode
imgreg_4dfp target_imag          none source_imag source_mask t4file mode
imgreg_4dfp target_imag          none source_imag          none t4file mode
```

Mode options

1	enable coordinate transform
2	enable 3D alignment
4	enable affine warp (12 parameters in 3D 6 parameters in 2D)
8	enable voxel size adjust
16	disable x voxel size adjust
32	disable y voxel size adjust
64	disable z voxel size adjust
128	unassigned
256	when set use difference image minimization (for similar contrast mechanisms)
512	superfine mode (2 mm cubic grid metric sampling)
1024	fast mode (12 mm cubic grid metric sampling)
2048	fine mode (5 mm cubic grid metric sampling)
4096	[T] restricted to translation explored at 7.5 mm intervals
8192	enable parameter optimization by computation of the metric gradient in parameter space and inversion of the Hessian

1.24.2 t4imgs_4dfp

apply transforms, resample and average (list directed)

Usage: t4imgs_4dfp [options] <inlist> <outfile>

Example inlist:

```
<imgfile1>      t4=<t4_file1>
<imgfile2>      t4=<t4_file2>
<imgfile3>      t4=<t4_file3>
```

Options

-z	normalize by sqrt(n) rather than n (for z images)
-s	interpolate by 3D cubic spline (default is 3D linear)
-N	output NaN (default 0.0) for undefined values
-B	internally convert to_711-2A_t4->to_711-2B_t4
-n	use nearest neighbor interpolation
-R	suppress creation of rec file
-O111	output in 111 space instead of default 333.0 space
-O222	output in 222 space instead of default 333.0 space
-O333.n	output in 333.n space (y shifted up by n pixels)
-Omy_image	duplicate dimensions of my_image.4dfp.ifh
-@<bl>	output big or little endian (default CPU endian)

N.B.: t4file intensity scale ignored with option -n

1.24.3 t4img_4dfp

single image wrapper for *t4imgs_4dfp*

Usage: t4img_4dfp <t4file> <imgfile> [outfile]

Examples:

```
t4img_4dfp vcel_mprS_to_711-2B_t4 vcel_mprS.4dfp.img -O222
t4img_4dfp vcel_mprS_to_711-2B_t4 vcel_mprS vce_mprS_711-2B -O222
t4img_4dfp none vcel_mprS vcel_mprS_222 -O222
```

Options

-z	normalize by sqrt(n) rather than n (for z images)
-s	interpolate by 3D cubic spline (default is 3D linear)
-N	output NaN (default 0.0) for undefined values
-B	internally convert to_711-2A_t4->to_711-2B_t4
-n	use nearest neighbor interpolation
-R	suppress creation of rec file
-O111	output in 111 space instead of default 333.0 space
-O222	output in 222 space instead of default 333.0 space
-O333.n	output in 333.n space (y shifted up by n pixels)
-Omy_image	duplicate dimensions of my_image.4dfp.ifh
-@<bl>	output big or little endian (default CPU endian)
outfile	specify name for output file (default is <imgfile>t)

N.B.: 4dfp filename extensions are optional N.B.: option -n causes fidl ROI names to be copied to the output ifh

1.24.4 wrpsmg_4dfp

apply transforms, resample and average difference images (list directed)

Usage: wrpsmg_4dfp [options] <inlist> <outfile>

Options

-N	output NaN (default 0.0) for undefined values
-w	create sum of weights image
-s	create square root variance (sd) image
-O111	output in 111 space
-O222	output in 222 space (default)
-O333.n	output in 333.n space (y shifted up by n pixels)
-Omy_image	duplicate dimensions of my_image.4dfp.ifh
-@<bl>	output big or little endian (default CPU endian)

1.24.5 stretch_out

remove transform stretch

Usage: stretch_out <t4file> [t4file_new]

N.B.: default output filename is <t4file>”r”

1.24.6 t4_mul

compose transforms

Usage: t4_mul <left_t4file> <right_t4file> [product_t4file]

Examples:

```
t4_mul vm11b_anat_ave_to_vm11b_234-3_t4 vm11b_234-3_to_711-2B_t4 [vm11b_anat_ave_to_
↪711-2B_t4]
```

1.24.7 t4_inv

invert transform

Usage: t4_inv <t4file> [inv_t4file]

Examples:

```
t4_inv vm11b_anat_ave_to_vm11b_234-3_t4 [vm11b_234-3_to_vm11b_anat_ave_t4]
```

Options

-u	suppress (intensity) scale field in output
----	--

1.24.8 t4_factor

decompose affine transform into components (translation, rotation, stretch)

Usage: t4_factor <t4file>

Examples:

```
t4_factor vm11b_anat_ave_to_vm11b_234-3_t4
```

1.24.9 t4_null

create an identity transform t4 file

Usage: t4_null <t4file>

Examples:

```
t4_null vm11b_mpr1_to_711-2B_t4
```

1.24.10 t4_resolve

compute optimal rigid body transforms connecting a set of images

Usage: t4_resolve <image1> <image2> ...

Options

-v	verbose mode
-m	generate mat file output
-s	include intensity scale factor in t4 file output
-w	weight inversely in proportion to scale in sub file output (sum counts mode)
-o<str>	write resolved output with specified fileroot
-r<flt>	set VOI rms radius in mm (default=50)

N.B.: t4_resolve looks for t4 files <image1>_to_<image2>_t4, <image1>_to_<image3>_t4, ... N.B.: t4_resolve automatically strips filename extensions when constructing t4 filenames

1.24.11 t4_pts

inter-convert coordinates, e.g., 711-2B ↔ MNI152

Usage: t4_pts <t4file> <pts.lst> [new pts.lst]

Examples:

```
t4_pts 711-2B_to_MNI152lin_T1_t4 711-2B_coords MNI152_coords
```

1.25 SPM-like voxelwise statistical operations

1.25.1 t2z_4dfp

t-map → Z-map

Usage: t2z_4dfp <(4dfp) t-image>

Examples:

```
t2z_4dfp NP705_cond1_zfrm_RFX -nNP705_cond1_N
```

Options

-l	output -log ₁₀ (prob(t))
-N<flt>	specify global n
-n<str>	specify 4dfp n-image (up to two allowed)
-@<bll>	output big or little endian (default input endian)

N.B.: undefined (1.e-37, NaN) voxels in input are output as 1.e-37 N.B.: output values are assigned the same sign as the input t value N.B.: the same n values apply to all volumes the input <t-image>

1.25.2 z2logp_4dfp

Z-map → log₁₀p-map

Usage: z2logp_4dfp <(4dfp) Z-image>

Examples:

```
z2logp_4dfp vce20_z[.4dfp[.img]]
```

Options

-2	two sided test (default one sided)
-p	output p-values (default output -log ₁₀ (p))
-@<bll>	output big or little endian (default input endian)

N.B.: probability computed on assumption that voxel values are N(0,1) N.B.: undefined (1.e-37, NaN, Inf) voxels in input are output as 1.e-37

1.25.3 rho2z_4dfp

r-map ↔ Fisher z-map

Usage: rho2z_4dfp <(4dfp) image> [outroot]

Examples:

```
rho2z_4dfp vce20_rho[.4dfp[.img]]
```

Options

-r	reverse (convert z to r) (output trailer = _corr)
-E	output undefined voxels as 1.0e-37 (default 0.0)
-@<bll>	output big or little endian (default input-endian)

N.B.: default r to z output filename = <image>_zfrm

1.26 Threshold and mask

1.26.1 zero_slice_4dfp

zero specified range of slices in selected direction

Usage: zero_slice_4dfp <4dfp image>

Examples:

```
zero_slice_4dfp vce20_mpr -z1to3
zero_slice_4dfp vce20_mpr <x|y|z> istart iend [outroot]
```

Options

-<x y z><int>to<int>	specify x y z limits (single required argument mode)
-f	interpret slice numbers using 4dfp<->analyze flips
-o	specify output fileroot (default = <image>z)
-@<bll>	output big or little endian (default input endian)

N.B.: slices count from 1 N.B.: two usages are supported: 1 or 4 required arguments

1.26.2 zero_lt_4dfp

threshold by voxel value

Usage: zero_lt_4dfp <flt> <file_4dfp> [outroot]

Examples:

```
zero_lt_4dfp 90 pt349_study9to9
zero_lt_4dfp 90 pt349_study9to9 pt349_study9to9z
```

Options

-@<bll>	output big or little endian (default input endian)
---------	--

N.B.: default output 4dfp root is <file_4dfp>”z”

1.26.3 zero_gt_4dfp

threshold by voxel value

Usage: zero_gt_4dfp <flt> <(4dfp) image> [outroot] [options]

Examples:

```
zero_gt_4dfp 90 pt349_study9to9
zero_gt_4dfp 90 pt349_study9to9 pt349_study9to9z
```

Options

-@<bll>	output big or little endian (default input endian)
---------	--

N.B.: default output 4dfp root is <(4dfp) image>”z” N.B.: first field can’t be used for options because threshold might be negative

1.26.4 zero_ltgt_4dfp

zero voxels **outside** specified range

Usage: zero_ltgt_4dfp <flt[to<flt>]> <(4dfp) image> [outroot] [options]

Examples:

```
zero_ltgt_4dfp -30to90 pt349_study9to9
```

Options

-@<bll>	output big or little endian (default input endian)
---------	--

N.B.: default output 4dfp root is <(4dfp) image>”z” N.B.: first field can’t be used for options because lower range might be negative

1.26.5 zero_gtlgt_4dfp

zero voxels **within** specified range

Usage: zero_gtlgt_4dfp <flt[to<flt>]> <(4dfp) image> [outroot] [options]

Examples:

```
zero_gtlgt_4dfp -30to90 pt349_study9to9
```

Options

-@<bll>	output big or little endian (default input endian)
---------	--

N.B.: default output 4dfp root is <(4dfp) image>”z” N.B.: first field can’t be used for options because lower range might be negative

1.26.6 maskimg_4dfp

apply 4dfp mask to 4dfp image

Usage: maskimg_4dfp <(4dfp) imgfile> <(4dfp) mskfile> <(4dfp) outfile>

Examples:

```
maskimg_4dfp -t23.2 va1234_mpr mask va1234_mpr_msk
```

Options

-N	replace NaN in <imgfile> with corresponding <mskfile> value
-e	report to stdout mean <imgfile> within-mask value
-l	apply first frame of <mskfile> to all frames of <imgfile>
-R	suppress creation of rec file
-v<flt>	specify <outfile> uniform within-mask value
-p<flt>	specify <mskfile> threshold as percent of <mskfile> max
-t<flt>	specify <mskfile> threshold directly (default = 0.0)
-A	threshold mask by absolute value of <mskfile>
-@<bl>	output big or little endian (default <imgfile> endian)

N.B.: <imgfile> and <mskfile> may be the same

1.26.7 cluster_4dfp

sort/count/zero (above threshold) contiguous voxels into clusters

Usage: cluster_4dfp <(4dfp) root>

Examples:

```
cluster_4dfp my_timage -At3.5
```

Options

-n<int>	zero out clusters with voxel count below specified criterion (output image trailer = 'clus')
-f<int>	address specified volume (counting from 1) of multi-volume stack (default is first volume)
-t<flt>	specify image value threshold (default = 0)
-a<str>	append specified string (preceded by "_") to all output filenames
-@<bl>	output big or little endian (default input endian)
-A	apply threshold test to image absolute value
-R	convert clusters to (fidl compliant) ROI image (output image trailer = 'ROI')
-l	create list file of region center of mass indices
-v	verbose mode

N.B.: -l center of mass indices can be converted to atlas coordinates using index2atl -af

1.27 Params/Instructions files

Many of the csh scripts take one or two text files as input (called "params" and "instructions"). The two files are for convenience. You can specify settings all in one file ("params"), or specify some in one ("params") and more in

another (“instructions”). A typical, convenient use case would be putting subject- or session-specific settings (e.g. patid, scan series numbers, etc.) in the “params” file and study-specific settings that don’t change across subjects (e.g. target atlas, BOLD scan TR, etc.) in the “instructions” file.

Because the instructions file is sourced after the params file, you can reference settings from the params file in the instructions file:

```
# in params
set patid = TM201

# in instructions
set inpath = /some/study/path/${patid}
```

1.27.1 Params file

```
# TM201.params

set patid = TM201
set mprs = ( 8 )
set t2ws = ( 10 )
set irun = ( thumb1 browthumb blink1 brow1 blink2 brow2 thumb2 thumbbrow )
set fstd = ( 14 16 18 20 22 24 26 28 )
set sefm = ( 11 12 )
```

1.27.2 Instructions file

```
# TM_instructions.txt

@ sorted = 1
@ economy = 5
@ go = 1
@ usescr = 0
set target = $REFDIR/TRIO_Y_NDC
@ nx = 72
@ ny = 72
set TR_vol = 0.6
set TE_vol = 33
set TR_slc = 0
set delta = ""
set ped = "y-"
set dwell = "0.59"
@ MBfac = 6
@ epidir = 0
@ skip = 5
@ epi2atl = 1
@ normode = 1
set tse = ( ${t2ws[1]} ) # for bold pp

set FDthresh = 0.2
@ min_frames = 120

set srcdir = $cwd

#####
```

(continues on next page)

(continued from previous page)

```
# FM unwarping
#####
set datain = /data/nil-bluearc/black/scripts/TicModel_fieldmap_topup_datain.txt
set FMmag = $srcdir/sefm/${patid}_sefm_mag.nii.gz
set FMphase = $srcdir/sefm/${patid}_sefm_fieldmap.nii.gz
set uwrp_cmnd = /data/gizmo/data1/NEWT_phantom/kqa778_20_vs_64/fmri_unwarp_se.csh
set rsam_cmnd = /data/nil-bluearc/benzinger2/Tyler/scripts/one_step_resample.tcsh
```