

Neuroimaging Analysis Kit – NIAK – user's guide



<http://www.nitrc.org/projects/niak/>

Release 0.7 – “ammo” – July 2013

What's NIAK ?

NIAK is ...

- an opensource (**MIT license**) library
- processing functional magnetic resonance images (**fMRI**)
- running inside **GNU Octave** or **Matlab(r)**
- running on Linux operating systems (**Ubuntu** recommended).
- supporting the **MINC** neuroimaging file format.

What's NIAK ?

NIAK features ...

- A pipeline for T_1 and fMRI **preprocessing**.
- A pipeline for **region growing** (full brain functional parcellation).
- A pipeline to generate **connectomes** and graph measures.

- 1 The fMRI preprocessing pipeline
- 2 The region growing pipeline
- 3 The connectome pipeline
- 4 Pipeline manager
- 5 Installation, contributions and further resources

How to start the fMRI preprocessing pipeline ?

Syntax

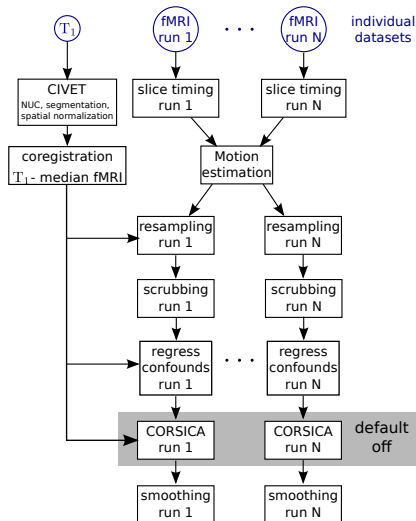
```
niak_pipeline_fmri_preprocess(files_in,opt)
```

- 1 **files_in** is a structure describing how the dataset is organized.
- 2 **opt** is a structure describing the options of the pipeline.

The list of the results and a detailed tutorial of the pipeline can be found at :

<http://www.nitrc.org/plugins/mwiki/index.php/niak:FmriPreprocessing>.

Flowchart of the fMRI preprocessing pipeline



Simplified flowchart of the fMRI preprocessing pipeline for one subject.

Describing the dataset

Input files

% Subject 1: structural scan

`files_in.subject1.anat` = `'/data/anat_subj1.mnc.gz'`;

% Subject 1: fMRI scans

`files_in.subject1.fmri.session1.motor` = `'/data/func_motor_subj1.mnc.gz'`;

`files_in.subject1.fmri.session1.rest` = `'/data/func_rest_subj1.mnc.gz'`;

% Subject 2: structural scan

`files_in.subject2.anat` = `'/data/anat_subj2.mnc.gz'`;

% Subject 2: fMRI scans

`files_in.subject2.fmri.session1.motor` = `'/data/func_motor_subj2.mnc.gz'`;

`files_in.subject2.fmri.session1.rest` = `'/data/func_rest_subj2.mnc.gz'`;

General options

Output directory

```
opt.folder_out = '/database/data_demo/fmri_preprocess/';
```

where to store the results of the pipeline. The pipeline manager will create but can also delete files and subfolders in that location.

General options

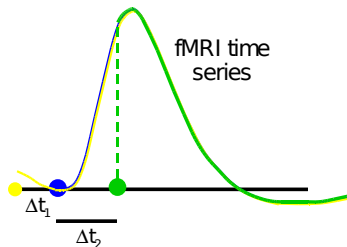
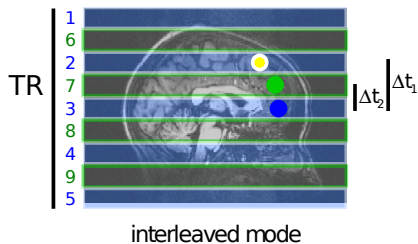
Size of outputs

```
opt.size_output = 'quality_control';
```

adjusts the quantity of intermediate results that are generated by the pipeline:

- 1 'quality_control' : Intermediate steps are deleted as soon as they are no longer necessary.
- 2 'all' : all possible outputs are generated at each stage of the pipeline.

Slice timing correction I



→ temporal interpolation to a single reference time for each volume (cubic spline interpolation)

Courtesy of Dr M. Pélégriani-Issac.

Slice timing correction II

Slice timing options – see `niak_brick_slice_timing`

% Number of dummy scans to suppress.

```
opt.slice_timing.suppress_vol = 0;
```

% Slice timing order. Available options :

% 'sequential ascending' , 'sequential descending',

% 'interleaved ascending' , 'interleaved descending'

```
opt.slice_timing.type_acquisition = 'interleaved ascending';
```

% Scanner manufacturer.

% Only the value 'Siemens' will actually have an impact

```
opt.slice_timing.type_scanner      = 'Bruker';
```

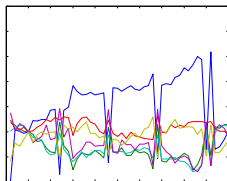
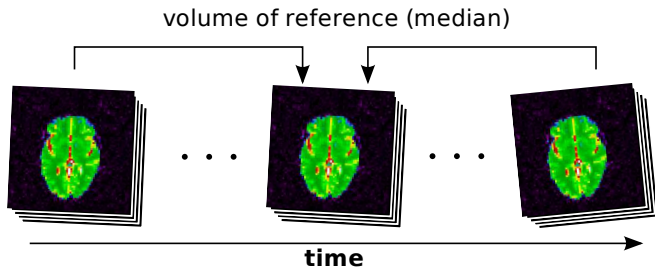
% The delay in TR ("blank" time between two volumes)

```
opt.slice_timing.delay_in_tr      = 0;
```

% Skip the slice timing (0: don't skip, 1 : skip)

```
opt.slice_timing.flag_skip        = 0;
```

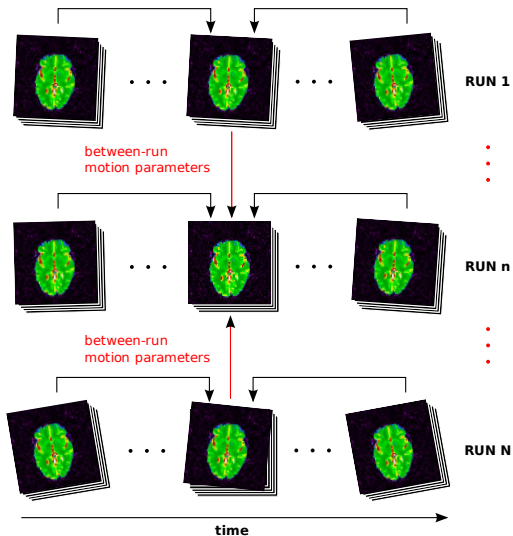
Motion estimation: within-run



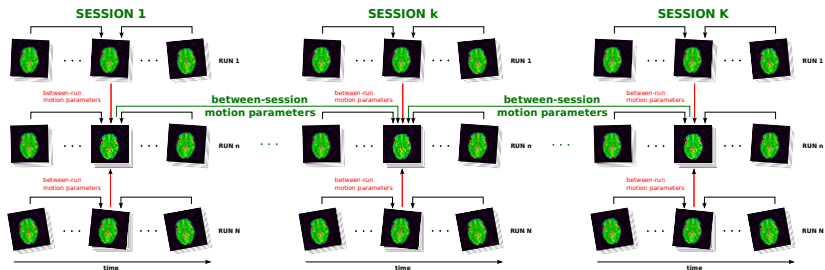
time

within-run motion parameters :
3 rotations
3 translations
for each volume

Motion estimation: between-run / within-session



Motion estimation: between sessions



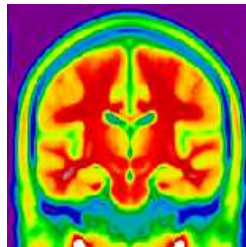
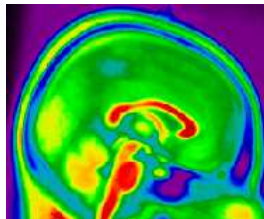
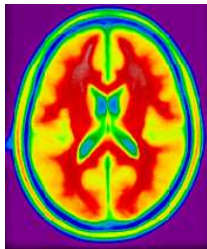
Estimation of between-run (between-session) rigid-body motion.

Motion estimation: parameters

Motion estimation – see `niak_pipeline_motion`

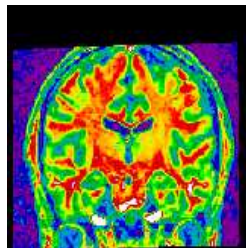
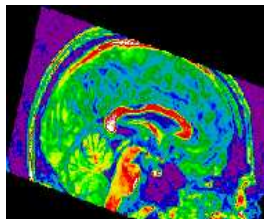
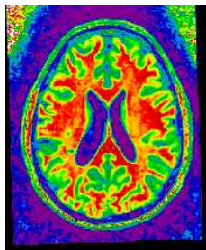
```
% The session that is used as a reference.  
% In general, use the session  
% including the acquisition of the T1 scan.  
opt.motion_correction.session_ref = 'session1';
```

T₁ processing: linear template



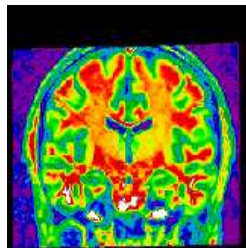
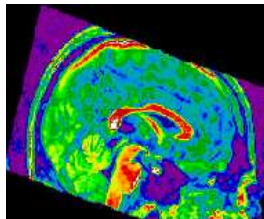
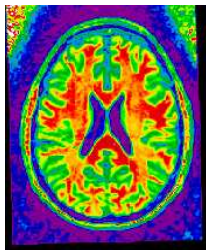
Linear ICBM template (average of 152 subjects)

T₁ processing: linear coregistration



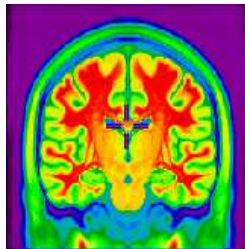
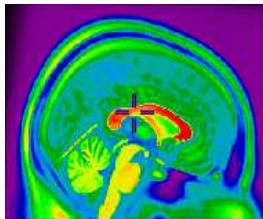
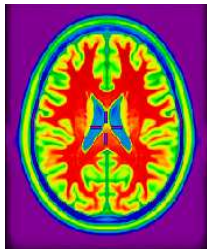
Individual structural scan (linear coregistration)

T₁ processing: non-linear coregistration



Individual structural scan (non-linear coregistration)

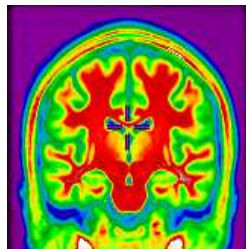
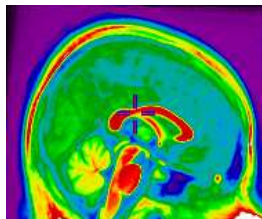
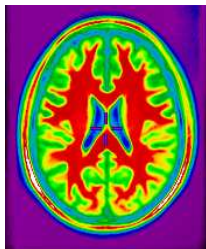
T₁ processing: nonlinear template



Symmetric non-linear ICBM template (average of 152 subjects)
release 2009a.

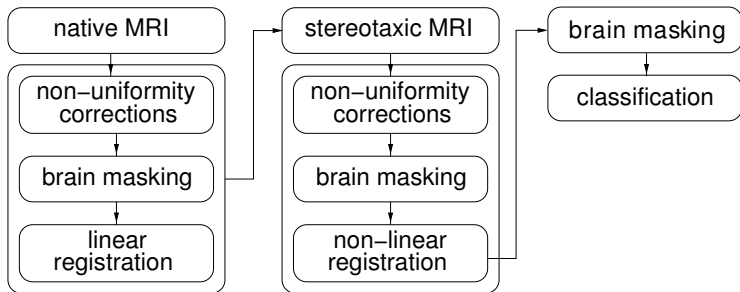
<http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152NLin2009>

T₁ processing: group average



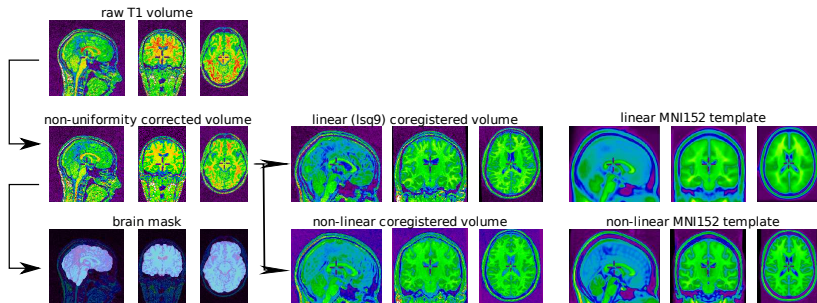
Average of 17 subjects (non-linear coregistration)

T₁ processing: Flowchart of the CIVET pipeline



Flowchart of the T1 preprocessing.

T_1 processing: main outputs



The main outputs of the T_1 processing pipeline.

T₁ processing: parameters

T₁ processing: see `niak_brick_t1_preprocess`

% Parameter for non-uniformity correction.

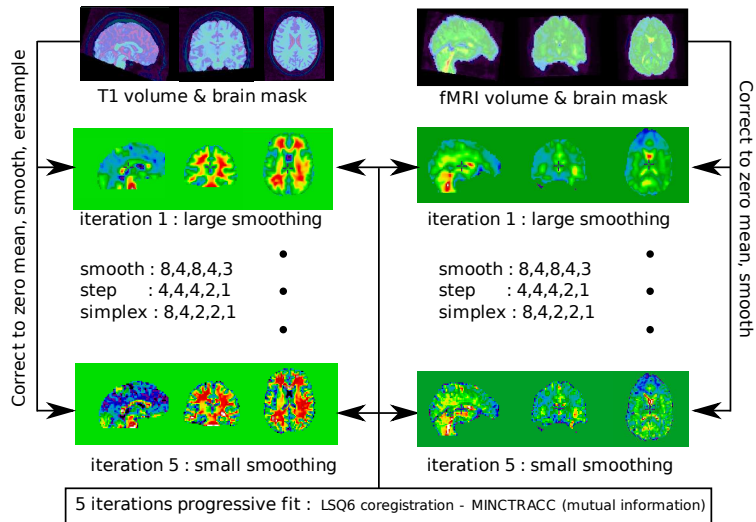
% Suggested values :

% 200 for 1.5T images,

% 50 for 3T images.

`opt.t1_preprocess.nu_correct.arg = '-distance 50';`

Coregistration between the T_1 and fMRI volumes I



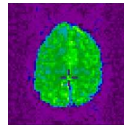
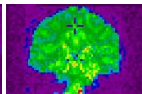
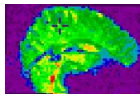
Coregistration between the T_1 and fMRI volumes II

T_1 -to-fMRI coregistration – see `niak_brick_anat2func`

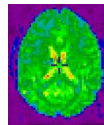
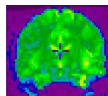
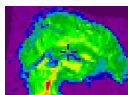
```
% An initial guess of the transform.  
% Possible values: 'identity', 'center'.  
% 'identity' starts from native space, untouched.  
% 'center' realigns centers of mass, which usually  
% does more harm than good.  
% Use it only if you have very big misrealignment  
% between the two images (say, > 2 cm).  
opt.anat2func.init = 'identity';
```

Spatial resampling I

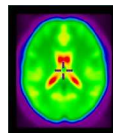
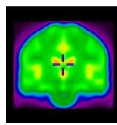
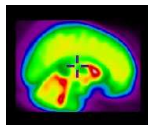
**native functional
space**



**stereotaxic space -
individual volume -
non-linear transform -**



**stereotaxic space -
average of 40 subjects**



The transformations to correct for rigid-body motion during the fMRI acquisition and the transformation to match the T_1 image and then (non-linearly) coregister into stereotaxic space are all combined, and a single step of spatial resampling is applied.

Spatial resampling II

Spatial resampling – see `niak_brick_resample_vol`

% The resampling scheme.

```
opt.resample_vol.interpolation      = 'trilinear';
```

% The voxel size to use in the stereotaxic space

```
opt.resample_vol.voxel_size        = [3 3 3];
```

% Skip resampling

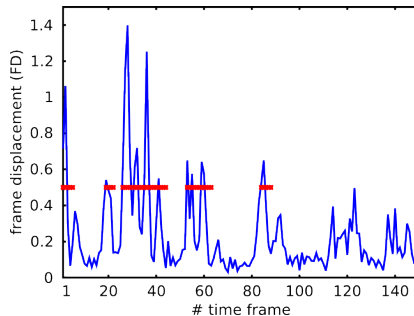
% (data will stay in native functional space

% after slice timing/motion correction)

% (0: don't skip, 1 : skip)

```
opt.resample_vol.flag_skip         = 0;
```

Scrubbing: frame displacement

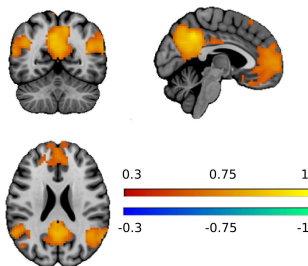


Frame displacement is the sum of absolute displacements in translation and rotation motion parameters. For each frame with excessive FD (here $FD > 0.5$), four frames are suppressed (the target one + one before + two after, marked with red stars on the figure). The original method was proposed by Power et al. Neuroimage 2012. Note that, unlike the original method, only FD is used in NIAK (and not DVARS).

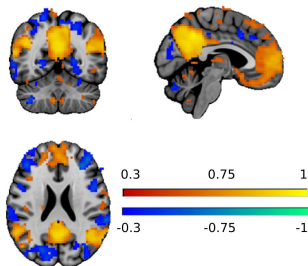
Scrubbing: example of impact on the DMN

Seed based analysis in the PCC Default mode

No scrubbing



Scrubbing 0.2



See Power et al. Neuroimage 2012 for more infos.

Scrubbing: parameters

Scrubbing – see `niak_brick_regress_confounds`

```
% Turn on/off the scrubbing of time frames
```

```
% with excessive motion
```

```
% (true: apply / false : don't apply)
```

```
opt.regress_confounds.flag_scrubbing = true;
```

```
% The threshold on frame displacement that is used to
```

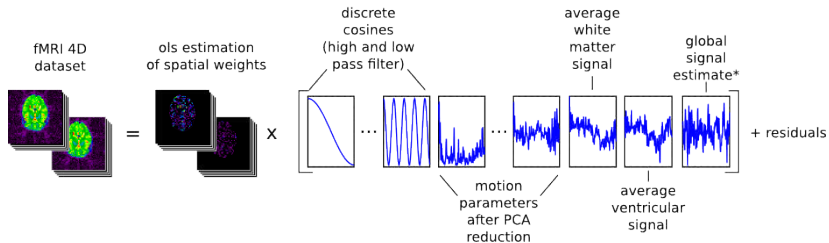
```
% determine frames with excessive motion
```

```
opt.regress_confounds.thre_fd = 0.5;
```

WARNING the scrubbing is changing the temporal grid of the dataset. Most software will not handle these changes properly. If NIAK is used to preprocess a dataset before analysis in another software package such as FSL, **it is important to skip scrubbing**.

Regress confounds: model

Regression model for confounds in resting-state fMRI



* the global signal estimate is based on a PCA decomposition, as described in Carbonell, Bellec and Shmuel, Brain connectivity 2012

Regress confounds: parameters

Regression of confounds – see `niak_brick_regress_confounds`

% Cut-off frequency for high-pass filtering (in Hz).

```
opt.time_filter.hp = 0.01;
```

% Cut-off frequency for low-pass filtering (in Hz).

```
opt.time_filter.lp = Inf;
```

% Turn on/off the regression of the average white matter signal

```
opt.regress_confounds.flag_wm = true;
```

% Turn on/off the regression of the average of the ventricles

```
opt.regress_confounds.flag_vent = true;
```

% Turn on/off the regression of the motion parameters

```
opt.regress_confounds.flag_motion_params = true;
```

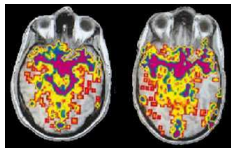
% Turn on/off the regression of the PCA-based estimation

% of the global signal

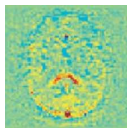
```
opt.regress_confounds.flag_gsc = false;
```


CORSICA: sources of structured noise

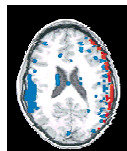
Many sources of space-time correlated noise fluctuations are corrupting fMRI time-series:



Cardiac fluctuations
Dagli et al. Neuroimage (1999)



Respiratory fluctuations
Raj et al. Phys. Med. Biol. (1999)

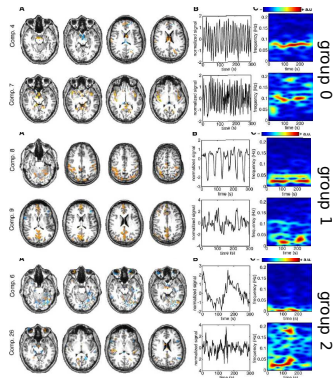
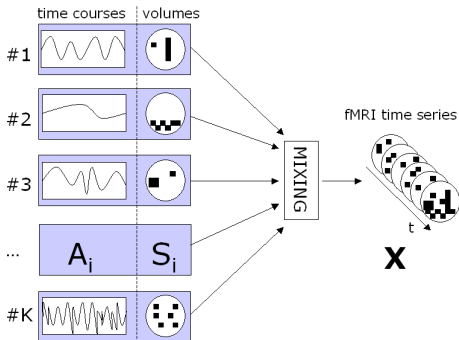


Residual motion
McKeown et al. HBM (1998)

1/f acquisition noise
space-time filtering
slice-timing correction
motion
motion correction



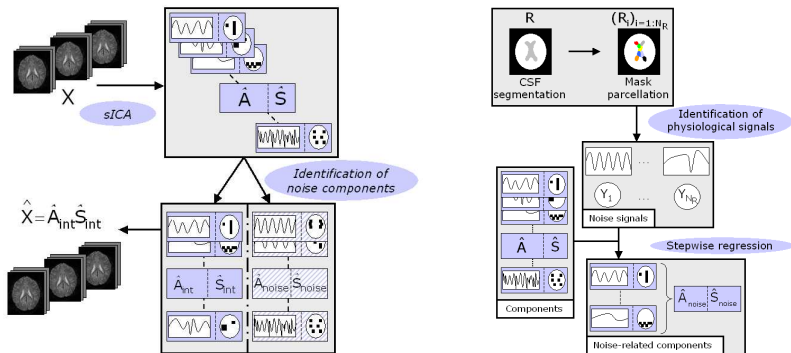
CORSICA: independent component analysis



spatially independent components analysis

Perlberg et al. Magnetic Resonance Imaging, 2007, 25: 35-46.

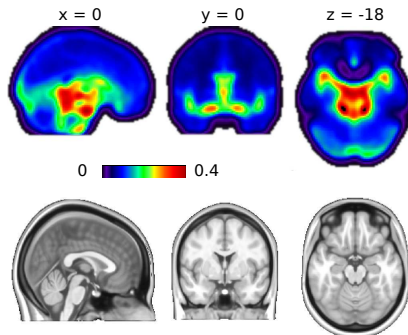
CORSICA: selection of “noise” components



Flowchart of the CORSICA algorithm for correcting structured noise in fMRI

Perlberg et al. Magnetic Resonance Imaging, 2007, 25: 35-46.

CORSICA: effect map



Relative variance of estimated structured noise using CORSICA. Average on 40 subjects, 5 tasks per subject.

P. Bellec, V. Perlberg and A. C. Evans, Magnetic Resonance Imaging, 2009, pp. 1382-1396..

CORSICA: parameters

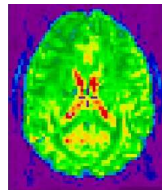
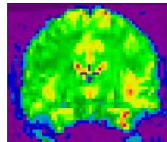
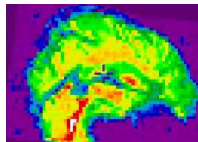
CORSICA – see `niak_pipeline_corsica`

```
% Number of components estimated during the ICA.  
% 20 at a minimum, 60 was used in the validation of CORSICA.  
opt.corsica.sica.nb_comp = 60;  
  
% Threshold for selecting noise components.  
% 0.15 has been calibrated on a validation database.  
opt.corsica.threshold    = 0.15;  
  
% Skip CORSICA (0: don't skip, 1 : skip).  
opt.corsica.flag_skip    = 1;
```

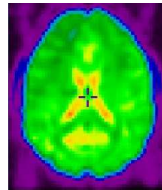
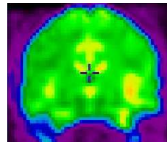
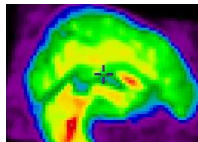
WARNING Although generally beneficial, the CORSICA selection is not stable enough to be used completely unsupervised. It is usually better to skip this step.

Spatial smoothing I

native resolution



smoothed image
isotropic Gaussian
kernel - 6 mm FWHM



Spatial smoothing II

Smoothing – see niak_brick_smooth_vol

```
% Full-width at half maximum (FWHM) of the  
% Gaussian blurring kernel, in mm.
```

```
opt.smooth_vol.fwhm      = 6;
```

```
% Skip spatial smoothing (0: don't skip, 1 : skip)
```

```
opt.smooth_vol.flag_skip = 0;
```

Quality control and assessment

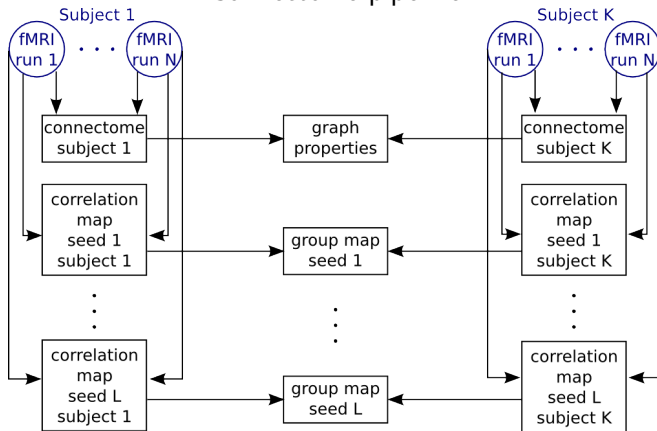
It is recommended to check the quality of the following steps:

- Coregistration of the individual T_1 image and the brain template (stereotaxic space).
- Coregistration of the individual T_1 image and the individual average BOLD volume.
- Ammount of motion.

An operations manual will be released shortly with a detailed QC procedure, as well as guidelines to fix issues when they arise.

Flowchart

Connectome pipeline



The networks and seeds input files have not been included for clarity.

User template

Grab preprocessed data

```
1 opt_g.min_nb_vol = 100;  
2 opt_g.min_xcorr_func = 0.5;  
3 opt_g.type_files = 'glm_connectome';  
4 files_in.fmri = niak_grab_fmri_preprocess('/data/fmri_preprocess',opt_g).fmri;
```

- ❶ Set the minimum acceptable number of time frames that passed scrubbing.
- ❷ Minimum spatial correlation between the individual BOLD volume and a group average (QC tool on coregistration).
- ❸ Prepare the files to feed in the connectome pipeline
- ❹ Grab the files

Input files

files_in.networks



network label

files_in.seeds

	A	B
1	roi_basc	
2	PCC	1
3	dMPFC	12
4	aMPFC	42
5	dMPFC2	46
6	TPJ	49
7	PCUN	53
8	PHC	62
9	FUS	71

number of
the network

User template ... continued

Input file: brain parcellation

```
files_in.networks = '/home/pbellec/svn/niak/template/roi_aal_3mm.mnc.gz';
```

Input file: seed regions (csv files)

```
files_in.seeds = '/home/toto/database/list_seeds.csv';
```

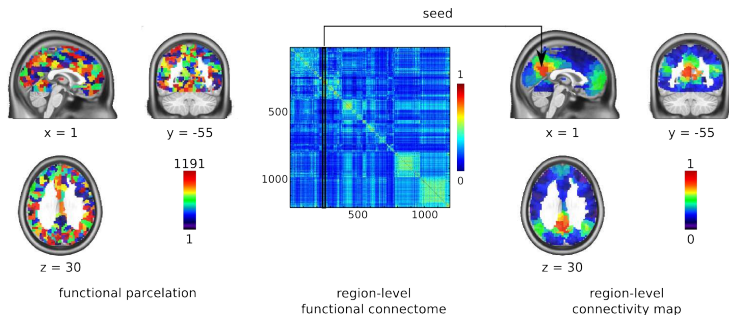
Options

```
opt.connectome.type = 'Z'; % The type of connectome.  
opt.connectome.thresh.type = 'sparsity_pos'; % how to binarize the connectome.  
opt.connectome.thresh.param = 0.2; % the cut-off on connectivity
```

Run the pipeline

```
pipeline = niak_pipeline_connectome(files_in,opt_pipe);
```

Output: individual connectomes



Saved in individual .mat files.

Output: graph properties

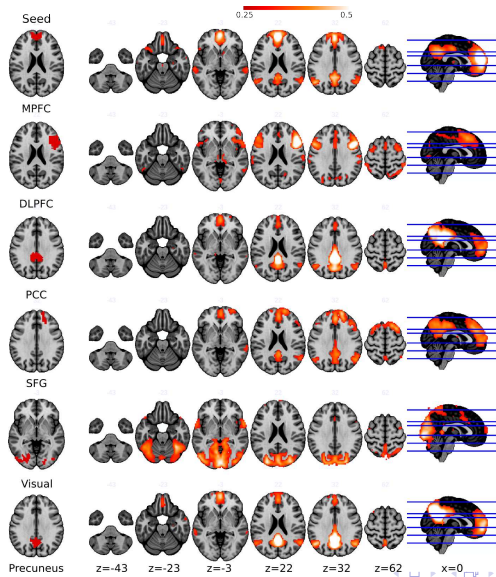
	A	B	C	D
1		Dcentrality_DLPFCI	Dcentrality_DMO	Dcentrality_FUS
2	session1_sub05676	-0.4180975675	0.1320308108	0.2420564864
3	session1_sub08224	0.7485541619	-0.1643167673	-0.5294651389
4	session1_sub08889	2.4971908638	-0.7043358847	-0.5442595472
5	session1_sub09607	1.0599716142	0.1382571671	0.1382571671
6	session1_sub14864	0.1862837144	0.6943302082	0.6943302082
7	session1_sub18604	1.1220449913	1.3059867931	1.8578121986
8	session1_sub22894	-0.5935012783	1.0648111169	1.0648111169
9	session1_sub27641	0.3672248004	0.2797903241	1.0667006108
10	session1_sub33259	1.1411152971	-0.4251213852	-0.089499239
11	session1_sub34482	1.3486391285	0.1226035571	1.1442998666
12	session1_sub36678	1.7417839427	0.3440560875	1.3117138334
13	session1_sub38579	-0.1304375846	0.3587033577	0.1956563769
14	session1_sub39529	1.1375060069	-0.8204961361	1.7901733879

• • •
graph metrics

-
- subjects
-

Graph properties are generated with the brain connectivity toolbox
<https://sites.google.com/site/bctnet/Home/functions>. See
 Rubinov and Sporns, Neuroimage 2010.

Output: connectivity maps



The pipeline system for Octave and Matlab (PSOM)

The pipeline execution is powered by a generic manager called PSOM. PSOM has many interesting features :

- **Parallel computing** : If you have access to multiple cpus or computers, PSOM can run multiple jobs in parallel.
- **Job failures** : Job failures will not crash the pipeline. You will have access to the logs to fix the problem, and restarting the pipeline will reprocess the failed jobs only.
- **Pipeline update** : If for some reason you decide to restart the pipeline after changing some options, PSOM will examine the changes made to the pipeline and restart only the jobs that need to be reprocessed. You can also add some subjects and restart the pipeline, PSOM will process only these new subjects.

More info can be found in the paper Bellec et al. Frontiers in Neuroinformatics, 2012.

Example of execution history

Example of pipeline running history on the BIC cluster for the demo NIAK dataset :

```
*****
The pipeline PIPE is now being processed.
Started on 25-Mar-2009 14:53:32
user: pbellec, host: zeus, system: unix
*****
25-Mar-2009 14:53:34 - The job anat_subject1 has been submitted to the queue (1 jobs in queue).
25-Mar-2009 14:53:35 - The job anat_subject2 has been submitted to the queue (2 jobs in queue).
25-Mar-2009 14:53:36 - The job motion_correction_subject1 has been submitted to the queue (3 jobs in queue).
25-Mar-2009 14:53:37 - The job motion_correction_subject2 has been submitted to the queue (4 jobs in queue).
.....
25-Mar-2009 15:05:49 - The job motion_correction_subject1 has been successfully completed (3 jobs in queue).
25-Mar-2009 15:05:50 - The job sica_subject1_run1 has been submitted to the queue (4 jobs in queue).
25-Mar-2009 15:05:51 - The job sica_subject1_run2 has been submitted to the queue (5 jobs in queue).
25-Mar-2009 15:06:41 - The job motion_correction_subject2 has been successfully completed (4 jobs in queue).
25-Mar-2009 15:06:51 - The job sica_subject2_run1 has been submitted to the queue (5 jobs in queue).
25-Mar-2009 15:06:52 - The job sica_subject2_run2 has been submitted to the queue (6 jobs in queue).
25-Mar-2009 15:08:07 - The job sica_subject1_run1 has been successfully completed (5 jobs in queue).
25-Mar-2009 15:09:04 - The job sica_subject2_run2 has been successfully completed (4 jobs in queue).
25-Mar-2009 15:09:19 - The job sica_subject1_run2 has been successfully completed (3 jobs in queue).
25-Mar-2009 15:09:31 - The job sica_subject2_run1 has been successfully completed (2 jobs in queue).
.....
(.. Some history lines were omitted to fit everything on one slide ...)
*****
The processing of the pipeline was completed.
25-Mar-2009 16:05:46
*****
All jobs have been successfully completed.
```

Configuration of the pipeline manager

A complete list of options for this step can be found in the web tutorial :

<http://code.google.com/p/psom/wiki/ConfigurationPsom>.

The configuration can be set up by editing a configuration file.

PSOM

```
% Number of jobs that can run in parallel.  
% This is usually the number of cores.  
opt.psom.max_queued          = 2;
```

Installation I

Downloading and installing the NIAK library

The latest stable version can be found here:

http://www.nitrc.org/frs/?group_id=411

Once the library has been decompressed, all you need to do is to start a Matlab or Octave session and add the NIAK path (with all its subfolders) to your search path, example :

```
path_niak = '/home/pbellec/public/niak/';  
P = genpath(path_niak);  
addpath(P);
```

Current requirements : Matlab 7+/Octave 3+ and Linux.

More detailed instructions can be found at :

<http://www.nitrc.org/plugins/mwiki/index.php/niak:Installation>

Installation II

Demo data

There is also a small demo dataset you can download in various formats at http://www.nitrc.org/frs/?group_id=411

The main functions available in NIAK have demonstrations (invoked by `niak_demo_`) that run on this data. You can either copy the demo data in the `/niak/data_demo/` folder (default location) or in an arbitrary folder that will need to be passed as an argument to the demo functions.

Minc tools

For most operations you will need to install a version of the MINC tools. The MINC tools are publicly available for LINUX and MAC OSX at <http://packages.bic.mni.mcgill.ca/>

Instructions for installation can be found at :

<http://en.wikibooks.org/wiki/MINC/Installation>

Who contributed to NIAK ? People

The kit was originally designed by Pierre Bellec in the lab of Alan C. Evans, Canada, 2008-10. The following people contribute to NIAK, either through code or ideas :

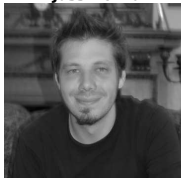
Pierre Bellec



Oliver C. Lyttelton



Jussi Tohka



Vincent Perlbarg



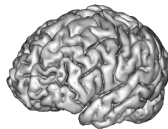
Felix Carbonell



Vladimir Fonov



Andrew Janke



Claude Lepage



Alan C. Evans

Who contributed to NIAK ? Institutions

The following institutions support the authors of NIAK. NITRC and Google Code are generously hosting the project.



McGill University



Montreal Neurological Institute



TAMPERE UNIVERSITY OF TECHNOLOGY



Who contributed to NIAK ? Software and testing

NIAK is including or depending on a number of software :

- ① The MINC tools that have been developed by members and collaborators of the MNI over the past 15 years.
- ② The linear model analysis is a port of the fMRIstat project developed by the late Keith Worsley, who will be sorely missed.
- ③ Some functions were based upon existing open-source software. See the NIAK website for a detailed list of contributions.

A lot of people have been involved in beta-testing the project and gave very precious feedback over the past two years. A non-exhaustive list includes Benjamin D'hont, Pr Christophe Grova's lab, Pr Jean Gotman's lab, Pr Alain Dhager's lab, Pr Pedro Rosa-Neto's lab and Sébastien Lavoie-Courchesne.

Useful links



1

The download page, with this pdf presentation, NIAK releases and the demo dataset

http://www.nitrc.org/frs/?group_id=411



2

The NIAK online user's guide

<http://www.nitrc.org/plugins/mwiki/index.php/niak:MainPage>



3

The NIAK project page and developer's guide

<http://code.google.com/p/niak/>



4

The PSOM project page

<http://code.google.com/p/psom/>

License

The NIAK project is under an MIT opensource license

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.