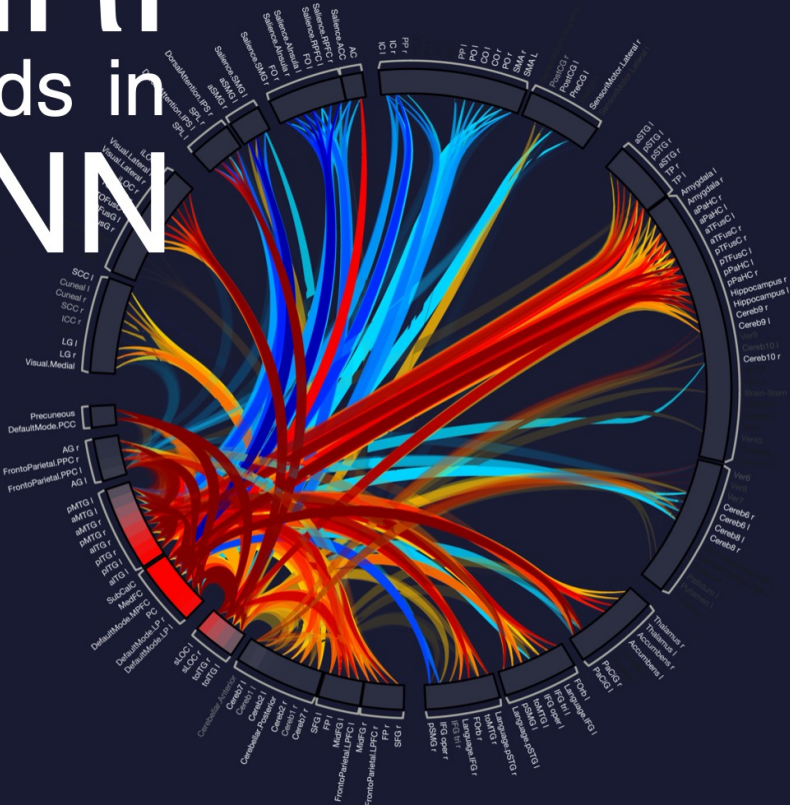


Handbook of fcMRI methods in CONN



ALFONSO NIETO-CASTANON

Handbook of functional connectivity Magnetic Resonance Imaging methods in CONN

Alfonso Nieto-Castanon^{1,2} 2020

^[1] Department of Speech, Language, and Hearing Sciences, Boston University, Boston, MA, United States

^[2] Department of Psychology, Northeastern University and McGovern Institute for Brain Research, Boston, MA, United States

Hilbert Press
Copyright © 2020 Alfonso Nieto-Castanon

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage or retrieval) without permission in writing from the author.

ISBN: 978-0-578-64400-4

Contents

1. FMRI MINIMAL PREPROCESSING PIPELINE 3

1.1. DEFAULT PREPROCESSING PIPELINE 5

Functional realignment and unwarp 5

Slice-Timing Correction 6

Outlier identification 7

Direct segmentation and normalization 8

Functional smoothing 9

1.2. ALTERNATIVES TO THE DEFAULT PREPROCESSING PIPELINE 10

Indirect segmentation and normalization 11

Functional/anatomical coregistration 12

Resample to fsaverage space 13

Functional surface-level smoothing 13

Option 1: using CONN's gui 14

Option 2: using CONN's batch commands 15

Option 3: using CONN's modular functions 15

2. FMRI DENOISING PIPELINE 17

2.1. DEFAULT DENOISING PIPELINE 18

Linear regression 18

Temporal band-pass filtering 19

2.2. EVALUATING DENOISING OUTPUTS 20

2.3. ALTERNATIVES TO THE DEFAULT DENOISING PIPELINE 22

2.4. HOW TO RUN CONN DENOISING PIPELINE 24

Option 1: using CONN's gui 24

Option 2: using CONN's batch commands 24

Option 3: using CONN's modular functions 25

3. FUNCTIONAL CONNECTIVITY MEASURES 26

3.1. SEED-BASED CONNECTIVITY MEASURES 27

Seed-Based Connectivity (SBC) maps 27

Multivariate Seed-Based Connectivity (mSBC) maps 28

Weighted Seed-Based Connectivity (wSBC) maps 30

Generalized Psycho-Physiological Interactions (gPPI) maps 31

3.2. HOW TO COMPUTE SBC MAPS 33

Option 1: using CONN's gui 33

Option 2: using CONN's batch commands 33

3.3. ROI-TO-ROI CONNECTIVITY MEASURES 35

ROI-to-ROI Connectivity (RRC) matrices 35

Multivariate ROI-to-ROI Connectivity (mRRC) matrices 37

Weighted ROI-to-ROI Connectivity (wRRC) matrices 37

Generalized Psycho-Physiological Interaction (gPPI) matrices 38

3.4. HOW TO COMPUTE RRC MATRICES 40

Option 1: using CONN's gui 40

Option 2: using CONN's batch commands 41

3.5. NETWORK MEASURES (VOXEL-LEVEL) 42

Intrinsic Connectivity (IC) 42

Global Correlation (GCOR) 43

Local Correlation (LCOR) 44

Multivariate Correlation (MCOR) (group-MVPA) 46

Independent Component Analyses (group-ICA) 48

Principal Component Analyses (group-PCA) 49

Notes on dimensionality reduction 49

3.6. HOW TO COMPUTE NETWORK MEASURES 51

Option 1: using CONN's gui 51

Option 2: using CONN's batch commands 52

3.7. GRAPH MEASURES (ROI-LEVEL) 53

Degree & Cost 53

Average path distance 54

Clustering Coefficient 54

Global Efficiency 55

Local Efficiency 55

Betweenness Centrality 56

3.8. DYNAMIC CONNECTIVITY MEASURES 57

Sliding window analyses 57

Dynamic Independent Component Analyses (dyn-ICA) 59

3.9. OTHER CONNECTIVITY MEASURES 61

Amplitude of Low-Frequency Fluctuations (ALFF) 61

Fractional Amplitude of Low-Frequency Fluctuations (fALFF) 62

4. GENERAL LINEAR MODEL 63

4.1. DEFINITION AND ESTIMATION 64

4.2. HYPOTHESIS TESTING 66

4.3. GLM MODEL AND CONTRAST SPECIFICATION 71

4.4. NUMERICAL EXAMPLES 75

4.5. HOW TO RUN CONN GENERAL LINEAR MODEL ANALYSES 80

Option 1: using CONN's gui 80

Option 2: using CONN's batch commands 81

Option 3: using CONN's modular functions 81

5. CLUSTER-LEVEL INFERENCES 83

5.1. CLUSTER-LEVEL INFERENCES IN VOXEL-BASED ANALYSES 84

Random Field Theory (RFT) parametric statistics 85

Randomization/permutation nonparametric statistics 87

Threshold Free Cluster Enhancement (TFCE) statistics 90

5.2. CLUSTER-LEVEL INFERENCES IN ROI-TO-ROI ANALYSES 93

Functional Network Connectivity (FNC) multivariate parametric statistics 94

Randomization/permutation Spatial Pairwise Clustering (SPC) statistics 96

Threshold Free Cluster Enhancement (TFCE) statistics 98

Alternatives to cluster-level inferences: connection-, ROI-, and network- level inferences 100

5.3. HOW TO USE CLUSTER-LEVEL INFERENCES 103

Option 1: using CONN's gui 103

Option 2: using CONN's commands 104

Introduction

This handbook describes methods for processing and analyzing functional connectivity Magnetic Resonance Imaging (fcMRI) data. The description is centered around those methods implemented in the CONN toolbox (Whitfield-Gabrieli and Nieto-Castanon, 2012), many of which have been improved and extended over the course of the years since CONN was first released.

The first section (**fmRI minimal preprocessing pipeline**) describes standard and advanced preprocessing steps in fcMRI. These steps are aimed at correcting or minimizing the influence of well-known factors affecting the quality of functional and anatomical MRI data, including effects arising from subject motion within the scanner, temporal and spatial image distortions due to the sequential nature of the scanning acquisition protocol, and inhomogeneities in the scanner magnetic field, as well as anatomical differences among subjects.

Even after these conventional preprocessing steps, the measured blood-oxygen-level-dependent (BOLD) signal often still contains a considerable amount of noise from a combination of physiological effects, outliers, and residual subject-motion factors. If unaccounted for, these factors would introduce very strong and noticeable biases in all functional connectivity measures. The second section (**fmRI denoising pipeline**) describes standard and advanced denoising procedures in CONN that are used to characterize and remove the effect of these residual non-neural noise sources.

Functional connectivity Magnetic Resonance Imaging studies attempt to quantify the level of functional integration across different brain areas. The third section (**functional connectivity measures**) describes a representative set of functional connectivity measures available in CONN, each focusing on different indicators of functional integration, including seed-based connectivity measures,

ROI-to-ROI measures, graph theoretical approaches, network-based measures, and dynamic connectivity measures.

Second-level analyses allow researchers to make inferences about properties of groups or populations, by generalizing from the observations of only a subset of subjects in a study. The fourth section (**General Linear Model**) describes the mathematics behind the General Linear Model (GLM), the approach used in CONN for all second-level analyses of functional connectivity measures. This description includes GLM model definition, parameter estimation, and hypothesis testing framework, as well as several practical examples and general guidelines aimed at helping researchers use this method to answer their specific research questions.

The last section (**cluster-level inferences**) details several strategies implemented in CONN allowing researchers to make meaningful inferences from their second-level analysis results while providing appropriate family-wise error control (FWEC), whether in the context of voxel-based measures, such as when studying properties of seed-based maps across multiple subjects, or in the context of ROI-to-ROI measures, such as when studying properties of ROI-to-ROI connectivity matrices across multiple subjects.

The CONN toolbox stable release is publicly available from NITRC website (www.nitrc.org/projects/conn), and updated tutorials, documentation, general information, and software development versions are available from CONN website (www.conn-toolbox.org). All examples shown in this document have been generated using the publicly available *Cambridge_Buckner* dataset part of the 1000 Functional Connectomes (fcon_1000.projects.nitrc.org)

1. fMRI minimal preprocessing pipeline

Functional Magnetic Resonance Imaging (fMRI) data, right out of the scanner, will usually be influenced by a number of well-known factors, including effects due to subject motion within the scanner, spatial and temporal distortions related to the specificities of the fMRI acquisition protocol, as well as unavoidable physiological and other non-neural effects on the measured blood-oxygen-level-dependent (BOLD) signal. Because these effects are ubiquitous, a number of relatively standard series of procedures have been developed to attempt to remove or at least ameliorate the effect of these well-known factors. These procedures are typically framed under the general umbrella of *preprocessing steps*, and consist of a series of sequential steps, each attempting to correct for one or several of these factors, that are applied to the raw functional and anatomical data before proceeding to any form of functional activation or connectivity analysis of these data.

While the specificities of these procedures vary across different software implementations (e.g. [SPM](#), [FSL](#), [AFNI](#), etc.) their general intent and approach is similar. A minimal preprocessing pipeline across any software package will typically include steps designed to: a) estimate and correct effects derived from subject motion within the scanner (realignment); b) estimate and correct spatial distortions due to inhomogeneities in the magnetic field within the scanner (susceptibility distortion correction); c) correct for temporal distortions due to the intrinsic sequential nature of the scanning

acquisition protocol (slice timing correction); and d) align functional and anatomical data across different subjects in a way that attempts to minimize the influence of anatomical differences among them (co-registration and normalization). The following section describes these individual preprocessing steps focusing on SPM software implementation as part of CONN's preprocessing pipeline.

1.1. Default preprocessing pipeline

CONN's default preprocessing pipeline (labeled "default preprocessing pipeline for volume-based analyses (direct normalization to MNI-space)" in CONN's gui, or '*default_mni*' in CONN's batch commands), performs the following steps: functional realignment and unwarp; slice-timing correction; outlier identification; direct segmentation and normalization; and functional smoothing

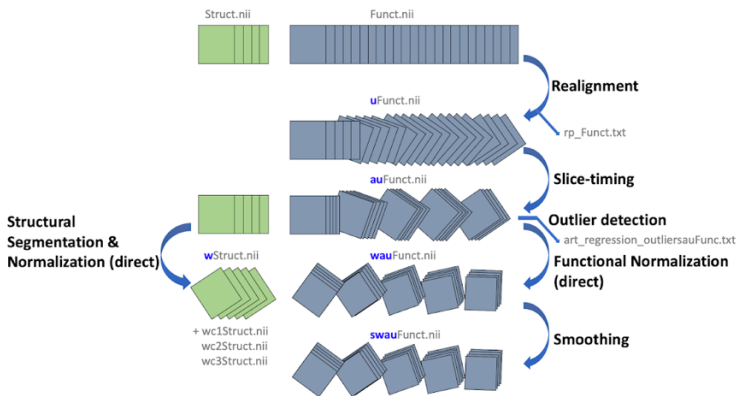


Figure 1. Schematic illustration of default minimal preprocessing pipeline in CONN

Functional realignment and unwarp

Functional data is realigned using SPM12 *realign & unwarp* procedure (Anderson et al. 2001), where all scans are coregistered and resampled to a reference image (first scan of the first session) using b-spline interpolation. This procedure also addresses potential susceptibility distortion-by-motion interactions by estimating the derivatives of the deformation field with respect to head movement, and resampling the functional data to match the deformation field of the reference image. If a double-echo sequence is available, the field inhomogeneity inside the scanner (fieldmap) is also estimated and

used for **Susceptibility Distortion Correction** (SDC) as part of the unwarp step, where the functional data is resampled along the phase-encoded direction in order to correct the absolute deformation state of the reference image caused by field inhomogeneities within the scanner

- Preprocessing step name: *functional_realign&unwarp (&fieldmap)*
- Inputs: functional data (in *primary* dataset)
- Optional inputs: double-echo sequence (one magnitude and one phase-difference image, in secondary *fmap* dataset), or direct fieldmap file in Hz units (in secondary *fmap* dataset), or pre-computed voxel-displacement map (in secondary *vdm* dataset). Double-echo timing information can be entered manually or automatically read from sidecar .json files
- Outputs: realigned functional data (in *primary* dataset). New reference image for functional data (average across all scans after realignment). Estimated motion parameters (as a .txt file for each subject/session, and also as a first-level covariate labeled 'realignment')
- See also: *#rtm*, *#vdm_t1*, *#vdm_t2*, *#vdm_ert*, *#vdm_blip*, *#vdm_type*, *#vdm_fmap* fields for advanced options

Slice-Timing Correction

Temporal misalignment between different slices of the functional data, introduced by the sequential nature of the fMRI acquisition protocol, is corrected using SPM12 slice-timing correction (STC) procedure (Henson et al. 1999), where the functional data is time-shifted and resampled using sinc-interpolation to match the time in the middle of each TA (acquisition time)

- Preprocessing step name: *functional_slicetime*
- Inputs: functional data (in *primary* dataset). Slice acquisition order information (or slice acquisition time, for simultaneous multislice acquisitions) can be entered manually, selected from a list of common acquisition sequences, or automatically read from sidecar .json file

- Outputs: STC-corrected functional data (in *primary* dataset)
- See also: `#sliceorder`, `#ta` fields for advanced options

Outlier identification

Potential outlier scans are identified from the observed global BOLD signal and the amount of subject-motion in the scanner. Acquisitions with framewise displacement above 0.9mm or global BOLD signal changes above 5 s.d. are flagged as potential outliers (note: an alternative "conservative" setting in CONN uses 0.5mm and 3 s.d. thresholds, while an alternative "liberal" setting uses 2mm and 9 s.d. thresholds). Framewise displacement is computed at each timepoint by considering a 140x180x115mm bounding box around the brain and estimating the largest displacement among six control points placed at the center of this bounding-box faces. Global BOLD signal change is computed at each timepoint as the change in average BOLD signal within SPM's global-mean mask scaled to standard deviation units

- Preprocessing step name: *functional_art*
- Inputs: functional data (in *primary* dataset). estimated head-motion parameters (created during *realignment* step; alternatively, manually entered as first-level covariate labeled 'realignment')
- Outputs: list of potential outliers (as a .txt file for each subject/session, and also as a first-level covariate labeled 'scrubbing'). List of scan-to-scan global BOLD change and head-motion measures (as a .txt file for each subject/session, and also as a first-level covariate labeled 'QC_timeseries'). New reference image for functional data (average across all scans except potential outlier scans)
- See also: `#art_thresholds` field for advanced options. See also `conn_convert12/1covariate` function to transform between different framewise displacement definitions (Jenkinson / Power / CONN) and/or to estimate a new list of outliers by applying different threshold values to an already preprocessed dataset

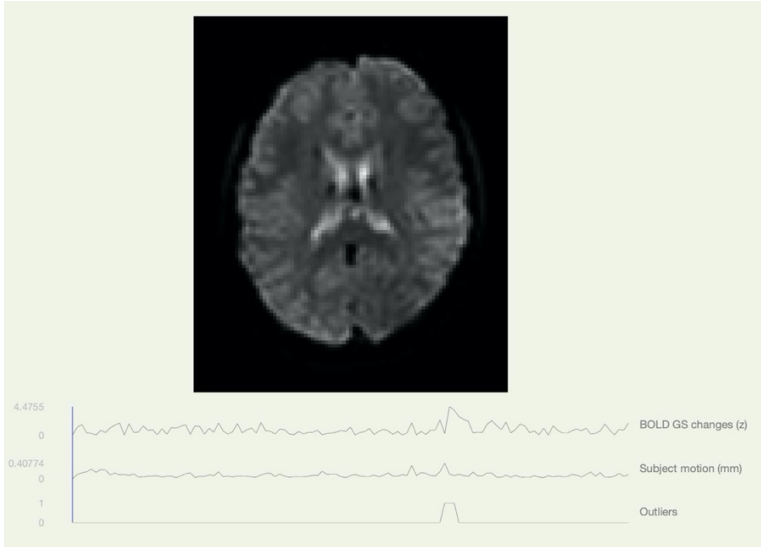


Figure 2. Estimated BOLD Global Signal change timeseries (**top**), Framewise Displacement timeseries (**middle**), and resulting identified outliers (**bottom**) for an example session from a representative subject

Direct segmentation and normalization

Functional and anatomical data are normalized into standard MNI space and segmented into grey matter, white matter, and CSF tissue classes using SPM12 unified segmentation and normalization procedure (Ashburner and Friston, 2005). This procedure iteratively performs tissue classification, estimating the posterior tissue probability maps (TPMs) from the intensity values of the reference functional/anatomical image, and registration, estimating the non-linear spatial transformation best approximating the posterior and prior TPMs, until convergence. Direct normalization applies this unified segmentation and normalization procedure separately to the functional data, using the mean BOLD signal as reference image, and to the structural data, using the raw T1-weighted volume as reference image. Both functional and anatomical data are resampled to a default 180x216x180mm bounding box, with 2mm isotropic voxels for functional data and 1mm for anatomical data, using 4th order spline interpolation

- Preprocessing step name: *functional_segment&normalize_direct*, *structural_segment&normalize*
- Inputs: functional data (in *primary* dataset), structural data
- Outputs: MNI-space functional and structural data. grey matter, white matter, and CSF masks. Skull-stripped structural volume
- See also: *#affreg*, *#boundingbox*, *#coregtomean*, *#interp*, *#tpm_template*, *#tpm_ngaus*, *#voxelsize_anat*, *#voxelsize_func* fields for advanced options

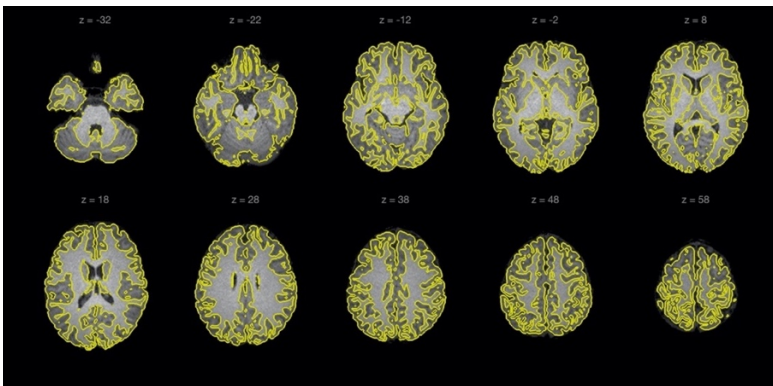


Figure 3. Example structural normalization and segmentation results in a representative subject. Structural data show in reference MNI-space, yellow lines represent 50% boundary of estimated grey matter posterior probability map

Functional smoothing

Last, functional data is smoothed using spatial convolution with a Gaussian kernel of 8mm full width half maximum (FWHM), in order to increase BOLD signal-to-noise ratio and reduce the influence of residual variability in functional and gyral anatomy across subjects

- Preprocessing step name: *functional_smooth*
- Inputs: functional data (in *primary* dataset)
- Outputs: smoothed functional data (in *primary* dataset)
- See also: *#fwhm* field for advanced options

1.2. Alternatives to the default preprocessing pipeline

While the above preprocessing pipeline is designed to offer robust performance in most common scenarios, researchers are encouraged to explore different alternatives tailored to the specificities of their scanning protocol or analysis plans. Some popular alternatives either directly implemented or supported by CONN are:

1. default preprocessing pipeline for volume-based analyses (indirect normalization to MNI-space) (also known as '**default_mnifield**' in CONN's batch commands) is an alternative preprocessing pipeline designed for cases when high-quality fieldmaps are available for susceptibility distortion correction. In those cases, it can be argued (Calhoun et al. 2017) that an *indirect normalization* procedure results in better performance compared to a *direct normalization* procedure. In a *direct normalization* procedure, the functional and anatomical data are normalized separately / independently, each using its own non-linear transformation to project the data from its original space to a common MNI-space. In contrast, in an *indirect normalization* procedure both sets use the same non-linear transformation, which is estimated using only the structural data (often with higher spatial resolution, better tissue contrast, and more anatomical detail). In particular, this pipeline uses the same sequence as CONN's default pipeline, but enforcing the usage of fieldmaps for susceptibility distortion correction during the **functional realignment & unwarp** step, as well as using an **Indirect segmentation and normalization** step instead of the original Direct segmentation and normalization step (see "indirect segmentation and normalization" section below for details about this step)

2. default preprocessing pipeline for surface-based analyses (also known as '**default_ssfield**' in CONN's batch commands) is an alternative preprocessing pipeline designed for surface-based rather than volume-based data analyses. It requires the structural data to have been already processed using FreeSurfer. The pipeline uses the same sequence as CONN's default pipeline, but: 1) enforcing the usage of fieldmaps for susceptibility distortion correction during the functional realignment & unwarp step; 2) using a functional /

anatomical coregistration step instead of the original Direct segmentation and normalization step (see "*functional/anatomical coregistration*" section below for details about this step); 3) sampling the functional data only within the cortical surface (see "*resample to fsaverage space*" section below for details); and 4) using a functional surface-based smoothing step instead of the original functional smoothing step (see "*surface-level smoothing*" section below for additional details about this step)

3. fMRIPrep preprocessing pipeline. It is also possible to use data in CONN that has been already either fully or partially preprocessed using other software packages like SPM or FSL. One popular and robust preprocessing pipeline combining multiple tools is fMRIPrep (<https://fmripred.readthedocs.io/>). When using data preprocessed by fMRIPrep, CONN can automatically import the resulting preprocessed MNI-space functional and anatomical volumes, as well as Grey/White/CSF tissue masks, a full list of potential confound regressor variables, and an associated CONN-formatted first-level realignment, scrubbing, and QC_timeseries covariates that can be directly used by CONN's default denoising procedure. After importing fMRIPrep data it is recommended to run only an additional **functional_smoothing** preprocessing step before directly proceeding to Denoising the resulting functional data (see Esteban et al. 2019 for additional details about fMRIPrep preprocessing pipeline)

Indirect segmentation and normalization

Functional data is first co-registered using an affine transformation to the structural data using SPM12 inter-modality coregistration procedure with a normalized mutual information cost function (Collignon et al. 1995, Studholme et al. 1998). Then the anatomical data is normalized into standard MNI space and segmented into grey matter, white matter, and CSF tissue classes using SPM12 unified segmentation and normalization procedure (Ashburner and Friston, 2005). This procedure iteratively performs tissue classification, estimating the posterior tissue probability maps (TPMs) from the intensity values of the reference anatomical image, and registration, estimating the non-linear spatial transformation best approximating the posterior and prior TPMs, until convergence. Indirect

normalization applies this unified segmentation and normalization procedure to the structural data using the raw T1-weighted volume as reference image, and then applies the same estimated non-linear transformation to the functional data. Both functional and anatomical data are resampled to a default 180x216x180mm bounding box, with 2mm isotropic voxels for functional data and 1mm for anatomical data, using 4th order spline interpolation

- Preprocessing step name: *functional_segment&normalize_indirect*
- Inputs: functional data (in *primary* dataset), structural data
- Outputs: MNI-space functional and structural data. Grey matter, Whiter matter, and CSF masks. Skull-stripped structural volume
- See also: *#affreg*, *#boundingbox*, *#interp*, *#tpm_template*, *#tpm_ngaus*, *#voxelsize_anat*, *#voxelsize_func* fields for advanced options

Functional/anatomical coregistration

Functional data is co-registered to the structural data using SPM12 inter-modality coregistration procedure with a normalized mutual information cost function (Collignon et al. 1995, Studholme et al. 1998). This procedure estimates an optimal affine transformation between the reference functional image (mean BOLD signal) and the reference structural image (T1-weighted volume) that maximizes the mutual information between the two, storing this information in the functional image voxel-to-world mapping header information without resampling the data

- Preprocessing step name: *functional_coregister_affine_noreslice*
- Inputs: functional data (in *primary* dataset), structural data
- Outputs: functional data voxel-to-world mapping updated
- See also: *#coregtomean* field for advanced options

Resample to fsaverage space

Functional data is resampled at the location of the subject-specific FreeSurfer-estimated cortical ribbon. This procedure computes the BOLD signal at the location of each vertex in fsaverage icosahedral level-8 tessellation (163,842 vertices and 327,680 faces per hemisphere), projected to the subject-specific white matter surface, and averaged across 10 intervals along the normal between the white matter and pial surfaces

- Preprocessing step name: *functional_surface_resample*
- Inputs: functional data in same space as FreeSurfer T1w reference (in *primary* dataset)
- Outputs: functional data in surface-level fsaverage space (in *primary* dataset)

Functional surface-level smoothing

Functional data in fsaverage space is smoothed using iterative diffusion smoothing with 40 iterations, approximately 8mm FWHM kernel (Hagler et al. 2006), in order to increase BOLD signal-to-noise ratio and reduce the influence of residual variability in functional and gyral anatomy across subjects

- Preprocessing step name: *functional_surface_smooth*
- Inputs: functional data in surface-level fsaverage space (in *primary* dataset)
- Outputs: smoothed functional data in surface-level fsaverage space (in *primary* dataset)
- See also: *#diffusionsteps* field for advanced options

1.3. How to run CONN preprocessing pipeline

CONN's default preprocessing pipeline can be run using any of the following options:

Option 1: using CONN's gui

After entering in CONN the raw functional data (in *Setup.functional*) and raw structural data (in *Setup.structural*) for one or multiple subjects, click on the '**Preprocessing**' button, select the first option/pipeline there, labeled '*default preprocessing pipeline for volume-based analyses (direct normalization to MNI-space)*' and click 'Ok' and '**Start**' (optionally change the '*local processing*' option available in that window to '*distributed processing*' if you want to parallelize this pipeline across multiple processors or nodes in an HPC cluster)

note: alternatively, if fieldmap sequence files are available, enter these (e.g. Magnitude and PhaseDiff images) in *Setup.functional* labeled as a 'fmap' secondary dataset, and then select the second or third pipelines here (for indirect or direct normalization procedures, respectively)

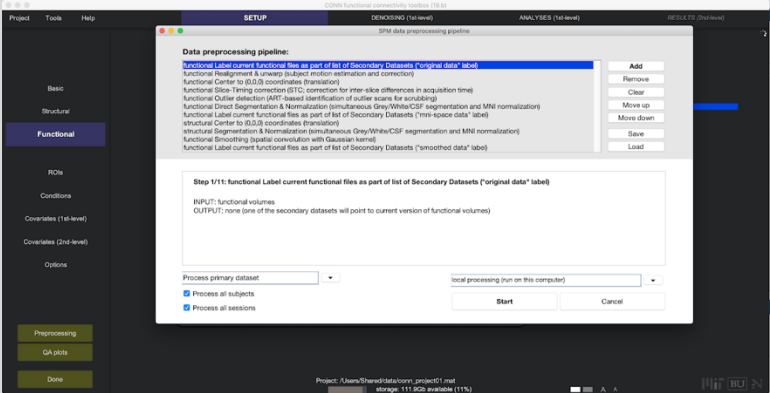


Figure 4. Example of CONN's preprocessing gui

Option 2: using CONN's batch commands

Similarly, if you have already entered your raw functional/anatomical files in CONN (either using the GUI or batch commands), you may run the default preprocessing pipeline using MATLAB command syntax:

```
conn_batch( 'Setup.preprocessing.steps', 'default_mni' )
```

optionally adding to this command any desired alternative field name/value pairs (see *doc conn_batch* for additional details), for example:

```
conn_batch( 'filename', '/data/Cambridge/conn_Cambridge.mat', ...  
            'Setup.preprocessing.steps', 'default_mni', ...  
            'Setup.preprocessing.sliceorder', 'interleaved (Siemens)', ...  
            'Setup.preprocessing.fwhm', 12 )
```

Option 3: using CONN's modular functions

If you prefer to run this preprocessing pipeline separately from the rest of CONN's functionality (e.g. to preprocess some data and then continue analyzing the preprocessed data on a different software package), you may also run CONN's default preprocessing pipeline using the following MATLAB command syntax:

```
conn_module( 'preprocessing', ...  
            'structurals', {'/data/anat.nii'}, ...  
            'functionals', {'/data/func.nii'}, ...  
            'steps', 'default_mni' )
```

optionally adding to this command any desired alternative field name/value pairs (see *doc conn_module* and *doc conn_batch* for additional details), for example:

```
conn_module( 'preprocessing', ...
             'structurals',    {'/data/anat.nii'}, ...
             'functionals',    {'/data/func.nii'}, ...
             'steps',          'default_mni', ...
             'RT',              2, ...
             'sliceorder',     'interleaved (Siemens)', ...
             'fwhm',            12 )
```

2. fMRI denoising pipeline

After the functional data has been preprocessed, the BOLD signal often still contains a considerable amount of noise or non-neural variability due to a combination of physiological, outlier, and residual subject-motion effects. These residual factors are particularly problematic in the context of functional connectivity MRI (fcMRI) analyses, because they introduce very strong and noticeable biases in all functional connectivity measures (see examples below in "evaluating denoising output" section). Because of this, conventional preprocessing steps in the context of fcMRI analyses have favored considerably more conservative strategies than those often found in activation-based fMRI analyses, focusing on eliminating or at least minimizing the influence of these residual noise components in the BOLD signal. These additional strategies are often framed under the general umbrella term of *denoising*. The following section describes these denoising procedures, focusing on CONN's implementation but also discussing alternative procedures, as well as ways to evaluate the success of different denoising strategies on each particular dataset

2.1. Default denoising pipeline

CONN's default denoising pipeline combines two steps: linear regression of potential confounding effects in the BOLD signal, and temporal band-pass filtering

Linear regression

Factors that are identified as potential confounding effects to the estimated BOLD signal are estimated and removed separately for each voxel and for each subject and functional run/session using Ordinary Least Squares (OLS) regression to project each BOLD signal timeseries to the sub-space orthogonal to all potential confounding effects. Potential confounding effects used in CONN's default denoising pipeline implement an anatomical component-based noise correction procedure (**aCompCor**), and include noise components from cerebral white matter and cerebrospinal areas (Behzadi et al. 2007), estimated subject-motion parameters (Friston et al. 1995), identified outlier scans or scrubbing (Power et al. 2014), constant and first-order linear session effects, and constant task effects, if applicable (Whitfield-Gabrieli and Nieto-Castanon, 2012):

- **noise components from white matter and cerebrospinal areas:** potential confounding effects are defined from the observed BOLD signal within each of two anatomically-defined noise areas computed by applying a one-voxel binary erosion step to the masks of voxels with values above 50% in white matter and CSF posterior probability maps. Within each area five potential noise components (Chai et al. 2012) are estimated: the first computed as the average BOLD signal, and the next four computed as the first components in a Principal Component Analysis of the covariance within the subspace orthogonal to the average BOLD signal and all other potential confounding effects
- **estimated subject-motion parameters:** a total of 12 potential noise components are defined from the estimated subject-motion parameters in order to minimize motion related BOLD variability: 3 translation and 3 rotation parameters plus their associated first-order derivatives

- **scrubbing:** a variable number of noise components (one for each identified outlier scan during the outlier identification preprocessing step) are used as potential confounding effects to remove any influence of these outlier scans on the BOLD signal
- **session and task effects:** constant and linear BOLD signal trends within each session, as well as main task or session effects convolved with a canonical hemodynamic response function, are defined as additional noise components in order to reduce the influence of slow trends, initial magnetization transients, as well as constant task-induced responses in the BOLD signal, if applicable

Temporal band-pass filtering

Temporal frequencies below 0.008 Hz or above 0.09 Hz are removed from the BOLD signal in order to focus on slow-frequency fluctuations while minimizing the influence of physiological, head-motion and other noise sources. Filtering is implemented using a discrete cosine transform windowing operation to minimize border effects, and performed after regression to avoid any frequency mismatch in the nuisance regression procedure (Hallquist et al. 2013)

2.2. Evaluating denoising outputs

The effect of denoising acting to minimize the influence of artifactual factors on functional connectivity measures can best characterized by estimating the distribution of functional connectivity values (FC) between randomly-selected pairs of points within the brain before and after denoising. Considering the BOLD signal after a standard minimal preprocessing pipeline (but before denoising), FC distributions show extremely large inter-session and inter-subject variability, and skewed distributions with varying degrees of positive biases, consistent with the influence of global or large-scale physiological and subject-motion effects. After denoising FC distributions show approximately centered distributions, with small but noticeable larger tails in the positive side, and considerably reduced inter-session and inter-subject variability

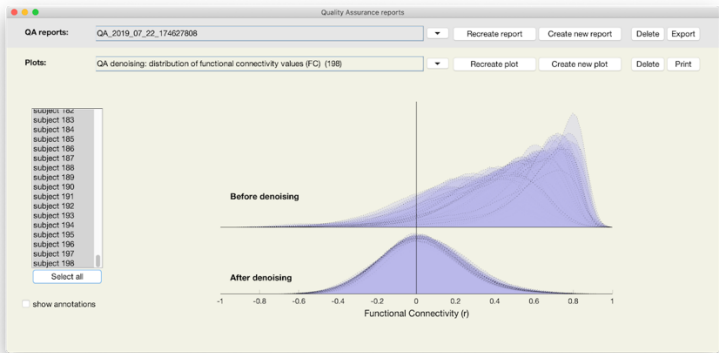


Figure 5. Distribution of Functional Connectivity (FC) values (correlation coefficients among random pairs of voxels) before (**top**) and after (**bottom**) denoising, in a representative sample study

Other useful ways to evaluate the quality of the denoising procedure outputs include computing QC-FC correlations (Ciric et al. 2017). QC-FC correlations look, again, at the connectivity values (FC) between randomly-selected pairs of points within the brain, but instead of simply displaying the distribution of these values, this method evaluates whether FC values are correlated, across subjects, with other Quality Control (QC) measures (e.g. subject-motion indicators). The method computes a QC-FC correlation value for each randomly-selected pair of points within the brain, and then it

displays the distribution of the resulting QC-FC correlation values (shown in gray before and after denoising in the plots below). In addition, this distribution can be directly compared to an associated null-hypothesis (NH) distribution (shown as red dashed lines), with an associated percent match value quantifying the degree of similarity between the two (95% or higher match with NH indicate lack of noticeable QC-FC associations)

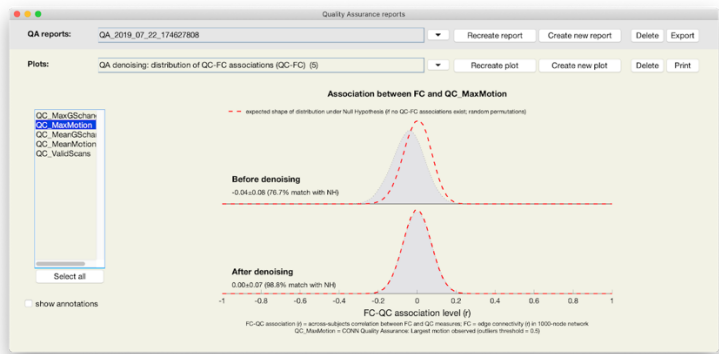


Figure 6. Distribution of between-subject correlations between Functional Connectivity values (FC) and Quality Control measures (QC) correlations before (top) and after (bottom) denoising, in a representative sample study. Red dashed line represents expected shape of this distribution in the absence of FC-QC correlations

2.3. Alternatives to the default denoising pipeline

While the above denoising pipeline is designed to remove most outlier, physiological, including respiratory and cardiac effects, and residual subject-motion effects from the BOLD signal, researchers are strongly encouraged to evaluate the quality of their data after denoising and consider modifying these or using additional denoising steps as necessary. In addition to simple variations of the default denoising strategy, such as using different scrubbing options (e.g. identifying outliers based on a lower framewise displacement threshold) or selecting a higher number of aCompCor noise components, some useful additional denoising approaches include:

1. **ICA denoising** is a data-driven approach where Independent Component Analyses are used to identify potential noise-related temporal components either manually or semi-automatically (Griffanti et al. 2017). These components can then be entered in CONN as additional potential confounding effects in the standard Linear Regression denoising step

2. **Retroicor** (Glover et al 2000) is a popular technique to use cardiac and respiratory state information recorded during the scanning session to build a series of predicted sine and cosine components of the respiratory and cardiac effects, which can then also be used in CONN as additional potential confounding effects in the standard Linear Regression denoising step

3. **Simult** (Hallquist et al. 2013) is an alternative implementation of the standard sequential regression followed by filtering approach, where both regression and filtering are implemented simultaneously as a single regression step (either by explicitly adding low- and high-frequency regressors or by pre-filtering the matrix of linear regressors). This approach can be used in CONN directly (e.g. switching the default 'RegBP' option to 'Simult'), or it can be used selectively over a subset of regressors only (by selecting the 'Filtered' option on the selected subset of potential confounding effects). Since *Simult* is equivalent to removing the frequency-specific effect of potential confounder variables (the effect estimated only within the frequency window defined by the band-pass filter) it is

recommended to not apply this filtering to confounders such as outliers which are broadband and not frequency-modulated

4. Friston²⁴ (Friston et al. 1996) is a richer set of motion-related regressors designed to remove an autoregressive-moving-average model of the effects of subject motion on the BOLD signal. In CONN this can be used simply selecting a polynomial expansion (quadratic effects) for the realignment parameters, in addition to the temporal expansion (first-order derivatives) that is already part of the default denoising strategy

5. Global Signal Regression (GSR) is an alternative approach which uses the average BOLD signal (across the entire brain) as a potential confounding effect. This approach is generally not recommended, since it can introduce artifactual biases (Murphy et al. 2009), remove potentially meaningful neural components (Chai et al. 2012), and introduce confounding effects across populations. It is nevertheless possible to use as a last resort or reference alternative approach in CONN simply defining a new ROI encompassing the entire brain (e.g. a subject-specific brain mask resulting from the outlier detection step during preprocessing, or gray matter mask resulting from the segmentation step) and using this ROI as an additional potential confounding effect in the standard Linear Regression denoising step

2.4. How to run CONN denoising pipeline

CONN's default denoising pipeline can be run using any of the following options:

Option 1: using CONN's gui

If you have already imported your data in CONN and run either one of CONN or fMRIPrep preprocessing pipelines, go to CONN's *Denoising* tab. All options here will be set by default to implement the denoising procedure described above, so simply click '**Done**' and '**Start**' to run CONN's default denoising pipeline (optionally change the '*local processing*' option available in that window to '*distributed processing*' if you want to parallelize this pipeline across multiple processors or nodes in an HPC cluster)

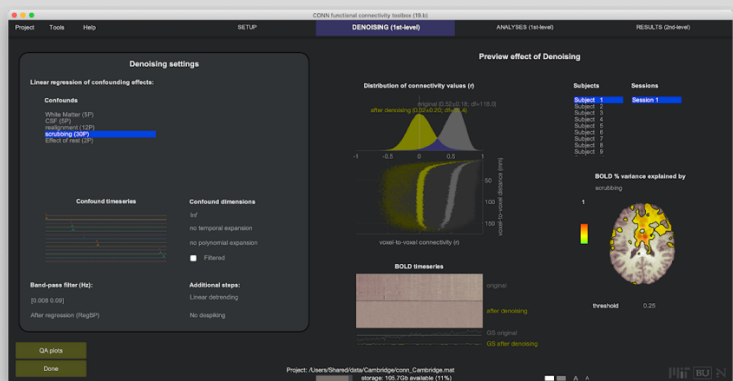


Figure 7. Example of CONN's denoising gui

Option 2: using CONN's batch commands

Similarly, if you have already entered and pre-processed your functional/anatomical files in CONN (either using the GUI or batch commands), you may run the default denoising pipeline using MATLAB command syntax:

```
conn_batch( 'Denoising.done', true )
```

optionally adding to this command any desired alternative field name/value pairs (see [doc conn batch](#) for additional details), for example:

```
conn_batch( 'filename', '/data/Cambridge/conn_Cambridge.mat', ...  
            'Denoising.done',      true, ...  
            'Denoising.filter',    [0.008 inf], ...  
            'Denoising.regbp',     2 )
```

Option 3: using CONN's modular functions

If you prefer to run this denoising pipeline separately from the rest of CONN's functionality (e.g. to preprocess and denoise some data that is then to be used by a different software package), you may also run CONN's default denoising pipeline simply as two additional preprocessing steps following CONN's minimal preprocessing pipeline or similar, using the following MATLAB command syntax:

```
conn_module( 'preprocessing', ...  
            'structurals',    {'/data/anat.nii'}, ...  
            'functionals',    {'/data/func.nii'}, ...  
            'steps',          {'default_mni', ...  
                               'functional_regression', ...  
                               'functional_bandpass'}, ...  
            'reg_names',      {'realignment',...  
                               'scrubbing',...  
                               'White Matter',...  
                               'CSF'}, ...  
            'reg_dimensions', [inf, inf, 5, 5], ...  
            'reg_deriv',      [1, 0, 0, 0], ...  
            'bp_filter',      [0.008 inf] )
```

modifying the parameter name/value pairs when appropriate (see [doc conn module](#) and [doc conn batch](#) for additional details)

3. Functional Connectivity measures

Functional connectivity Magnetic Resonance Imaging (fcMRI) attempts to quantify the level of functional integration across different brain areas by measuring the temporal correlations among the BOLD signal fluctuations in these areas. Despite the relative simplicity of its definition, there are easily hundreds of different functional connectivity metrics and analytic approaches. In this section we offer only a small window into this large variety of methods, choosing to focus only on those functional connectivity measures currently available in CONN. While this necessarily leaves out many other similarly interesting and useful approaches, it still encompasses a relatively large and representative array of methods, which are classified into the following categories: a) seed-based connectivity measures, aimed at investigating FC properties from individual a priori seeds or regions; b) ROI-to-ROI measures, aimed at investigating connectivity patterns among multiple regions; c) graph theoretical approaches, aimed at studying topological properties of functional connectivity graphs across multiple regions; d) network-based measures, aimed at studying properties of the entire whole-brain connectome; and e) dynamic connectivity measures, aimed at studying sources of temporal variability in functional connectivity patterns.

3.1. Seed-Based Connectivity measures

Seed-based connectivity metrics characterize the connectivity patterns with a pre-defined seed or ROI (Region of Interest). These metrics are often used when researchers are interested in one, or a few, individual regions and would like to analyze in detail the connectivity patterns between these areas and the rest of the brain. Perhaps the most common functional connectivity metric is seed-based connectivity maps (SBC). There are also several variations of this metric aimed at studying potential connectivity paths (mSBC), estimating condition-specific connectivity measures (wSBC), or identifying task-related modulations in event-related designs (gPPI)

Seed-Based Connectivity (SBC) maps

SBC maps represent the level of functional connectivity between a seed/ROI and every voxel or location in the brain. SBC maps are computed as the Fisher-transformed bivariate correlation coefficients between an ROI BOLD timeseries and each individual voxel BOLD timeseries:

$$r(x) = \frac{\int S(x,t)R(t)dt}{(\int R^2(t)dt \int S^2(x,t)dt)^{1/2}}$$

$$Z(x) = \tanh^{-1}(r(x))$$

where **S** is the BOLD timeseries at each voxel (for simplicity all timeseries are considered centered to zero mean), **R** is the average BOLD timeseries within an ROI, **r** is the spatial map of Pearson correlation coefficients, and **Z** is the **SBC map** of Fisher-transformed correlation coefficients for this ROI.

The connectivity pattern with an individual seed (see example below for a Medial Prefrontal Cortex seed) often encompasses the same area, indirectly quantifying the level of homogeneity within this region, but also often several other distant areas, directly quantifying inter-regional connectivity strength (e.g. connectivity with other

Default Mode Network areas such as PCC and bilateral LP areas in the example below)

Implementation notes: BOLD timeseries are preprocessed and denoised separately for each run/session, then concatenated and normalized to build the $S(x,t)$ and $R(t)$ timeseries above. Seed-based correlation analyses are defined in the first-level analyses tab, selecting 'functional connectivity (weighted GLM)' and 'Seed-to-Voxel' in the analysis type section, and 'bivariate correlation' and 'no weighting' in the analysis options section.

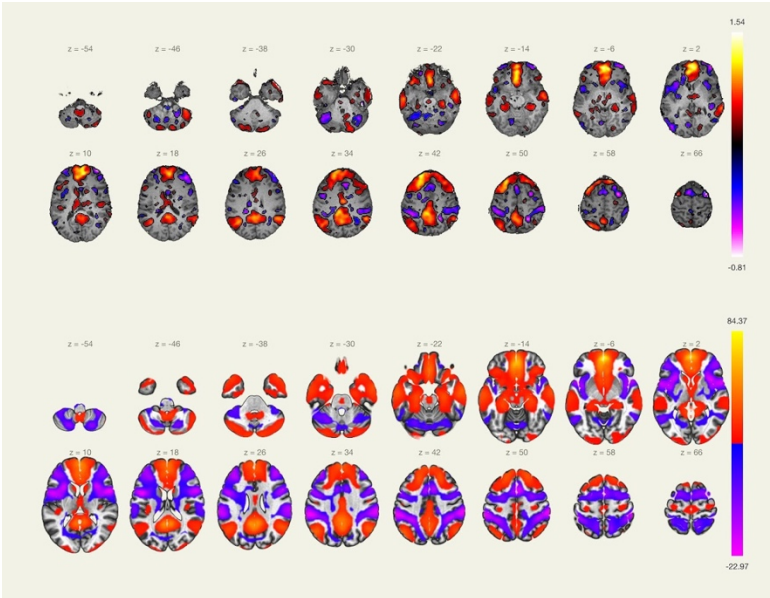


Figure 8. Example SBC map with DefaultMode.MPFC seed during rest, in single-subject (**top**; color coded Fisher-transformed correlation coefficients units), and average across 198 subjects (**bottom**; color coded one-sample T-test statistics)

Multivariate Seed-Based Connectivity (mSBC) maps

Multivariate SBC maps are computed as semipartial correlation coefficients between an ROI BOLD timeseries and each individual voxel BOLD timeseries, after controlling for one or several other ROI BOLD timeseries:

$$S(x, t) = \sum_k \beta_k(x) R_k(t) + \epsilon(x, t)$$

$$\beta_k(x) = \min_{\beta_k(x)} \int \epsilon^2(x, t) dt$$

$$Z_k(x) = \tanh^{-1}(\beta_k(x) c^{1/2})$$

$$c = \frac{\int \epsilon_k^2(t) dt}{\int S^2(x, t) dt}$$

where **S** is the BOLD timeseries at each voxel, **R** is the average BOLD timeseries within each ROI among a predefined set of ROIs, **B** is the map of multivariate regression coefficients for each ROI, estimated using an Ordinary Least Squares (OLS) solution to the above linear model, and **Z** is the **mSBC map** of Fisher-transformed semipartial correlation coefficients for each ROI

Semipartial or multivariate SBC maps are aimed at investigating potential connectivity paths. They represent the level of *effective* or *direct* connectivity between an individual seed/ROI and every voxel or location in the brain after discounting effects that may be mediated or accounted for by other ROIs (see in example below MPFC effective connectivity while controlling for other 31 network-based ROIs)

Implementation notes: Multivariate seed-based connectivity analyses are defined in the first-level analyses tab, selecting 'functional connectivity (weighted GLM)' and 'Seed-to-Voxel' in the analysis type section, and 'semipartial correlation' and 'no weighting' in the analysis options section

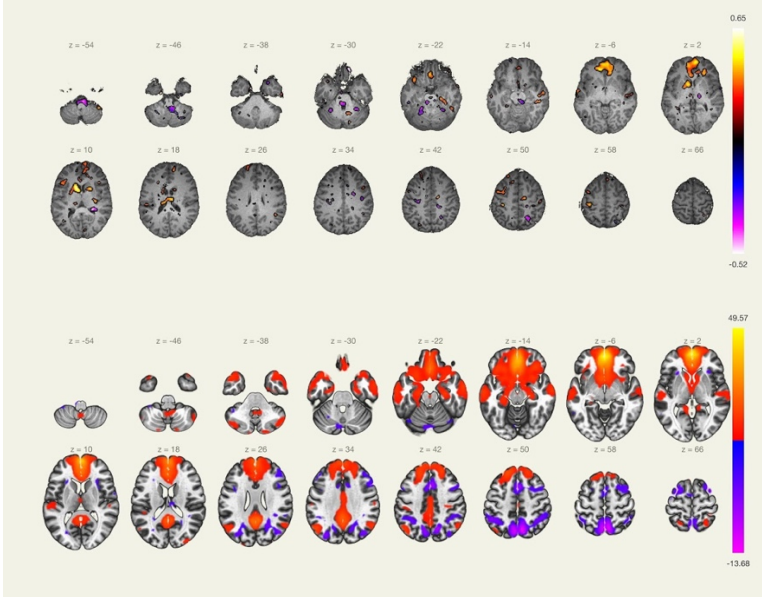


Figure 9. Example mSBC map with DefaultMode.MPFC seed during rest while controlling for all other network ROIs in single-subject (top; color coded Fisher-transformed coefficients), and average across 198 subjects (bottom; color coded one-sample T-test statistics)

Weighted Seed-Based Connectivity (wSBC) maps

Weighted SBC maps are used to characterize task- or condition-specific functional connectivity strength (i.e. functional connectivity during each task/condition). wSBC maps are computed using a weighted Least Squares (WLS) linear model with user-defined temporal weights identifying each individual experimental task/condition. In block- or event- related task designs, weights are defined as a condition-specific boxcar timeseries convolved with a canonical hemodynamic response function. In pure resting-state analyses, weights are defined to encompass entire runs or sessions (e.g. in a pre- vs. post- intervention design)

$$S(x, t) = \beta_n(x)R(t) + \epsilon(x, t)$$

$$\beta_n(x) \mid \min_{\beta_n(x)} \int w_n^2(t) \epsilon^2(x, t) dt$$

$$w_n(t) = [h_n(t) * f(t)]^+$$

$$Z_n(x) = \tanh^{-1}(\beta_n(x) c^{1/2})$$

$$c = \frac{\int R^2(t) dt}{\int S^2(x, t) dt}$$

where **S** is the BOLD timeseries at each voxel, **R** is the average BOLD timeseries within an ROI, **w** is the temporal weighting function for each condition, computed as the rectified convolution of the task/condition boxcar timeseries **h** and a canonical hemodynamic response function **f**, **B** is the map of bivariate regression coefficients for each condition estimated using a Weighted Least Squares (WLS) solution to the above linear model, and **Z** is the **wSBC map** of Fisher-transformed bivariate correlation coefficients for each task/condition.

Weighted SBC maps can be interpreted exactly in the same way as standard SBC maps, only restricted to the duration of one specific task or condition (wSBC are defined so that they are exactly equal to SBC maps when using constant weights encompassing the entire timeseries).

Implementation notes: weighted seed-based connectivity analyses are defined in the first-level analyses tab, selecting 'functional connectivity (weighted GLM)' and 'Seed-to-Voxel' in the analysis type section, and 'hrf weighting' in the analysis options section. BOLD timeseries orthogonalization to task effects is defined in the Denoising tab, selecting 'effect of task' in the confounding effects list

Generalized Psycho-Physiological Interactions (gPPI) maps

gPPI measures represent the level of task-modulated effective connectivity between a seed/ROI and every voxel or location in the brain (i.e. changes in functional association strength covarying with the external or experimental factor). They are mainly aimed at investigating task-related modulation of functional connectivity patterns in the context of event-related designs. gPPI is computed using a separate multiple regression model for each target voxel

BOLD timeseries. Each model includes as independent variables: a) all of the selected task effects convolved with a canonical hemodynamic response function (main psychological factor in PPI nomenclature); b) the seed ROI BOLD timeseries (main physiological factor in PPI nomenclature); and c) the interaction term specified as the product of (a) and (b) (PPI term). gPPI output is defined as the map of regression coefficients associated with the interaction term in these models:

$$S(x, t) = \left(\beta(x) + \sum_k \gamma_k(x) h_k^*(t) \right) R(t) + \sum_k \alpha_k(x) h_k^*(t) + \epsilon(x, t)$$

$$h_k^*(t) = h_k(t) * f(t)$$

$$\alpha_k(x), \beta(x), \gamma_k(x) \mid \min \int \epsilon^2(x, t) dt$$

where $\mathbf{S}$ is the BOLD timeseries at each voxel (for simplicity all BOLD timeseries are considered orthogonal to task effects and centered to zero mean), $\mathbf{h}$ is the task/condition boxcar timeseries which is convolved with a canonical hemodynamic response function $\mathbf{f}$, and gamma is the **gPPI map** of regression coefficients for each condition, estimated jointly with alpha and beta parameters using a least squares solution to the above linear model

Implementation notes: this implementation of gPPI in CONN is similar to that in FSL, and differs from the one in SPM, by modeling the interaction in terms of the raw BOLD signal and convolved psychological factors, rather than in terms of the deconvolved BOLD signals and raw psychological factors. Seed-based gPPI analyses are defined in the first-level analyses tab, selecting 'task modulation (gPPI)' and 'Seed-to-Voxel' in the analysis type section, and 'bivariate regression' in the analysis options section

3.2. How to compute SBC maps

CONN's SBC measures can be computed using any of the following options:

Option 1: using CONN's gui

If you have already imported and denoised your data in CONN (either through the GUI or batch commands) go to CONN's *Analyses (1st-level)* tab, and select 'Seed-to-Voxel' connectivity measures (optionally select 'Create/rename new first-level analyses' if defining multiple sets of first-level analyses). All options here will be set by default to compute SBC maps as described above, so simply click 'Done' and 'Start' to compute SBC maps for each subject, condition, and seed ROI (optionally change the 'local processing' option available in that window to 'distributed processing' if you want to parallelize this pipeline across multiple processors or nodes in an HPC cluster)

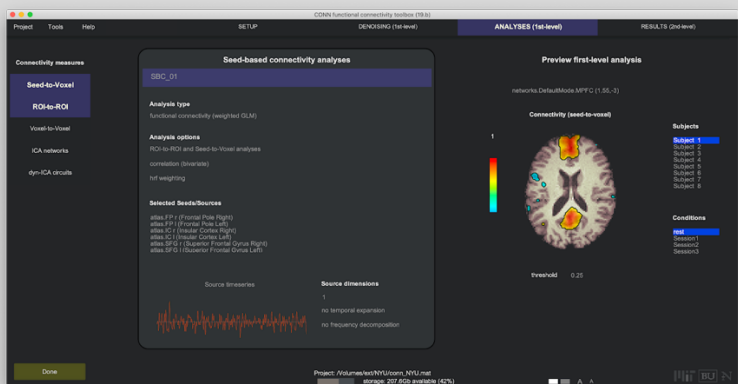


Figure 10. Example of CONN's seed-to-voxel first-level analysis gui

Option 2: using CONN's batch commands

Similarly, if you have already imported and denoised your data in CONN (either through the GUI or batch commands), you may

compute standard SBC maps across all subjects, conditions, and seed ROIs, using MATLAB command syntax:

```
conn_batch( 'Analysis.name', 'SBC', 'Analysis.done', true )
```

optionally adding to this command any desired alternative field name/value pairs (see [doc conn_batch](#) for additional details), for example:

```
conn_batch( 'filename', '/data/Cambridge/conn_Cambridge.mat', ...  
            'Analysis.name',      'mSBC', ...  
            'Analysis.type',      'seed-to-voxel', ...  
            'Analysis.measure',    'correlation (semipartial)', ...  
            'Analysis.sources',    {'networks.DefaultMode'}, ...  
            'Analysis.weight',     'none', ...  
            'Analysis.done',       true)
```

3.3. ROI-to-ROI connectivity measures

ROI-to-ROI connectivity metrics characterize the connectivity between all pairs of ROIs among a pre-defined set of regions. The definition of these metrics follows exactly the same organization and properties as the seed-based connectivity measures in the previous section, but avoids the SBC asymmetry between seeds (ROIs) and targets (voxels). These metrics are often used when researchers are interested in the simultaneously study of entire networks of connections.

ROI-to-ROI Connectivity (RRC) matrices

RRC matrices represent the level of functional connectivity between each pair of ROIs. Each element in an RRC matrix is defined as the Fisher-transformed bivariate correlation coefficient between a pair of ROIs BOLD timeseries:

$$r(i, j) = \frac{\int R_i(t)R_j(t)dt}{\left(\int R_i^2(t)dt \int R_j^2(t)dt\right)^{1/2}}$$

$$Z(i, j) = \tanh^{-1}(r(i, j))$$

where R is the BOLD timeseries within each ROI (for simplicity all timeseries here are considered centered to zero mean), r is a matrix of correlation coefficients, and Z is the **RRC symmetric matrix** of Fisher-transformed correlation coefficients

Implementation notes: ROI-to-ROI correlation analyses are defined in the first-level analyses tab, selecting 'functional connectivity (weighted GLM)' and 'ROI-to-ROI' in the analysis type section, and 'bivariate correlation' and 'no weighting' in the analysis options section

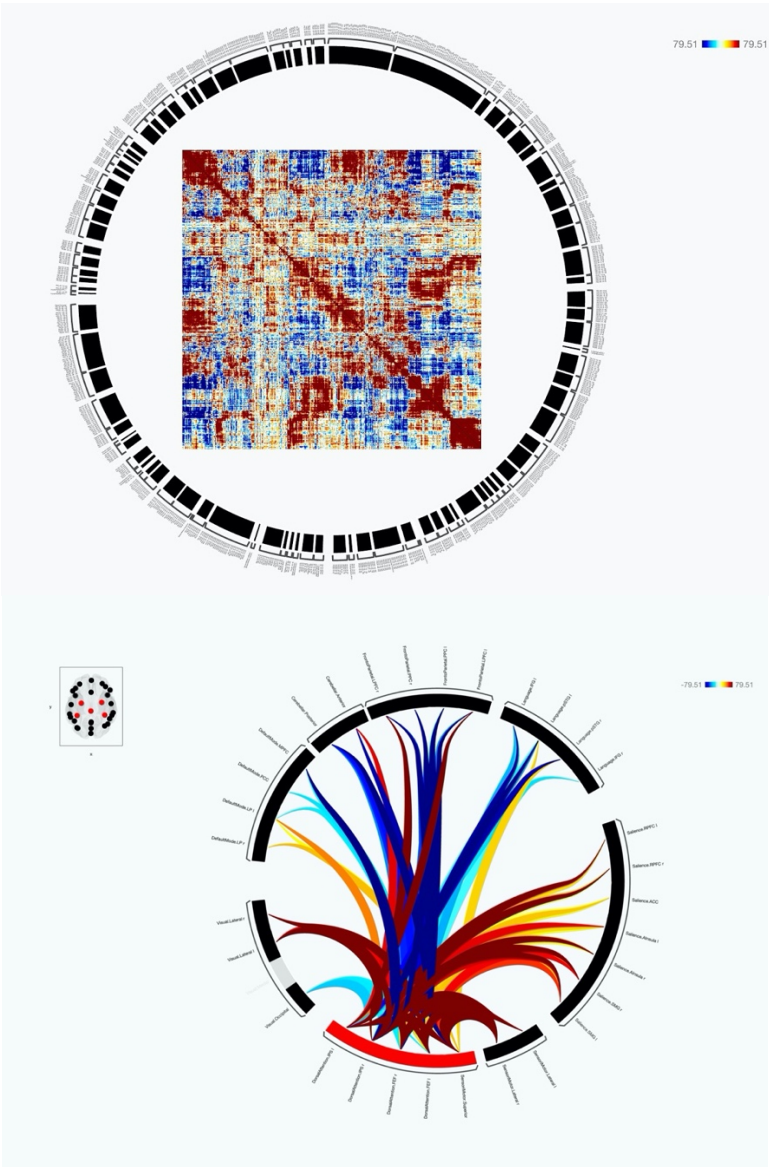


Figure 11. **Top:** Example 870-ROIs RRC matrix during rest, average across 198 subjects (color coded one-sample T-test statistics). **Bottom:** Example 8-networks 32-ROIs RRC matrix during rest, average across 198 subjects (color coded one-sample T-test statistics, highlight of connectivity with dorsal attentional network)

Multivariate ROI-to-ROI Connectivity (mRRC) matrices

Semipartial or multivariate RRC matrices represent the level of *effective* or *direct* connectivity between each pair of ROIs after discounting effects that may be mediated or accounted for by other seeds/ROIs. Each element in a mRRC matrix is defined as the semipartial correlation coefficient between each pair of ROIs among a pre-defined set of ROIs while controlling for all other ROIs in the same set:

$$R_j(t) = \sum_{i \neq j} \beta(i, j) R_i(t) + \epsilon_j(t)$$

$$\beta(i, j) | \min_{\beta(i, j)} \int \epsilon_j^2(t) dt$$

$$Z(i, j) = \tanh^{-1}(\beta(i, j) c^{1/2})$$

$$c = \frac{\int \widehat{R}_i^2(t) dt}{\int R_j^2(t) dt}$$

where $\mathbf{R}$ is the average BOLD timeseries within each ROI among a predefined set of ROIs, $\mathbf{B}$ is the matrix of multivariate regression coefficients, estimated using an Ordinary Least Squares (OLS) solution to the above linear model, and $\mathbf{Z}$ is the **mRRC matrix** of Fisher-transformed semipartial correlation coefficients among each pair of ROIs

Implementation notes: Multivariate ROI-to-ROI connectivity analyses are defined in the first-level analyses tab, selecting 'functional connectivity (weighted GLM)' and 'ROI-to-ROI' in the analysis type section, and 'semipartial correlation'/'multivariate regression' and 'no weighting' in the analysis options section

Weighted ROI-to-ROI Connectivity (wRRC) matrices

Weighted RRC measures are used to characterize task- or condition-specific functional connectivity strength (i.e. functional connectivity

during each task/condition) among a pre-defined set of ROIs. wRRC matrices are computed using a weighted Least Squares (WLS) linear model with user-defined temporal weights identifying each individual experimental task/condition. In block- or event- related task designs, weights are defined as a condition-specific boxcar timeseries convolved with a canonical hemodynamic response function. In pure resting-state analyses, weights are defined to encompass entire runs or sessions (e.g. in a pre- vs. post- intervention design):

$$\begin{aligned}
 R_j(t) &= \beta_n(i, j)R_i(t) + \epsilon_{i,j}(t) \\
 \beta_n(i, j) &| \min_{\beta_n(i, j)} \int w_n^2(t) \epsilon_{i,j}^2(t) dt \\
 w_n(t) &= [h_n(t) * f(t)]^+ \\
 Z_n(i, j) &= \tanh^{-1}(\beta_n(i, j) c^{1/2}) \\
 c &= \frac{\int R_i^2(t) dt}{\int R_j^2(t) dt}
 \end{aligned}$$

where $\mathbf{R}$ is the average BOLD timeseries within each ROI, $\mathbf{w}$ is the temporal weighting function for each condition, computed as the rectified convolution of the task/condition boxcar timeseries $\mathbf{h}$ and a canonical hemodynamic response function $\mathbf{f}$, $\mathbf{B}$ is the matrix of bivariate regression coefficients for each condition estimated using a Weighted Least Squares (WLS) solution to the above linear model, and $\mathbf{Z}$ is the **wRRC symmetric matrix** of Fisher-transformed bivariate correlation coefficients for each task/condition

Implementation notes: weighted ROI-to-ROI connectivity analyses are defined in the first-level analyses tab, selecting 'functional connectivity (weighted GLM)' and 'ROI-to-ROI' in the analysis type section, and 'hrf weighting' in the analysis options section.

Generalized Psycho-Physiological Interaction (gPPI) matrices

gPPI measures represent the level of task-modulated effective connectivity between two ROIs (i.e. changes in functional association

strength covarying with the external or experimental factor). gPPI is computed using a separate multiple regression model for each target ROI timeseries (outcome). Each model includes as predictors: a) all of the selected task effects convolved with a canonical hemodynamic response function (main psychological factor in PPI nomenclature); b) each seed ROI BOLD timeseries (main physiological factor in PPI nomenclature); and c) the interaction term specified as the product of (a) and (b) (PPI term). gPPI output is defined as the regression coefficients associated with the interaction term in these models:

$$R_j(t) = \left(\beta(i, j) + \sum_k \gamma_k(i, j) h_k^*(t) \right) R_i(t) + \sum_k \alpha_k(i, j) h_k^*(t) + \epsilon_{ij}(t)$$

$$h_k^*(t) = h_k(t) * f(t)$$

$$\alpha_k(i, j), \beta(i, j), \gamma_k(i, j) \mid \min \int \epsilon_{ij}^2(t) dt$$

where $\mathbf{R}$ is the BOLD timeseries at each ROI (for simplicity all BOLD timeseries are considered orthogonal to task effects and centered to zero mean), $\mathbf{h}$ is the task/condition boxcar timeseries which is convolved with a canonical hemodynamic response function $\mathbf{f}$, and gamma is the **gPPI matrix** of regression coefficients for each condition, estimated jointly with alpha and beta parameters using an OLS solution to the above linear model.

Implementation notes: this implementation of gPPI in CONN is similar to that in FSL, and differs from the one in SPM, by modeling the interaction in terms of the raw BOLD signal and convolved psychological factors, rather than in terms of the deconvolved BOLD signals and raw psychological factors. ROI-to-ROI gPPI analyses are defined in the first-level analyses tab, selecting 'task modulation (gPPI)' and 'ROI-to-ROI' in the analysis type section, and 'bivariate regression' in the analysis options section

3.4. How to compute RRC matrices

CONN's RRC measures can be computed using any of the following options:

Option 1: using CONN's gui

If you have already imported and denoised your data in CONN (either through the GUI or batch commands) go to CONN's *Analyses* (1st-level) tab, and select 'Seed-to-Voxel' connectivity measures (optionally select 'Create/rename new first-level analysis' if defining multiple sets of first-level analyses). All options here will be set by default to compute SBC maps as described in the previous section, so simply change the option that reads 'Seed-to-Voxel analyses only' to '**ROI-to-ROI analyses only**', and then 'click 'Done' and 'Start' to compute RRC maps for each subject, condition, and seed ROI (optionally change the 'local processing' option available in that window to 'distributed processing' if you want to parallelize this pipeline across multiple processors or nodes in an HPC cluster)



Figure 12. Example of CONN's ROI-to-ROI first-level analysis gui

Option 2: using CONN's batch commands

Similarly, if you have already imported and denoised your data in CONN (either through the GUI or batch commands) , you may compute standard RRC matrices across all subjects, conditions, and seed ROIs, using MATLAB command syntax:

```
conn_batch( 'Analysis.name', 'RRC', 'Analysis.type', 'ROI-to-ROI', ...  
            'Analysis.done', true )
```

optionally adding to this command any desired alternative field name/value pairs (see [doc conn batch](#) for additional details), for example:

```
conn_batch( 'filename', '/data/Cambridge/conn_Cambridge.mat', ...  
            'Analysis.name',      'RRC', ...  
            'Analysis.type',      'ROI-to-ROI', ...  
            'Analysis.measure',    'correlation (bivariate)', ...  
            'Analysis.sources',    {'networks.DefaultMode',...  
                                    'networks.Language'}, ...  
            'Analysis.weight',     'none', ...  
            'Analysis.done',       true)
```

3.5. Network measures (voxel-level)

Network measures attempt to summarize properties of the entire voxel-to-voxel connectome (all functional connections between every pair of voxels in the brain) into a series of reduced and interpretable measures at each individual voxel. These include measures that address properties specified a priori, and estimate how those properties are expressed in each individual subject, such as Intrinsic Connectivity (IC), Global Correlation (GCOR), and Local Correlation (LCOR), as well as data-driven measures that are first informed by group-level properties and then attempt to determine how those observed properties are expressed in each individual subject, such as Independent Component Analyses (group-ICA), Principal Component Analyses (group-PCA), and Multivariate Pattern Analyses (group-MVPA).

Intrinsic Connectivity (IC)

IC maps represent a measure of network centrality at each voxel, characterized by the strength of connectivity between a given voxel and the rest of the brain. IC is defined as the root mean square of correlation coefficients between each individual voxel and all of the voxels in the brain (Intrinsic Connectivity Contrast, Martuzzi et al. 2011):

$$IC(x) = \left(\int_{y \in M} r^2(x, y) dy \right)^{1/2}$$

where r is the map of voxel-to-voxel correlations between every pair of voxels, M is a pre-defined mask (by default covering the entire brain), and IC is the **Intrinsic Connectivity map**

Implementation notes: voxel-to-voxel connectivity matrices $r(x,y)$ are represented in terms of their Singular Value Decomposition (subject-specific SVD) with a maximum of 256 components. Separately for each voxel-to-voxel measure users may further limit the number of components used in the reconstruction of $r(x,y)$ from the singular vectors/values as a form of subject-level dimensionality reduction (see notes on dimensionality reduction section below). Users may also normalize IC measures to have a $N(0, 1)$ Gaussian distribution with zero mean and unit variance over all voxels x . Task/condition-specific IC measures are computed starting with the corresponding task/condition-specific correlation $r(x,y)$ computed using

weighted GLM. IC analyses are defined in the first-level voxel-to-voxel analyses tab, selecting 'intrinsic connectivity' in the analysis type section

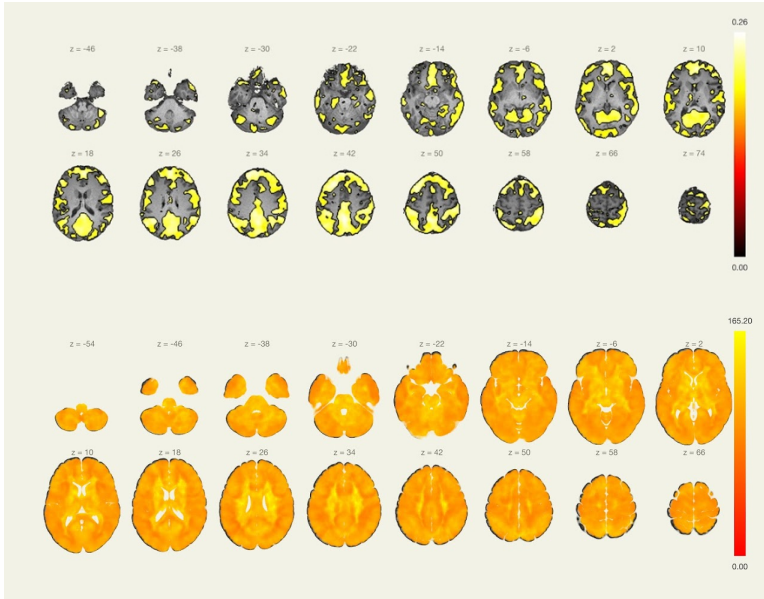


Figure 13. Example IC map during rest in single-subject (*top*; rms correlation coefficient units), and average across 198 subjects (*bottom*, color coded one-sample T-test statistics)

Global Correlation (GCOR)

GCOR maps represents a measure of network centrality at each voxel, characterized by the strength *and sign* of connectivity between a given voxel and the rest of the brain. GCOR is defined as the average of correlation coefficients between each individual voxel and all of the voxels in the brain:

$$GCOR(x) = \int_{y \in M} r(x, y) dy$$

where r is the map of voxel-to-voxel correlations between every pair of voxels, M is a pre-defined mask (by default covering the entire brain), and **GCOR** is the **Global Correlation map**. The spatial average of a GCOR map represents the GCOR quality control measure (Saad

et al. 2013) which can be used as a subject-level covariate characterizing brain-wide correlation properties

Implementation notes: same dimensionality reduction, normalization, and task/condition-specific options available as in IC measure above. GC analyses are defined in the first-level voxel-to-voxel analyses tab, selecting 'global correlation' in the analysis type section. Note: IC and GC measures are inter-related, as $IC(x)^2 - GCOR(x)^2$ equals the variability in seed-based correlations between each voxel x and the rest of the brain

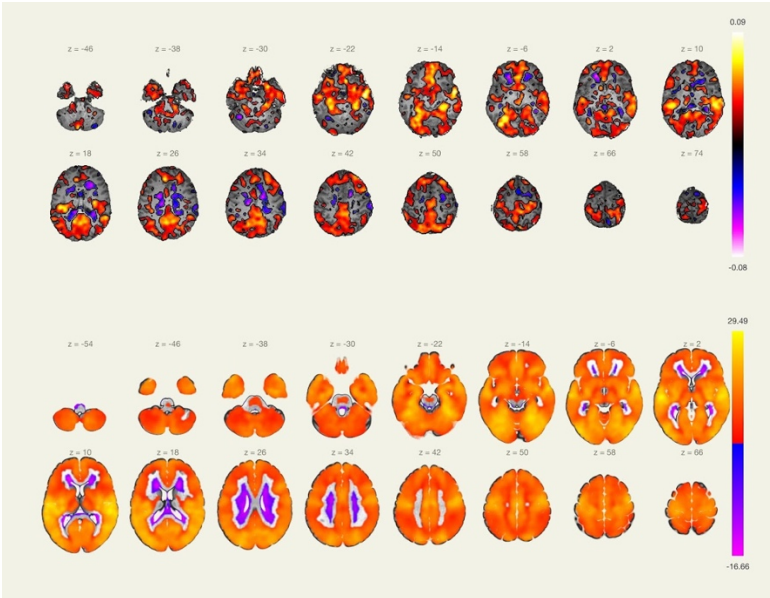


Figure 14. Example GCOR map during rest in single-subject (**top**; average correlation coefficient units), and average across 198 subjects (**bottom**; one-sample T-test statistics)

Local Correlation (LCOR)

LCOR maps represent a measure of local coherence at each voxel, characterized by the strength and sign of connectivity between a given voxel and its neighboring areas. LCOR is defined as the average of correlation coefficients between each individual voxel and a region of neighboring voxels (Integrated Local Correlation, Deshpande et al. 2009) :

$$LCOR(x) = \frac{\int w(x-y) r(x,y) dy}{\int w(x-y) dy}$$

$$w(z) = e^{-\frac{|z|^2}{2\sigma^2}}$$

where r is the map of voxel-to-voxel correlations between every pair of voxels, w is an isotropic Gaussian weighting function with size **sigma** characterizing the size of the local neighborhood, and **LCOR** is the **Local Correlation map**

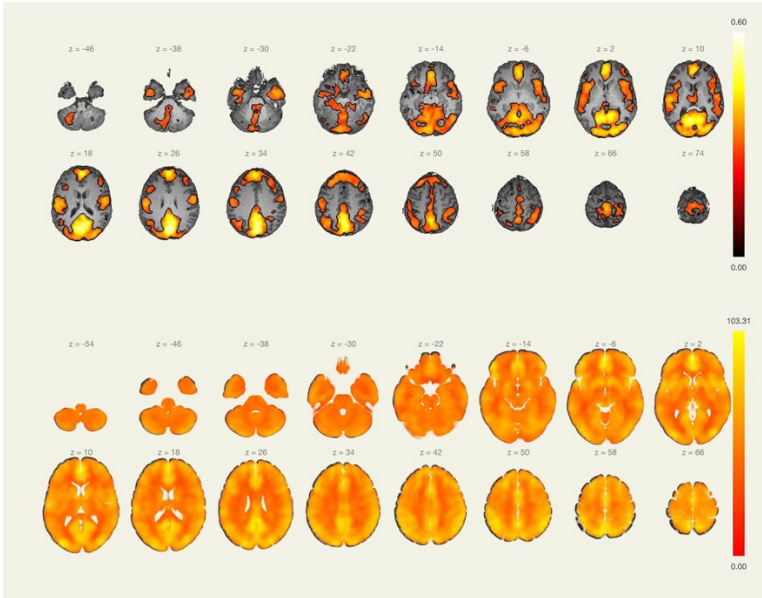


Figure 15. Example LCOR map with 25mm FWHM during rest in single-subject (**top**; average correlation coefficient units), and average across 198 subjects (**bottom**; one-sample T-test statistics)

Implementation notes: same dimensionality reduction, normalization, and task/condition-specific options available as in IC measure above. Users may choose the size of the Gaussian kernel width σ characterizing the degree of locality of the analyses. LCOR analyses are defined in the first-level voxel-to-voxel analyses tab, selecting 'local correlation' in the analysis type section. Note: LCOR and GCOR measures are inter-related, as LCOR values converge towards GCOR values for sufficiently large kernel widths σ

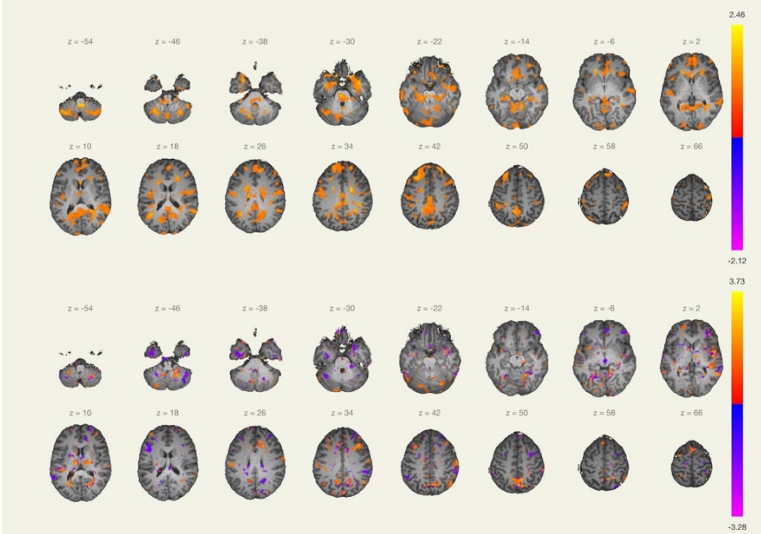
Multivariate Correlation (MCOR) (group-MVPA)

MCOR maps represent for each voxel the m most salient spatial features of the SBC maps seeded at this same voxel. MCOR maps are defined from a Singular Value Decomposition (SVD, Strang 2007), separately for each seed-voxel, of the patterns of seed-based correlations across all subjects:

$$r_n(x, y) = \sum_{i=1}^m MCOR_{n,i}(x) Q_i(x, y) + \epsilon_{n,m}(x, y)$$

$$MCOR_{n,i}(x), Q_i(x, y) \mid \min \sum_n \int_{y \in M} \epsilon_{n,i}^2(x, y) dy$$

$$\int_{y \in M} Q_i(x, y) Q_j(x, y) dy = \delta_{i,j}$$



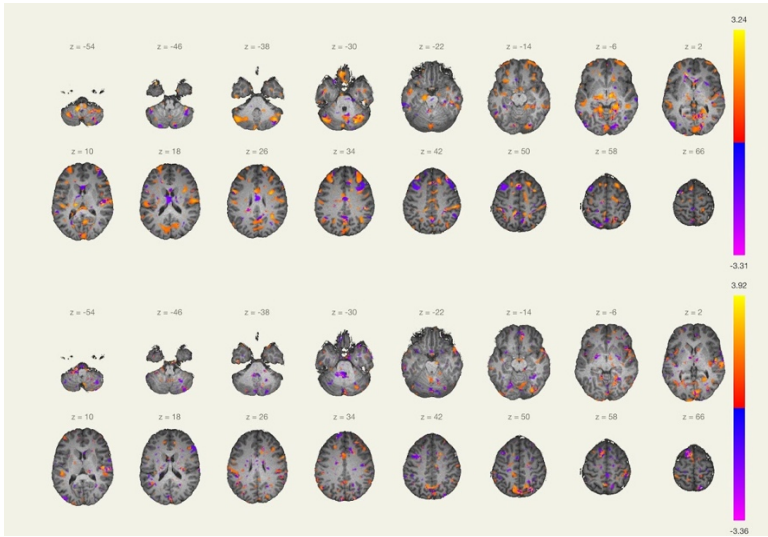


Figure 16. Example first-four MCOR maps during rest in single-subject (standard units)

where r is the map of voxel-to-voxel correlations between every pair of voxels for each subject n , $\mathbf{Q}$ is an orthogonal spatial basis characterizing the m most salient SBC spatial patterns over a pre-defined mask area $\mathbf{M}$ (by default covering the entire brain) for each seed voxel x , and **MCOR** are the m -th dimensional **Multivariate Connectivity maps** for each subject. MCOR maps can be used to perform Multivariate Pattern Analyses (Norman et al. 2006) evaluating differences between subjects in the entire patterns of seed-to-voxel connectivity.

Implementation notes: same dimensionality reduction options available as in IC measure above. MVPA analyses are defined in the first-level voxel-to-voxel analyses tab, selecting 'group-MVPA' in the analysis type section. MVPA outputs multiple Multivariate Correlation spatial maps MCOR(x), which may be used jointly in multivariate second-level analyses as a low-dimensional proxy for the entire pattern of connectivity between each voxel and the rest of the brain



Figure 17. percent of SBC maps variability explained by first-only (top) and first-four (bottom) MCOR maps during rest across 198 subjects

Independent Component Analyses (group-ICA)

ICA maps represent a measure of different networks expression and connectivity at each voxel. CONN's ICA implementation follows Calhoun's group-ICA methodology (Calhoun et al. 2001), with optional subject-level dimensionality reduction, concatenation across subjects, group-level Singular Value Decomposition for dimensionality reduction, a fastICA algorithm for group-level independent component definition (Hyvarinen 1999, with G1/tanh, G2/gauss, or G3/pow3 non-linear contrast function), and GICA1 or GICA3 for subject-level back-projection (see Calhoun et al. 2001 for a detailed description of this method)

Implementation notes: ICA analyses are defined in the first-level voxel-to-voxel analyses tab, selecting 'group-ICA' in the analysis type section. These analyses produce multiple outputs, including the individual subject-level maps $S(x,t)$ (in the second-level ICA-networks 'Spatial Properties' tab) and the variability and frequency of the timeseries $Rk(t)$ (in the second-level ICA-networks 'Temporal Properties' tab)

Principal Component Analyses (group-PCA)

CONN's PCA implementation is identical to the above Calhoun's group-ICA methodology but without the ICA rotation/weighting step, so the spatial maps represent the group-level maximal-variance components instead

Implementation notes: PCA analyses are defined in the first-level voxel-to-voxel analyses tab, selecting 'group-PCA' in the analysis type section. These analyses produce multiple outputs, including the individual subject-level maps $S(x,t)$ (in the second-level PCA-networks 'Spatial Properties' tab) and the variability and frequency of the timeseries $R_k(t)$ (in the second-level PCA-networks 'Temporal Properties' tab)

Notes on dimensionality reduction

All of the above properties are either directly or indirectly defined from r , the maps of voxel-to-voxel correlations between every pair of voxels for each subject and condition. It is often useful to represent these symmetric voxel-to-voxel correlation maps in terms of their orthogonal Singular Value Decomposition (SVD) components:

$$r(x, y) = \sum_{k=1}^m \sigma_k^2 Q_k(x) Q_k(y) + \epsilon_m(x, y)$$

$$Q_k(x) \mid \min \iint \epsilon_k^2(x, y) dx dy$$

$$\int Q_i(x) Q_j(x) dx = \delta_{i,j}$$

where r is the map of voxel-to-voxel correlations between every pair of voxels for one individual subject and condition, Q is an orthogonal spatial basis characterizing each of the m maximal-variance spatial components or eigenvectors of r , and sigma are the eigenvalues of r

characterizing the variance associated with each of these components.

When using this representation, it is also possible to remove from consideration those components explaining minimal residual variance in the BOLD signal after all standard preprocessing and denoising steps, simply by using a value of m lower than the rank of the matrix r . This is often useful for computational simplicity, but also as an additional subject-level denoising strategy, as well as to minimize potential differences in effective degrees of freedom of the residual BOLD signal across subjects. This procedure is referred to as "subject-level dimensionality reduction", and in CONN it can be optionally applied to any of the above network measures. The default recommended value in CONN keeps the first 64 components to characterize the voxel-to-voxel correlation matrix separately for each individual subject and experimental condition (see Whitfield-Gabrieli and Nieto-Castanon 2012 for method details; see Calhoun et al. 2001 for subject-level dimensionality reduction in the context of group-level ICA)

3.6. How to compute network measures

All CONN's network (voxel-level) measures can be computed using any of the following options:

Option 1: using CONN's gui

If you have already imported and denoised your data in CONN (either through the GUI or batch commands) go to CONN's *Analyses (1st-level)* tab, and select 'Voxel-to-Voxel' connectivity measures. Select 'Create new first-level analysis' and give this analysis a name (e.g. **LCOR**), then in the 'Analysis type' field select the type of network measure you would like to compute (e.g. **LocalCorrelation**). All options there will be set by default to standard values appropriate to the chosen measure. Modify these options if needed and simply click 'Done' and 'Start' to compute the corresponding maps for each subject and condition (optionally change the 'local processing' option available in that window to 'distributed processing' if you want to parallelize this pipeline across multiple processors or nodes in an HPC cluster)

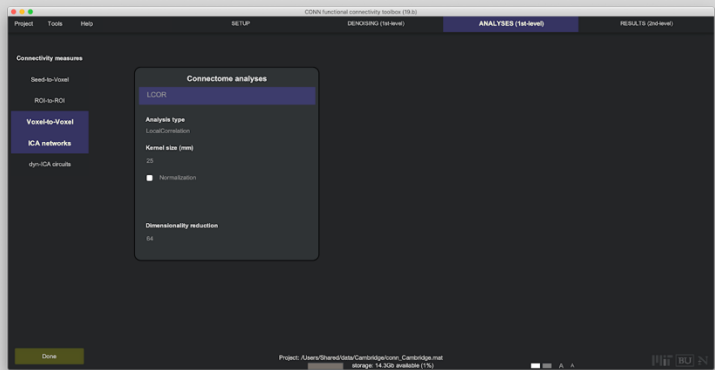


Figure 18. Example of CONN's voxel-to-voxel first-level analysis gui

Option 2: using CONN's batch commands

Similarly, if you have already imported and denoised your data in CONN (either through the GUI or batch commands) , you may compute any network maps across all subjects and conditions using MATLAB command syntax:

```
conn_batch( 'vvAnalysis.name', 'LCOR', 'vvAnalysis.measures',  
'LocalCorrelation', 'vvAnalysis.done', true )
```

optionally adding to this command any desired alternative field name/value pairs (see [doc conn batch](#) for additional details), for example:

```
conn_batch( 'filename', '/data/Cambridge/conn_Cambridge.mat', ...  
            'vvAnalysis.name', 'LCOR', ...  
            'vvAnalysis.measures.names', 'LocalCorrelation', ...  
            'vvAnalysis.measures.kernelsupport', 25, ...  
            'vvAnalysis.done', true)
```

3.7. Graph measures (ROI-level)

All **ROI-level** graph measures below are based on user-defined nondirectional graphs with nodes = ROIs, and edges = supra-threshold connections. For each subject (and condition) a graph adjacency matrix $\mathbf{A}$ is computed by thresholding the associated ROI-to-ROI Correlation (RRC) matrix $\mathbf{r}$ by an absolute (e.g. $z > 0.5$) or relative (e.g. highest 10%) threshold. Then, from the resulting graphs, a number of measures can be computed addressing topological properties of each ROI within the graph as well as of the entire network of ROIs (see Latora and Marchiori, 2001, and Achard and Bullmore, 2007, for further details about these and other graph theoretical measures)

Degree & Cost

Degree and Cost are defined, respectively, at each node as the number (degree) or proportion (cost) of edges from/to each node. Similarly, network degree/cost represent the average or the degree/cost across all nodes within a graph:

$$d_i = \sum_j A_{i,j}$$

$$d = \frac{\sum_i d_i}{N}$$

$$c_i = \frac{\sum_j A_{i,j}}{N - 1}$$

$$c = \frac{\sum_i c_i}{N}$$

where $\mathbf{A}$ is an adjacency matrix, N is the total number of nodes in a graph, c is the cost of a graph (and of each individual node/ROI), and d is the degree of a graph (and of each individual node/ROI)

Degree and Cost at each node/ROI represent measures of network centrality, characterizing the degree of local connectedness of each ROI within a graph. Adjacency matrix thresholding is typically

implemented using a fixed network cost level (e.g. keeping the strongest 10% of connections) in order to allow sensitive between-network comparisons of other graph measures of interest

Average path distance

Path distance between each pair of nodes in a graph is defined as the minimum number of edges traversed in an optimal path between them. Average path distance at each node is defined as the average path-distance between this node and all other nodes in the subgraph of connected nodes:

$$L_i = \frac{\sum_{j \in \Omega_i} D_{i,j}}{N_i - 1}$$

$$L = \frac{\sum_i L_i}{N}$$

where $\mathbf{D}$ is the shortest-path distance matrix, $\mathbf{N}$ is the total number of nodes in a graph, and $\mathbf{L}$ is the averages path distance of a graph (and of each individual node/ROI)

Average path distance represents a measure of node centrality within a network, characterizing the degree of global connectedness of each ROI within a graph. Similarly, network average path distance represents a measure of inter-connectedness or radius of the entire network (e.g. random graphs have comparatively low/compact average path distances, compared to grids)

Clustering Coefficient

Clustering Coefficient is defined as the proportion of connected edges in the local neighboring sub-graph for each node/ROI:

$$CC_i = \frac{\sum_{j,k \in \Gamma_i} A_{j,k}^{(i)}}{d_i(d_i - 1)}$$

$$CC = \frac{\sum_i CC_i}{N}$$

where d is the degree of each node, A is the adjacency matrix within the neighboring sub-graph at each node, characterized by all nodes neighboring this node and all existing edges among them, and CC is the clustering coefficient of a graph (and of each individual node/ROI)

Clustering coefficient represents a measure of local integration, characterizing the degree of inter-connectedness among all nodes within a node neighboring sub-graph. Similarly, network clustering coefficient represents a measure of network locality or coherence (e.g. grid topologies have comparatively high clustering coefficients, compared to random graphs)

Global Efficiency

Global Efficiency at a node is defined as the average of inverse-distances between this node and all other nodes in the same graph:

$$GE_i = \frac{\sum_{j \neq i} 1/D_{i,j}}{N - 1}$$

$$GE = \frac{\sum_i GE_i}{N}$$

where D is the shortest-path distance matrix, N is the number of nodes in a graph, and GE is the Global Efficiency of a graph (and of each individual node/ROI). Similar to Average path distance, Global efficiency at a node represents a measure of this node centrality within the network, characterizing the degree of global connectedness of each ROI. Similarly, network global efficiency represents a measure of inter-connectedness or radius of the entire network (e.g. with higher / more compact global efficiency in random graphs compared to grids)

Local Efficiency

Local Efficiency at each node is defined as the Global efficiency of the neighboring sub-graph of this node:

$$LE_i = \frac{\sum_{j \neq k \in \Gamma_i} 1/D_{j,k}^{(i)}}{d_i(d_i - 1)}$$

$$LE = \frac{\sum_i LE_i}{N}$$

where d is the degree of each node, D is the shortest-path distance matrix within the neighboring sub-graph at each node, characterized by all nodes neighboring this node and all existing edges among them, and LE is the Local Efficiency of a graph (and of each individual node/ROI). Similar to clustering coefficients, Local efficiency represents a measure of local integration or coherence, characterizing the degree of inter-connectedness among all nodes within a node neighboring sub-graph. Similarly, network local efficiency represents a measure of local integration in a network (e.g. with higher local efficiency in grids compared to random graphs)

Betweenness Centrality

Betweenness centrality represents an alternative measure of node centrality within a graph. It is defined as the proportion of times that a node is part of a shortest-path between any two pairs of nodes within a graph:

$$BC_i = \frac{\sum_{j,k \neq i} [i \in P_{j,k}]}{(N-1)(N-2)}$$

$$BC = \frac{\sum_i BC_i}{N}$$

where P is the set of nodes in shortest-path between each pair of nodes, N is the number of nodes in a graph, and BC is the Betweenness Centrality of a graph (and of each individual node/ROI)

3.8. Dynamic connectivity measures

Dynamic connectivity measures are aimed at characterizing and studying sources of temporal variability in functional connectivity patterns. Some of the most common dynamic connectivity techniques are based on sliding-window approaches, where functional connectivity measures of interest are estimated over relatively small time-windows in order to analyze potential changes in these measures over time. In addition, the section below also describes dyn-ICA, a technique currently being actively developed and aimed at characterizing clusters of connections that show similar temporal variations in functional connectivity

Sliding window analyses

Every connectivity measure in CONN, including seed-based, ROI-to-ROI, network and graph measures, can also be estimated from windowed BOLD timeseries using a series of sequential sliding windows. Each individual window is treated as a separate condition, and weighted GLM is used to compute the corresponding condition-/time- specific measures. Variability of these measures across time is then computed as the main measure of interest characterizing dynamic connectivity properties. One example of such sliding-window measures of connectivity is dynamic variability in seed-based or ROI-to-ROI connectivity measures:

- **Dynamic variability in seed-based connectivity (dvSBC):** dvSBC maps represent the degree of temporal variability in functional connectivity between a seed/ROI and every location in the brain. They are defined as the standard deviation in bivariate, multivariate, or semipartial correlation or regression measures between seed ROI and each target voxel, computed using weighted Least Squares (WLS) within a discrete set of temporal sliding windows

e.g. dynamic variability in bivariate regression SBC

$$S(x, t_n + \tau) = \beta_n(x)R(t_n + \tau) + \epsilon_n(x, t_n + \tau)$$

$$\beta_n(x) \mid \min_{\beta_n(x)} \sum_{\tau} w^2(\tau/L) \epsilon_n^2(x, t_n + \tau)$$

$$w(\tau) = \sin^2\left(\frac{\pi}{2}[1 - |\tau|]^+\right)$$

$$DV^2(x) = \frac{1}{N-1} \sum_{n=1}^N (\beta_n(x) - \bar{\beta}(x))^2$$

where **S** is the BOLD timeseries (for simplicity all timeseries are considered centered to zero mean), **R** is the BOLD timeseries within a seed/ROI, **w** is a Hann sliding window of length $2L$, **beta** is the bivariate regression coefficient map within each time window, estimated using weighted least squares (WLS), and **DV** is the **dynamic variability** in SBC connectivity map

- **Dynamic variability in ROI-to-ROI connectivity (dvRRC):** dvRRC matrices represent the degree of temporal variability in functional connectivity between pairs of ROIs. They are defined as the standard deviation in bivariate, multivariate, or semipartial correlation or regression measures between two ROIs, computed using weighted Least Squares (WLS) within a discrete set of temporal sliding windows

e.g. dynamic variability in bivariate regression RRC

$$R_j(t_n + \tau) = \beta_n(i, j)R_i(t_n + \tau) + \epsilon_n(i, j, t_n + \tau)$$

$$\beta_n(i, j) \mid \min_{\beta_n(i, j)} \sum_{\tau} w(\tau/L)^2 \epsilon_n^2(i, j, t + \tau)$$

$$w(\tau) = \sin^2\left(\frac{\pi}{2}[1 - |\tau|]^+\right)$$

$$DV_{i,j}^2 = \frac{1}{N-1} \sum_t (\beta_n(i, j) - \bar{\beta}(i, j))^2$$

where $\mathbf{R}$ is the BOLD timeseries within each seed/ROI, $\mathbf{w}$ is a Hann sliding window of length $2L$, $\mathbf{beta}$ is the bivariate regression coefficient matrix, estimated using weighted least squares (WLS), and $\mathbf{DV}$ is the **dynamic variability** in RRC connectivity matrix

Implementation notes: Sliding window analyses are defined first in the Setup.Conditions tab by selecting 'temporal decomposition (sliding-window)' in the 'time-frequency decomposition' field. This will define a number of new conditions each covering an individual temporal window. Selecting these conditions when running any first-level analysis will compute the individual time-centered connectivity measures as well as the summary dynamic variability measures

Dynamic Independent Component Analyses (dyn-ICA)

Dynamic ICA matrices represent a measure of different modulatory circuits expression and rate of connectivity change between each pair of ROIs, characterized by the strength and sign of connectivity changes covarying with a given component/circuit timeseries. Dyn-ICA matrices are defined as the gPPI interaction terms between each component/circuit timeseries (data-driven gPPI psychological factors) and a series of ROI BOLD timeseries (user-defined gPPI physiological factors).

Group-level dynamic ICA is implemented using iterative dual regression on group-level data obtained by concatenation across-subjects, followed by Independent Component Analyses and gPPI back-projection. Specifically, group-level modulatory components $\text{Gamma}_l(i,j)$ are first estimated following a simplified gPPI model of the form:

$$R_{n,j}(t + \tau) = \left(\tilde{\beta}_n(i,j) + \sum_l \tilde{\gamma}_l(i,j) \tilde{h}_{n,l}(t) \right) R_{n,i}(t + \tau) + \epsilon_{n,i,j}(t, \tau)$$

$$\tilde{\beta}_n(i,j), \tilde{\gamma}_l(i,j), \tilde{h}_{n,l}(t) \mid \min \sum_{t,\tau} w^2(\tau/\sigma) \epsilon_{n,i,j}^2(t, \tau)$$

$$w(z) = \sin^2 \left(\frac{\pi}{2} [1 - |z|]^+ \right)$$

where $\mathbf{R}$ are the BOLD timeseries within each ROI and for each subject n (as before, for simplicity all timeseries are considered centered to zero mean), and $\mathbf{beta}$, the subject-independent stationary connectivity matrix of regression coefficients between each pair of ROIs, $\mathbf{gamma}$, the subject-independent matrix of connectivity changes associated with the l -th modulatory component, and $\mathbf{h}$, the **subject-specific timecourse of each of the estimated modulatory components**, are all estimated using iterative dual regression

The Gamma matrices are then rotated using fastICA with a hyperbolic tangent contrast function, and the ICA mixing matrix $\mathbf{W}$ is inverted to compute the dynamic independent component/circuit timeseries:

$$h_{n,k}(t) = \sum_l \tilde{h}_{n,l}(t) [\mathbf{W}^{-1}]_{l,k}$$

Last, back-projection of the group-level gamma matrices into a series of subject-specific gamma components is performed using standard first-level gPPI models with the estimated dynamic independent component/circuit timecourses $\mathbf{h}$ as gPPI psychological factors:

$$R_{n,j}(t) = \left(\beta_n(i,j) + \sum_k \gamma_{n,k}(i,j) h_{n,k}(t) \right) R_{n,i}(t) + \sum_k \alpha_{n,k}(i,j) h_{n,k}(t) + \epsilon_{n,i,j}(t)$$

$$\beta_n(i,j), \gamma_{n,k}(i,j), \alpha_{n,k}(i,j) \mid \min \sum_t \epsilon_{n,i,j}^2(t)$$

where $\mathbf{R}$ are the BOLD timeseries within each ROI and for each subject n , and $\mathbf{gamma}$ is the **gPPI matrix** of regression coefficients for each modulatory component/circuit, estimated, together with $\mathbf{alpha}$ and $\mathbf{beta}$ above, using least squares (OLS)

Implementation notes: Dynamic ICA analyses are defined in the first-level dyn-ICA tab. These analyses produce multiple outputs, including the individual subject-level matrices $\gamma_{n,k}(i,j)$ (in the second-level dynICA-circuits 'Spatial Properties' tab) and the variability and frequency of the dynamic component/circuit timeseries $\mathbf{h}(t)$ (in the second-level dynICA-circuits 'Temporal Properties' tab)

3.9. Other connectivity measures

This last section describes two additional measures characterizing an aspect of the BOLD timeseries which is often disregarded by most functional connectivity (FC) metrics. All correlation-based metrics are scale invariant, so the actual overall size of the BOLD fluctuations driving many observed FC measures remains unknown. The measures described in this section estimate the variability of the BOLD signal within the standard frequency window of interest (e.g. 0.008Hz to 0.09Hz). While strictly speaking these are not measures of functional connectivity, they are nevertheless useful metrics as they offer often complementary information to many other FC measures.

Amplitude of Low-Frequency Fluctuations (ALFF)

ALFF maps represent a measure of BOLD signal power within the frequency band of interest (e.g. 0.01 - 0.10 Hz). ALFF is defined as the root mean square of BOLD signal at each individual voxel after low- or band- pass filtering (Yang et al. 2007)

$$ALFF(x) = \sqrt{\frac{1}{N} \sum_t (h(t) * S(x, t))^2}$$

where S is original BOLD timeseries before band- or low- pass filtering, h is a low- or band-pass filter, N is the number of timepoints, and **ALFF** is the **Amplitude of Low-Frequency Fluctuations** map

Implementation notes: $S(x,t)$ timeseries are the signals after preprocessing and the initial denoising linear regression step (e.g. session-specific regression of aCompCor, motion, outlier scans, mean, and trend factors) but before the last denoising band-pass filtering step. Filtering is implemented using a Discrete Cosine Transform (DCT) rectangular window for each session-specific BOLD signal timeseries, and the resulting timeseries are then concatenated across multiple runs/sessions; note: an alternative/equivalent formulation in terms of the mean and variance of the denoised and filtered timeseries is the following

$$ALFF(x) = \sqrt{\sigma^2(x)}$$

where μ & σ are the mean and variance, respectively, of the denoised and filtered BOLD signal

$$\mu(x) = \frac{1}{N} \sum_t (h(t) * S(x, t))$$

$$\sigma^2(x) = \frac{1}{N} \sum_t (h(t) * S(x, t) - \mu(x))^2$$

Fractional Amplitude of Low-Frequency Fluctuations (fALFF)

fALFF maps represent a relative measure of BOLD signal power within the frequency band of interest (e.g. 0.01 - 0.10 Hz) compared to that over the entire frequency spectrum. fALFF is defined as the ratio of root mean square of BOLD signal at each individual voxel after vs. before low- or band- pass filtering (Zou et al. 2008) :

$$fALFF(x) = \sqrt{\frac{\sum_t (h(t) * S(x, t))^2}{\sum_t S(x, t)^2}}$$

where S is original BOLD timeseries before band- or low- pass filtering, h is a low- or band-pass filter, and **fALFF** is the **Fractional Amplitude of Low-Frequency Fluctuations** map

4. General Linear Model

Second-level analyses allow researchers to make inferences about properties of groups or populations, by generalizing from the observations of only a subset of subjects in a study. CONN uses the **General Linear Model** (GLM) for all second-level analyses of functional connectivity data. This section describes the mathematics behind GLM, including model definition, parameter estimation, and hypothesis testing framework. It also includes several practical examples and general guidelines aimed at helping researchers use this method to answer their specific research questions.

4.1. Definition and estimation

The General Linear Model defines a multivariate linear association between a set of **explanatory/independent measures $\mathbf{X}$** , and a set of **outcome/dependent measures $\mathbf{Y}$** . In the context of functional connectivity MRI analyses, an outcome variable $\mathbf{y}[n]$ will typically take the form of a row vector encoding functional connectivity values recorded from the n -th subject in a study across one or multiple experimental conditions, and the explanatory variable $\mathbf{x}[n]$ will be a row vector encoding one or several group, behavioral, or demographic variables for that same subject (see *numerical examples* section below).

General Linear Model

$$y_n = x_n B + \varepsilon_n$$

The (generally unknown) effective association in the population between each of the explanatory measures in $\mathbf{X}$ and each of the outcome measures in $\mathbf{Y}$ is characterized by the matrix $\mathbf{B}$ in the GLM equation. The vector epsilon represents the cumulative contribution on the outcome measure $\mathbf{Y}$ of all other unspecified factors beyond those that can be predicted from the knowledge of $\mathbf{X}$. (i.e. the model error term). Based on the Central Limit Theorem, it is often reasonable to model this cumulative contribution as a Normally distributed random term. GLM assumes this term is independent across subjects and follows a multivariate normal distribution with mean zero and an arbitrary variance-covariance structure across outcome measures.

We are typically interested in quantifying or estimating the values of the matrix $\mathbf{B}$, characterizing the net effect of each individual explanatory measure in $\mathbf{X}$ on each individual outcome measure in $\mathbf{Y}$. Because this matrix is constant across subjects, acquiring enough subjects' data enables us to compute a reasonable unbiased estimate of $\mathbf{B}$ using an Ordinary Least Squares (OLS) solution.

Ordinary Least Squares

$$\hat{B} = (X^t X)^{-1} X^t Y$$

For example, from N subjects' data, we would typically construct the **data** and **design matrices**, respectively, as $\mathbf{X} = [x_1' \ x_2' \ x_3' \ \dots \ x_N']'$ and $\mathbf{Y} = [y_1' \ y_2' \ y_3' \ \dots \ y_N']'$, resulting from vertically concatenating the corresponding $\mathbf{x}[n]$ and $\mathbf{y}[n]$ row vectors across all relevant subjects, and then use the OLS equation above to compute the best linear unbiased estimator of the unknown matrix $\mathbf{B}$ from the observed $\mathbf{X}$ and $\mathbf{Y}$ data.

4.2. Hypothesis testing

In addition to estimating an approximation of the matrix **B**, we would often also like to evaluate specific hypotheses about this unknown matrix **B** given our available data. In particular, GLM allows us to use a standard **Likelihood Ratio Test** (LRT) to specify and evaluate any hypothesis of the form " $CBM' = D$ " for any user-defined arbitrary contrast matrices **C**, **M** and **D**.

Hypothesis

$$CBM^t = D ?$$

Choosing different forms of the matrix **C** allows us to construct hypotheses that address specific combinations of explanatory measures **X**, as each column in the vector/matrix **C** is paired with the same column of the design matrix **X**. Similarly, choosing different forms of the matrix **M** allows us to construct hypotheses that address specific combinations of outcome measures **Y**, as each column in the contrast vector/matrix **M** is paired with the same column of the data matrix **Y**. Last, the choice of contrast matrix **D** determines the hypothesized net effect of the selected combination of explanatory measures on the selected combination of outcome measures (e.g. **D** is set to zero in many standard null-hypothesis scenarios).

For any user-defined contrast matrices **C**, **M** and **D**, the associated $CBM' = D$ hypothesis is evaluated using a **Wilks' Lambda** statistic, defined in the context of a Likelihood Ratio Test by comparing a model that is constrained by this hypothesis (i.e. a model where CBM' equals precisely **D**) to an unconstrained model (i.e. a model where CBM' may take any value). In particular, Wilks lambda values range between 0 and 1, and are computed as the ratio of the residual errors of the unconstrained model over those of the constrained model. Low values (close to 0) indicate that the tested hypothesis may be false (i.e. it is appropriate to conclude from our observations that CBM' is likely not equal or close to **D**), while high values (close to 1) typically indicate that there is not enough evidence in our data to reject the tested hypothesis (CBM' might be precisely **D** or,

perhaps more likely, simply close enough so that we still need more data if we hope to find a significant departure from D).

Likelihood Ratio Test

$$\lambda = \frac{|W|}{|W + H|}$$

$$W = M(Y - X\hat{B})^t(Y - X\hat{B})M^t$$

$$H = (C\hat{B}M^t - D)^t(C(X^tX)^{-1}C^t)^{-1}(C\hat{B}M^t - D)$$

In order to more precisely define whether a particular value of lambda (e.g. 0.1) is low-enough to warrant our rejection of the tested hypothesis $CBM^t=D$, the observed lambda value is typically compared to the distribution of lambda values that we could expect if the tested hypothesis was actually true (a **Wilks' Lambda distribution**), choosing to reject our hypothesis if the observed lambda value is below a pre-specified false-positive level (e.g. using a $p < 0.05$ threshold means that we will reject our hypothesis if the observed lambda value falls below the 5% percentile of the Wilks' Lambda distribution).

Wilks Lambda distribution

$$\lambda \sim \Lambda_{a,b,c}$$

$$a = \text{rank}(M)$$

$$b = N - \text{rank}(X)$$

$$c = \text{rank}(XC^t)$$

Wilks' Lambda distributions have three parameters: the number of dimensions a , the error degrees of freedom b , and the hypothesis degrees of freedom c , which are fully determined from the dimensionality and rank of the original data and choice of contrast matrices. Unfortunately, Wilks' Lambda distributions are only tabulated for a limited number of scenarios/dimensions, so CONN GLM implementation uses the following transformations in order to derive appropriate statistics and p-values for any tested hypothesis, depending on the specific values of a , b , and c :

Case 1. Statistics based on **Student's t-distribution**, when $a=1$ and $c=1$ (e.g. both **M** and **C** are vectors, and **D** is a scalar)

Examples: two-sample t-test, linear regression

LRT case 1: $\Lambda_{1,b,1}$ ($a=1, c=1$)

$$\sqrt{\frac{1-\lambda}{\lambda}} = \frac{C\hat{B}M^t - D}{\sqrt{WC(X^tX)^{-1}C^t}} \sim \frac{1}{\sqrt{b}}T_b$$

Case 2. Statistics based on **F-distribution**, when $a>1$ and $c=1$ (e.g. **M** is a matrix, and both **C** and **D** are vectors)

Examples: Hotelling's two sample t-square test, repeated measures ANOVA, multivariate regression

LRT case 2: $\Lambda_{a,b,1}$ ($a>1, c=1$)

$$\frac{1-\lambda}{\lambda} = \frac{(C\hat{B}M^t - D)W^{-1}(C\hat{B}M^t - D)^t}{C(X^tX)^{-1}C^t} \sim \frac{a}{b-a+1}F_{a,b-a+1}$$

Case 3. Statistics based on **F-distribution**, when $a=1$ and $c>1$ (e.g. **C** is a matrix, and both **M** and **D** are vectors)

Examples: ANOVA, ANCOVA, multiple regression omnibus test

LRT case 3: $\Lambda_{1,b,c}$ ($a=1, c>1$)

$$\frac{1-\lambda}{\lambda} = \frac{H}{W} \sim \frac{c}{b}F_{c,b}$$

Case 4. Statistics based on Rao's approximating **F-distribution**, when $a>1$ and $c>1$ (e.g. all **M**, **C**, and **D** are matrices)

Examples: MANOVA, MANCOVA, multivariate regression omnibus test

LRT case 4: $\Lambda_{a,b,c}$ ($a>1, c>1$)

$$\frac{1 - \lambda^{1/e}}{\lambda^{1/e}} \sim \frac{ac}{d} F_{ac,d}$$

$$d = \left(b - \frac{a - c + 1}{2} \right) e - \frac{ac}{2} + 1$$

$$e = \sqrt{\frac{a^2 c^2 - 4}{a^2 + c^2 - 5}}$$

Wilks' Lambda alternatives and generalizations. In some cases, when the number of conditions is large and the sample size is too small ($a>b$), it is not possible to use statistics based on Wilks' Lambda, since the unknown error covariance **W** cannot be properly estimated. This is common, for example, in the context of omnibus tests, where we wish to evaluate whether some effect is present over a potentially large number of individual cases (e.g. simultaneously evaluating multiple SBC maps, each based on a different seed area). In those scenarios, one alternative available in CONN is to use statistics based on a conservative Satterthwaite F-distribution approximation:

LRT alternative case 2 or 4: $\Lambda_{a,b,c}$ ($a>b$)

$$\frac{tr(H)}{tr(W)} \sim \frac{c}{b} F_{kc, kb}$$

$$k = \frac{tr(W)^2}{tr(W^2)}$$

Another common alternative in these scenarios is to apply first a linear dimensionality reduction step by projecting the original data **Y**

over the subspace spanned by the first few singular vectors from a model-agnostic Singular Value Decomposition (Strang 2007). As long as the dimensionality of this subspace is chosen to be smaller than the error degrees of freedom b , the full GLM error covariance over this subspace can be properly estimated, allowing the evaluation of Likelihood Ratio Tests using the standard Wilks' Lambda statistics (LRT case 2 or case 4).

General observations. It could be argued that two-sided hypotheses of the form " $\text{CBM}'=0$ " are almost surely false in real world data, where an effect may be arbitrarily small but almost never precisely zero. Because of this, failure to reject a hypothesis of this form often simply indicates that the effect being evaluated (e.g. difference in connectivity between two groups) is too small to be detectable with the current experimental setup (e.g. with the current acquisition parameters and number of subjects) rather than truly non-existent. In this context, it is generally recommended, and not a bad idea at all, to attempt to always quantify and report the effects measured (e.g. report the estimated $\mathbf{B}$ values) instead of only relying on and reporting the significance of the hypotheses being evaluated. This can help build increasingly better model-based estimates of these effects, going beyond the initial but limited question of whether they "exist" (i.e. are they non-zero) or not. Last, testing one-tailed hypotheses of the form " $\text{CBM}'>D$ " (these are available in CONN for univariate effects using LRT case 1) can also be used as a tool to combine "practical" significance (is an effect large?) with "statistical" significance (how confident are we of this, given the available data?)

4.3. GLM model and contrast specification



Figure 19. Schematic illustration of the four elements necessary to fully specify a GLM analysis and contrasts of interest: **subject-effects**, defining the columns of the design matrix **X**; **between-subject contrasts**, defining the contrast vector/matrix **C**; **conditions**, defining the columns of the data matrix **Y**; and **between-conditions contrast**, defining the contrast vector/matrix **M** (unless otherwise specified the matrix **D** is assumed equal to zero)

Using the same General Linear Model framework, it is possible to specify a very large array of classical analyses, including bivariate, multiple, and multivariate regression models, one-sample, two-sample, and paired t-tests, mixed within- and between- subject n-way ANOVAs, MANOVAs, etc. To define an individual analysis, it is only necessary to specify four items, associated with **X**, **Y**, **C**, and **M** matrices, respectively, in the GLM framework:

1. **Subject-effects**: what is the list of explanatory/independent measures that we would like to include in this analysis? (i.e. what are the columns of **X**?)

This is typically defined simply by listing a series of subject-level covariates (e.g. age, IQ). In CONN these variables are defined in *Setup.Covariates (2nd-level)*. In addition to continuous variables, dummy-coded group variables are often used to identify groups of

subjects in our studies (e.g. a *Patients* covariate may take a value of 1 for patients, and 0 for controls). One such dichotomous variable identifying the entire group of subjects in our study is also often used in simple designs where constant effects across all subjects are needed (in CONN this variable is automatically created and named *AllSubjects*, containing the value 1 for every subject)

2. **Between-subjects contrasts:** among these explanatory / independent measures, which one(s) or which combination of them do we want to evaluate/test? (i.e. what is the **C** vector/matrix?)

In the simplest scenario this contrast is just a vector, with as many elements as explanatory/independent measures, and having a value of 1 for the individual effect that we would like to evaluate/test and 0's for all other elements (e.g. if we have entered two subject-effects, characterizing *Patient* and *Control* subjects, respectively, a contrast vector with values [1, 0] would specify that we would like to evaluate/test the effect in Patients only. Other simple scenarios involve a contrast that acts to compare two effects, which is defined simply by entering a 1 and a -1 in the two elements that we would like to compare, and 0's in all other elements -if there are any- (e.g. a contrast vector with values [-1, 1] in the previous example would compare the effect in Controls to that in Patients). More complex contrast vectors can be specified simply as the weights of any desired linear combination of our model subject-effects (e.g. a contrast vector with values [0.5, 0.5] in the previous example would estimate the average effect across both Patients and Control subjects). Last, contrast matrices can be used to evaluate/test multiple effects jointly, where each individual effect is defined in the regular manner as a contrast vector, and those vectors are simply concatenated into a matrix (e.g. in a model with three groups instead of two, a contrast matrix [1, -1, 0; 0, 1, -1; 1, 0, -1] evaluates/tests the presence of any differences between the three groups)

3. **Conditions** (also known as outcomes or measures): what is the list of outcome/dependent measures that we would like to include in this analysis? (i.e. what are the columns of **Y**?)

This is typically defined by listing the individual outcome variables that we would like to investigate (e.g. SBC maps during a single rest

condition in a standard resting state analysis, or SBC maps during *Pre* and *Post* conditions in an intervention design). In CONN this is defined by a combination of choosing which particular first-level functional connectivity measures, and which experimental conditions (if applicable), we would like to evaluate (e.g. SBC maps with one or several seeds during rest)

4. **Between-conditions contrast** (also known as between-measures or within-subjects contrast): among these outcome/dependent measures, which one(s) or which combination of them do we want to evaluate/test? (i.e. what is the M vector/matrix?)

These contrasts are defined in the same way as the *between-subject* contrasts above, but now spanning across conditions instead of across subject-effects (e.g. if we have selected two conditions, characterizing SBC maps pre- and post- intervention, a contrast vector with values [-1, 1] would compare the connectivity values across these two conditions)

Behind the apparent simplicity of these choices, a perhaps surprising array of different analyses can be specified using this framework. Some examples are shown in the table below

Analysis type	Subject-effects [Between-subjects contrast]	Conditions [Between- conditions contrast]
<i>one-sample t-test</i>	AllSubjects [1]	Rest [1]
<i>two-sample t-test</i>	Patients, Controls [1 -1]	Rest [1]
<i>paired t-test</i>	AllSubjects [1]	Pre, Post [-1 1]
<i>regression</i>	AllSubjects, behavioral [0 1]	Rest [1]
<i>multiple regression (unique effect evaluated at fixed level of other effects)</i>	AllSubjects, behav, age [0 1 0]	Rest [1]

<i>multiple regression (joint effects)</i>	AllSubjects, behav1, behav2 [0 1 0; 0 0 1]	Rest [1]
<i>2x2 between-subjects ANOVA interaction</i>	PatientsTreat, PatientsSham, ControlTreat, ControlSham [1 -1 -1 1]	Rest [1]
<i>2x2 within-subjects / repeated measures ANOVA interaction</i>	AllSubjects [1]	PreTask, PreRest, PostTask, PostRest [-1 1 1 -1]
<i>2x2 mixed ANOVA interaction</i>	Patients, Controls [1 -1]	Pre, Post [-1 1]
<i>3x2 mixed ANOVA interaction</i>	PatientsA, PatientsB, PatientsC [1 -1 0; 0 1 -1]	Pre, Post [-1 1]
<i>2x2x2 mixed ANOVA interaction</i>	PatientsTreat, PatientsSham, ControlTreat, ControlSham [1 -1 -1 1]	Pre, Post [-1 1]
<i>one-way ANCOVA covariate control</i>	Patients, Controls, age [1 -1 0]	Rest [1]
<i>one-way ANCOVA covariate interaction (comparing regression between groups)</i>	Patients, Controls, behavPatients, behavControls [0 0 1 -1]	Rest [1]
<i>General Linear Model data fit: $Y = X*B$ hyp test: $C*B*M' = 0$</i>	Columns of X (effects) [C]	Columns of Y (conditions) [M]

Table 1. Example of equivalent model and contrast specification in GLM for an array of classical analyses

4.4. Numerical examples

While generally, in the context of fMRI analyses, we are interested in simultaneously evaluating or testing thousands of individual measures (e.g. SBC maps containing one measure of interest at each voxel), in this section we will consider, for simplicity and illustration purposes, just a single measure, and proceed to manually define a General Linear Model and use it to test some simple properties of this measure across subjects.

For these examples, imagine we have 10 subjects, and for each subject we have computed two functional connectivity measures of interest (e.g. connectivity strength between two a priori ROIs, estimated pre- and post- treatment). The data matrix would look something like the example to the right (**Y** matrix), where the first column in **Y** represents the connectivity values pre-treatment and the second column the values post-treatment for each of the 10 subjects.

Imagine also that these 10 subjects were divided in two groups (e.g. patients from two different clinics undergoing different types of treatments). In order to encode this information, we would create a design matrix like the one in the example to the right (**X** matrix), where the first column indicates those subjects from clinic #1, and the second column those subjects from clinic #2.

Data (**Y**) and Design (**X**) matrices

$$Y = \begin{bmatrix} 0.38 & 0.74 \\ 0.39 & 0.67 \\ 0.47 & 0.56 \\ 0.31 & 0.53 \\ 0.41 & 0.62 \\ 0.28 & 0.36 \\ 0.29 & 0.35 \\ 0.26 & 0.41 \\ 0.09 & 0.53 \\ 0.29 & 0.15 \end{bmatrix} \quad X = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Example 1: Imagine now we would like to quantify and evaluate potential differences in functional connectivity values between the patients from the two clinics, either before or after intervention. To do this we want to define the matrices C, M, and D as shown to the right. In particular the contrast C is defined as [-1 1] in order to compare the effect of the two explanatory measures (the two clinics), and the contrast M is defined as the identity matrix in order to evaluate the effect on any of the outcome measures (either pre- or post- conditions). Rejecting this hypothesis would allow us to conclude that mean connectivity values in the two clinics are unlikely to be the same at either pre- or post- intervention.

Hypothesis

$$[B_{11} \ B_{12}] = [B_{21} \ B_{22}] \iff CBM^t = D$$

$$C = [-1 \ 1] \ M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \ D = [0 \ 0]$$

In order to evaluate this hypothesis, we could manually compute the lambda value, and compare that to the Wilks' Lambda distribution with 2, 8, and 1 degrees of freedom, or we could, for example, use the syntax:

`[h, f, p, dof] = conn_glm( X, Y, C, M, D )`

and CONN will use case-2 transformations to evaluate this hypothesis, returning the F- statistics and associated p-values shown here:

Results

$$C\hat{B}M^t = [-0.15 \ -0.26]$$

$$F(2, 6) = 21.50$$

$$p = 0.0010$$

These results indicate that the functional connectivity trajectories before and after intervention in the two groups are significantly different (with higher connectivity in clinic #1 compared to clinic #2; 0.15 higher pre-intervention and 0.26 higher post-intervention). Note that GLM analyses in this context are exactly equivalent to those from a mixed-model two-way ANCOVA evaluating potential main effects of *clinic* (a between-subjects factor).

Example 2: Let's say now that we would like to quantify and evaluate potential differences in functional connectivity values between the two timepoints (pre- vs. post- intervention) in any of the two groups/clinics. To do this, we could define the matrices C , M , and D as shown to the right. In particular the matrix C is defined as the identity matrix in order to evaluate the effect of any of the two explanatory measures (the two clinics), while the matrix M is defined as $[1 \ -1]$ in order to compare the two different outcome measures (pre- and post- conditions). This has the net effect of evaluating whether the connectivity values are different pre- vs. post-intervention in either of the two groups.

Hypothesis

$$\begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} = \begin{bmatrix} B_{12} \\ B_{22} \end{bmatrix} \iff CBM^t = D$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad M = [-1 \quad 1] \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

As before, we could evaluate this hypothesis by manually computing the lambda value, and comparing that to the Wilks' Lambda distribution with now 1, 8, and 2 degrees of freedom. Equivalently, if we use instead a `conn_glm` call, CONN will use case-3 transformations to evaluate this hypothesis, returning the F- statistics and associated p-values shown here.

Results

$$C\hat{B}M^t = \begin{bmatrix} 0.23 \\ 0.12 \end{bmatrix}$$

$$F(2, 8) = 6.29$$

$$p = 0.0229$$

These results indicate that there are significant functional connectivity changes post- vs. pre- intervention in our subjects (with general increases in connectivity post- intervention; a 0.23 increase in clinic #1 and a 0.12 increase in clinic #2). Note that GLM analyses in this context are exactly equivalent to those from a mixed-model two-way ANCOVA evaluating potential main effects of *treatment* (a within-subjects factor).

Example 3: Last, let's evaluate whether these increases in connectivity with intervention (a 0.23 increase in clinic #1 vs. a 0.12 increase in clinic #2) are significantly different between the two clinics; or equivalently whether the difference in connectivity between the two clinics (a 0.15 difference pre-intervention vs. a 0.26 difference post-intervention) are significantly different between the two timepoints. To do this we would want now to define the matrices C, M, and D as shown to the right. In this case the matrix C is set to [-1 1] as in Example 1 in order to compare the contribution of the two predictor measures (the two clinics), and the matrix M is set to [-1 1] as in Example 2 in order to compare the effect across the two outcome measures (pre- and post- conditions). This has the net effect of comparing the between-group differences in connectivity between the two time-points.

Hypothesis

$$B_{22} - B_{12} = B_{21} - B_{11} \iff CBM^t = D$$

$$C = [-1 \quad 1] \quad M = [-1 \quad 1] \quad D = [0]$$

As before, we could evaluate this hypothesis by manually computing the lambda value, and comparing that to the Wilks' Lambda distribution with now 1, 8, and 1 degrees of freedom, or, use a `conn_glm` call. CONN will use case-1 transformations to evaluate this hypothesis, returning the T- statistics and associated p-values shown here.

Results

$$C\hat{B}M^t = [-0.11]$$

$$t(8) = -1.10$$

$$p = 0.3041$$

These results indicate that, if there is a clinic by treatment interaction, the effect is relatively small and cannot be detected with this study sample size. Note that, as before, GLM analyses in this context are exactly equivalent to those from a mixed-model two-way ANCOVA evaluating potential interactions between *clinic* (a between-subjects factor) and *treatment* (a within-subjects factor).

4.5. How to run CONN General Linear Model analyses

CONN's second-level analyses can be run using any of the following options:

Option 1: using CONN's gui

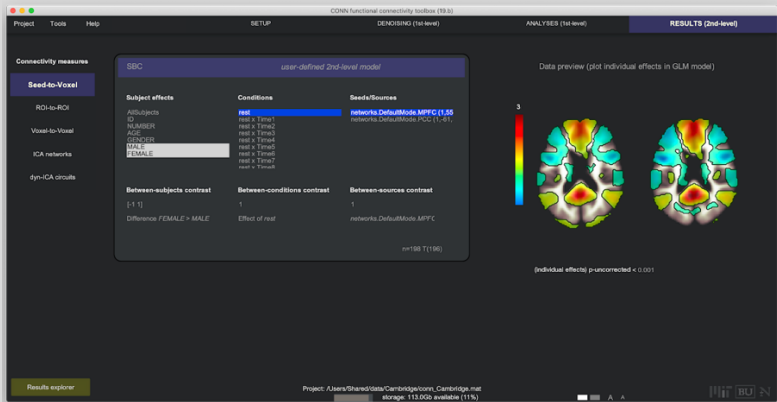


Figure 20. Example of CONN's second-level analysis gui

If you have analyzed your data in CONN, in the *Results* window you may define a new second-level analyses by:

- a) selecting in the **Subject effects** list the desired set of independent measures X in your model (e.g. group variables MALES and FEMALES in the example above)
- b) entering or selecting an associated **between-subjects contrast** C across the selected measures (e.g. [1 -1] in the example above to compare the two groups)
- c) selecting in the **Conditions** list the desired set of dependent measures Y in your model (e.g. REST condition in the example above)
- and d) entering or selecting an associated **between-conditions contrast** M across the selected measures (e.g. 1

in the example above to look only at connectivity measures during the REST condition)

After this simply click on '**Results explorer**' to have CONN compute these analyses and display the results (see [cluster-level inferences](#) section for additional details). Optionally select '*user-defined 2nd-level model*' and rename this model for easy access to this analysis from CONN's gui in the future.

Option 2: using CONN's batch commands

Similarly, if you have analyzed your data in CONN, you may also run the same second-level analysis shown in the example above using MATLAB command syntax:

```
conn_batch( 'filename', '/data/Cambridge/conn_Cambridge.mat', ...
'Results.analysis_number', 'SBC', ...
'Results.between_subjects.effect_names', {'MALE', 'FEMALE'}, ...
'Results.between_subjects.contrast', [-1 1], ...
'Results.between_conditions.effect_names', {'REST'}, ...
'Results.between_conditions.contrast', [1], ...
'Results.between_sources.effect_names', {'networks.DefaultMode.MPFC'}, ...
'Results.between_sources.contrast', [1], ...
'Results.display', true )
```

optionally adding to this command any desired alternative field name/value pairs (see [doc conn_batch](#) for additional details)

Option 3: using CONN's modular functions

If you have not analyzed your data in CONN but still would like to run CONN's second-level GLM analyses on data from other sources (e.g. to analyze a set of nifti contrast images computed using any arbitrary software package), you may do so using the following MATLAB command syntax:

```
conn module glm
```

to manually specify a new second-level GLM analysis. Optionally add to this command any desired field name/value pairs (see [doc conn module](#) for additional options). For example:

```
conn_module( 'glm' , ...  
            'data', Y, ...  
            'design_matrix', X, ...  
            'contrast_between', C, ...  
            'contrast_within', M, ...  
            'folder', outputfoldername )
```

where **X** is the GLM design matrix (matrix with one row per subject and one column per modeled effect) , **Y** is the input NIFTI image files (a cell matrix containing a list of filenames, with one row per subject and one column per measure/condition), **C** is the between-subjects contrast (one vector/matrix with the same number of elements/columns as X), and **M** is the within-subjects contrast (one vector/matrix with the same number of elements/columns as Y).

note: when using CONN's modular functions, input NIFTI images can be standard 3d volumes (e.g. .nii files) for voxel-based analyses, fsaverage volumes (e.g. .surf.nii files) for surface-based analyses, or 2d matrices (e.g. .mtx.nii files) for ROI-to-ROI analyses. See [doc conn surf write](#), [doc conn surf curv2nii](#), or [doc conn surf gii2nii](#) for help creating fsaverage nifti files from your data, or [doc conn mtx write](#) for help creating matrix nifti files from your data.

5. Cluster-level inferences

Whether in the context of voxel-based measures, such as when studying properties of SBC maps across multiple subjects, or in the context of ROI-to-ROI measures, such as when studying properties of RRC matrices across the same subjects, second-level analyses will perform a separate statistical test for each individual analysis unit (voxels in the former case or ROI-to-ROI connections in the latter). This often poses a considerable *multiple-comparisons* problem, where traditional false positive control strategies at the level of these individual units (e.g. thresholding individual voxels at a $p < 0.05$ level) would result in unacceptably high rates of false positives across the entire analysis (e.g. across the entire brain, in voxel-based analyses). The following sections describe a number of strategies that have been developed to address this issue while simultaneously improving researchers' ability to make meaningful inferences from their second-level analysis results.

5.1. Cluster-level inferences in voxel-based analyses

A standard second-level General Linear Model analysis of fMRI functional activation or fcMRI connectivity maps produces a single statistical parametric map, with one T- or F- value for each voxel in this map characterizing the effect of interest (e.g. difference in connectivity between two groups) at each location. When reporting or interpreting these results, rather than focusing on individual voxels, it is often convenient to focus on areas sharing similar effects or results. In order to support our ability to make inferences about these areas, a number of methods have been developed that precisely specify how these areas/clusters are defined from the data, and how to assign statistics to each area/cluster in a way that allows us to make inferences about them while controlling the analysis-wise chance of false positives

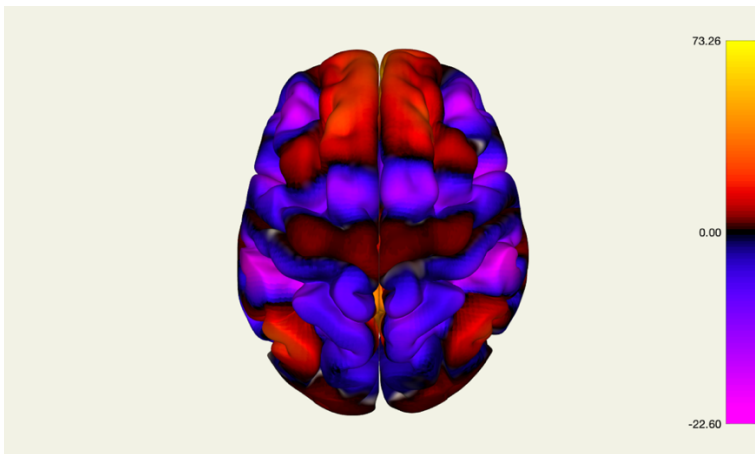


Figure 21. Example statistical parametric map showing average SBC with DefaultMode.MPFC seed during rest (color coded one-sample T-test statistics)

CONN implements three popular methods that offer family-wise error control at the level of individual clusters: 1) parametric statistics based on **Random Field Theory** (Worsley et al. 1996); 2) nonparametric statistics based on **permutation/randomization**

analyses (Bullmore et al. 1999); and 3) nonparametric statistics based on **Threshold Free Cluster Enhancement** (Smith and Nichols, 2007)

Random Field Theory (RFT) parametric statistics

Cluster-level inferences based on Gaussian Random Field theory (Worsley et al. 1996) start with a statistical parametric map of T- or F-values estimated using a General Linear Model. This map is first thresholded using an a priori "height" threshold level (e.g. $T > 3$ or $p < 0.001$). The resulting suprathreshold areas define a series of non-overlapping clusters (neighboring voxels using an 18-connectivity criterion when analyzing 3D volumes). Each cluster is then characterized by its extent/size (number of voxels), and these sizes are compared to a known distribution of expected cluster sizes under the null hypothesis, as estimated from a combination of the analysis degrees of freedom, the approximated level of spatial autocorrelation of the general linear model residuals, and the selected height threshold level. The results are summarized, for each individual cluster, by a ***cluster-level uncorrected p-value***, defined as the likelihood of a randomly-selected cluster having this size or larger under the null hypothesis, as well as a ***cluster-level FWE-corrected p-value***, defined as the likelihood under the null hypothesis of observing at least one or more clusters of this or larger size over the entire analysis volume, and a ***cluster-level FDR-corrected p-value*** (topological False Discovery Rate, Chumbley et al. 2010), defined as the expected proportion of false discoveries among all clusters of this or larger size over the entire analysis volume, again under the null hypothesis.

A standard criterion ("***standard settings for cluster-based inferences #1: Random Field Theory parametric statistics***" in CONN's results explorer gui) for thresholding voxel-based functional activation or connectivity spatial parametric maps while appropriately controlling the family-wise error rate, uses RFT with a combination of an uncorrected $p < 0.001$ height threshold to initially define clusters of interest from the original statistical parametric maps, and a FDR-

corrected $p < 0.05$ cluster-level threshold to select among the resulting clusters those deemed significant (those larger than what we could reasonably expect under the null hypothesis)

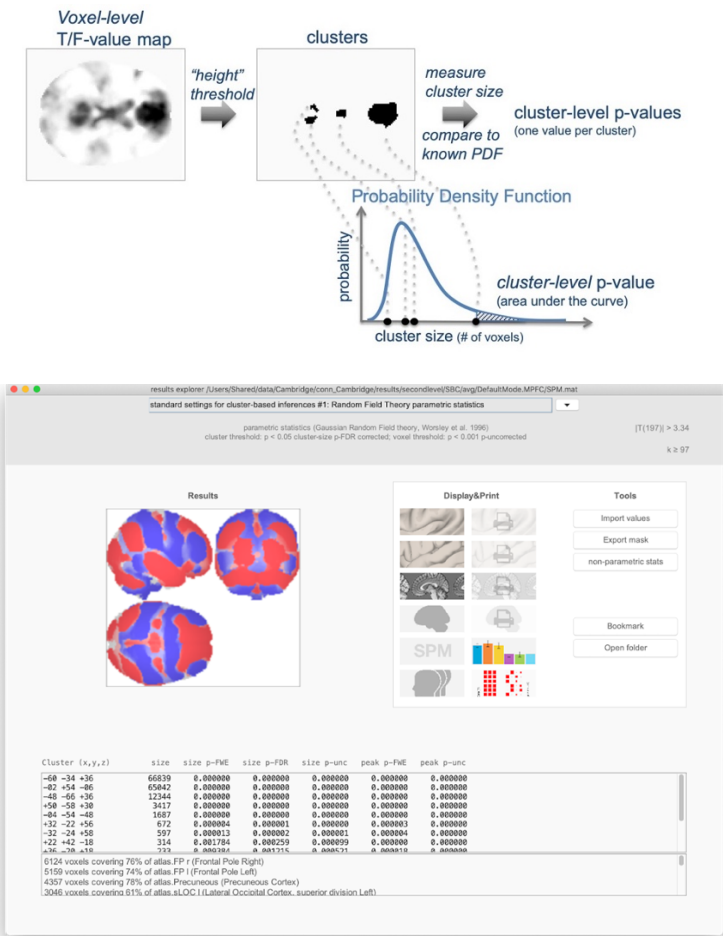


Figure 22. **Top:** schematic procedure to derive RFT cluster-level statistics (see text for details); **bottom:** example report of RFT cluster-level inferences

Randomization/permutation nonparametric statistics

Cluster-level inferences based on randomization/permutation analyses (Bullmore et al. 1999) in CONN use the same cluster-forming procedure as above, with two main differences. First, instead of relying on Random Field Theory assumptions to approximate the probability density function of each cluster size under the null hypothesis, these and related distributions are numerically estimated using multiple (1,000 or higher) randomization/permutation iterations of the original data designed to explicitly simulate the null hypothesis. For each of these iterations, the statistical parametric map of T- or F- values is computed and thresholded in the same way as in the original data, and the properties of the resulting clusters are combined to numerically estimate the desired probability density functions under the null hypothesis for our choice of cluster metrics. And second, instead of relying simply on a cluster size to evaluate each cluster significance level, nonparametric analyses rely on each cluster mass (the sum of the T-squared or F- statistics across all voxels within each cluster), defined as:

$$m_i = \int_{x \in \Omega_i} h^H(x) dx$$

where m is the mass of the i -th cluster, $h(x)$ is the original statistical parametric map, and $H=2$ for T-statistic input maps, or $H=1$ for F-statistic input maps. Compared to cluster size, this measure can be expected to afford higher sensitivity not only to effects distributed over large areas but also to strong effects that may be concentrated over relatively small areas.

As before, the results are summarized, for each individual cluster, by uncorrected, FWE-corrected, and FDR-corrected cluster-level p -values. **Uncorrected cluster-level p -values** are computed by comparing the mass of a given cluster with the observed distribution of cluster mass values across all clusters observed in the permutation/randomization iterations, **FDR-corrected cluster-level p -**

values are computed using the standard Benjamini and Hochberg's FDR algorithm based on the estimated uncorrected p-values, and **FWE-corrected cluster-level p-values** are computed by comparing the mass of a given cluster with the distribution of the maximum/largest cluster mass across the entire analysis volume observed in each permutation/randomization iteration

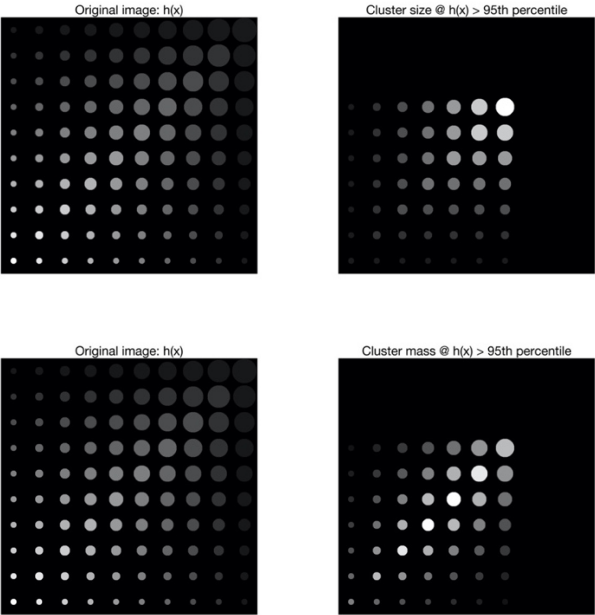


Figure 23. Illustrative example of cluster-size and cluster-mass measures. Original image (left) is first thresholded by intensity. The resulting clusters are intensity-coded by cluster-size (top-right), or by cluster-mass (bottom-right)

A second standard criterion ("**standard settings for cluster-based inferences #2: permutation/randomization analyses**" in CONN's results explorer gui) for thresholding voxel-based spatial parametric maps that also appropriately controls family-wise error rates, uses randomization/permutation analyses with a combination of a uncorrected $p < 0.01$ height threshold in order to initially define clusters of interest, and a FDR-corrected $p < 0.05$ cluster-level

threshold to select among the resulting clusters those deemed significant (clusters with larger mass than what we could reasonably expect under the null hypothesis)

Note that one of the main advantages of this approach over alternatives such as RFT, is that it remains valid over any user-defined choice of height threshold values (e.g. $p < 0.01$ as used here), while Random Field Theory assumptions expect the height threshold to be relatively conservative (e.g. $p < 0.001$ or smaller, see Eklund 2016). That makes permutation/randomization analyses well suited to deal with small samples and/or low-powered studies where expected effects may be too weak to reasonably surpass conservative voxel-level height thresholds

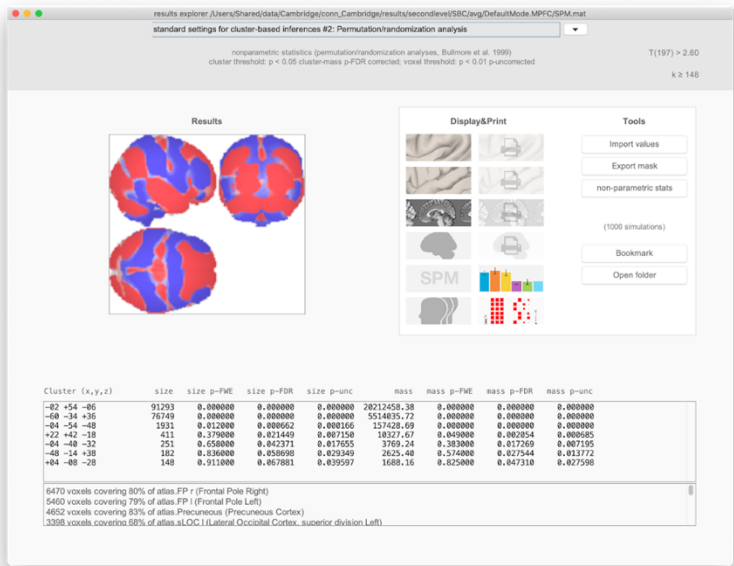


Figure 24. Example report of randomization/permutation nonparametric cluster-level inferences

Threshold Free Cluster Enhancement (TFCE) statistics

Cluster-level inferences based on Threshold Free Cluster Enhancement analyses (Smith and Nichols 2007) aim at removing the dependency of other cluster-level inference methodologies on the choice of an a priori cluster-forming height threshold. TFCE analyses in CONN start with a statistical parametric map of T- or F- values estimated using a General Linear Model. Instead of thresholding this map, a derived TFCE score map is instead computed as:

$$TFCE(x) = \int_{h_{min}}^{h(x)} h^H e_x^E(h) dh$$

where $h(x)$ is the original statistical parametric map, and $e(h)$ is the extent of a cluster thresholded at height level h and containing the point x . In CONN implementation, TFCE model parameters are set by default to $h_{min}=1$, $E=0.5$, and $H=2$ for T-statistic input maps, or $H=1$ for F-statistic input maps, and the TFCE scores are computed using an exact integration method ($dh \rightarrow 0$ limit, unlike other implementations that use $dh=0.1$ discrete approximations)

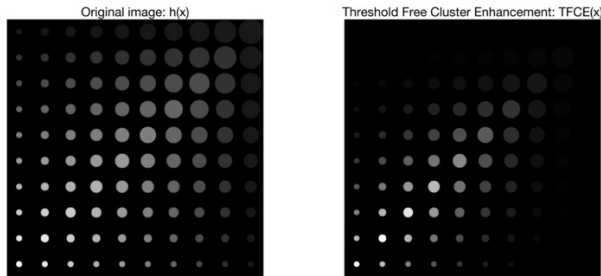


Figure 25. Illustrative example of TFCE operation. Original image (left) and associated TFCE-score image (right)

The resulting TFCE scores at each voxel combine the strength of the statistical effect at this location with the extent of all clusters that

would appear at this location when thresholding the original statistical parametric maps at any arbitrary height threshold. Then, as before, the expected distribution of TFCE values under the null hypothesis is numerically estimated using multiple (1,000 or higher) randomization/permutation iterations of the original data. Comparing at each voxel the observed TFCE value with the null-hypothesis distribution of maximum TFCE values across the entire analysis volume is used to compute **voxel-level FWE-corrected p-values** (defined as the likelihood under the null hypothesis of observing at least one or more voxels with this or larger TFCE scores over the entire analysis volume). Similarly, and following the approach in Chumbley et al. 2010, comparing each local-extremum/peak in the TFCE map with the null hypothesis distribution of local-peak TFCE values can be used to compute **peak-level uncorrected p-values** , defined as the likelihood under the null hypothesis of one randomly-selected peak in the TFCE map having this or larger scores, and associated **peak-level FDR-corrected p-values**, defined as the expected proportion of false discoveries across the entire analysis volume among peaks having this or larger TFCE scores

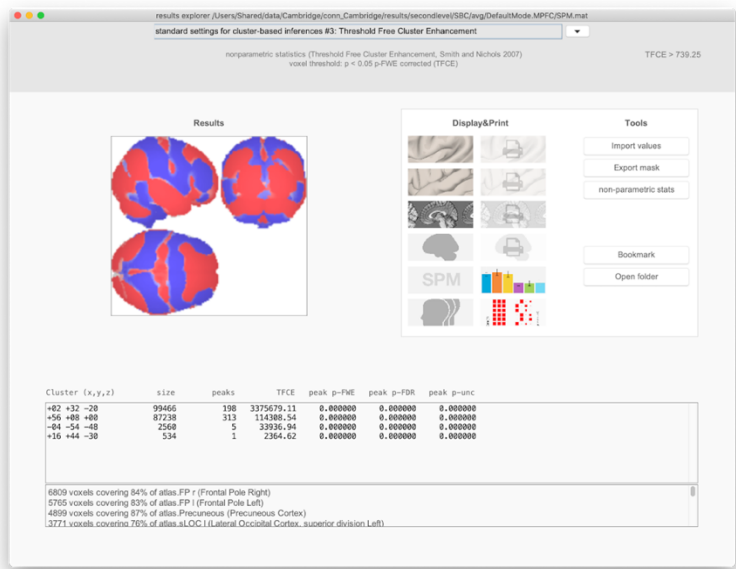


Figure 26. Example report of TFCE nonparametric cluster-level inferences

Yet another standard criterion ("***standard settings for cluster-based inferences #3: Threshold Free Cluster Enhancement***" in CONN's results explorer gui) for thresholding voxel-based spatial parametric maps while appropriately controlling the family-wise error rate, uses TFCE analyses with a FWE-corrected $p < 0.05$ voxel-level threshold to select among the resulting maps those areas deemed significant (including only voxels having TFCE scores larger than what we could reasonably expect under the null hypothesis).

5.2. Cluster-level inferences in ROI-to-ROI analyses

A standard second-level General Linear Model analysis of fMRI connectivity matrices produces a single statistical matrix of T- or F-values, characterizing the effect of interest (e.g. difference in connectivity between two groups) among all possible pairs of ROIs. Similarly to the voxel-based analyses case, when the number of ROIs is large (e.g. from an atlas defining hundreds of regions across the entire brain, which translates to tens of thousands of connections), rather than focusing on individual connections between all possible pairs of ROIs, it is often convenient to focus on groups of nearby or related connections sharing similar effects or results. In order to support our ability to make inferences about these groups of connections, multiple methods have been developed that precisely specify how these groups/clusters can be defined from the data, and how to assign statistics to each of these groups/clusters in a way that allows us to make inferences about them while controlling the analysis-wise chance of false positives.

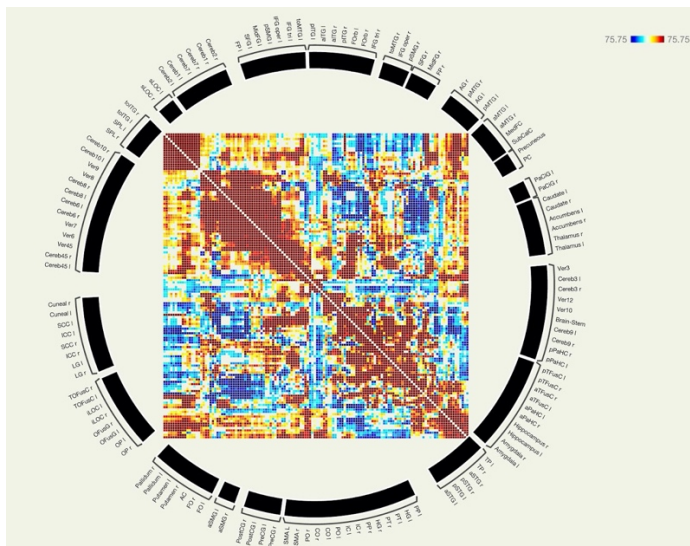


Figure 27. Example 132-ROIs RRC matrix during rest, average across 198 subjects (color coded one-sample *T*-test statistics)

CONN implements three popular methods offering family-wise error control at the level of individual clusters: 1) parametric statistics based on **Functional Network Connectivity**; 2) nonparametric statistics based on **permutation/randomization analyses**; and 3) nonparametric statistics based on **Threshold Free Cluster Enhancement**

Functional Network Connectivity (FNC) multivariate parametric statistics

Cluster-level inferences based on multivariate statistics start by considering groups/networks of related ROIs. These networks can be manually defined by researchers (e.g. from an atlas), or they can be defined using a data-driven hierarchical clustering procedure (complete-linkage clustering, Sorensen 1948) based on ROI-to-ROI anatomical proximity or functional similarity metrics. Once networks of ROIs are defined, FNC analyzes the entire set of connections between all pairs of ROIs in terms of the *within-* and *between-* network connectivity sets (Functional Network Connectivity, Jafri et al. 2008), performing a multivariate parametric General Linear Model analysis for all connections included in each of these sets/clusters of connections. This results in a F- statistic for each pair of networks and an associated ***uncorrected cluster-level p-value***, defined as the likelihood under the null hypothesis of a randomly selected pair of networks showing equal or larger effects than those observed between this pair of networks, and a ***FDR-corrected cluster-level p-value*** (Benjamini and Hochberg, 1995), defined as the expected proportion of false discoveries among all pairs of network with similar or larger effects across the entire set of FNC pairs.

A standard criterion ("***standard settings for cluster-based inferences #1: parametric multivariate statistics***" in CONN's ROI-to-ROI results explorer gui) for thresholding ROI-to-ROI parametric maps while appropriately controlling the family-wise error rate, uses FNC with a FDR-corrected $p < 0.05$ cluster-level threshold to select among all network-to-network connectivity sets those deemed significant

(showing larger multivariate effects than what we could reasonably expect under the null hypothesis), together with a post-hoc uncorrected $p < 0.05$ height (connection-level) threshold to help characterize the pattern of individual connections that show some of the largest effects within each significant set.

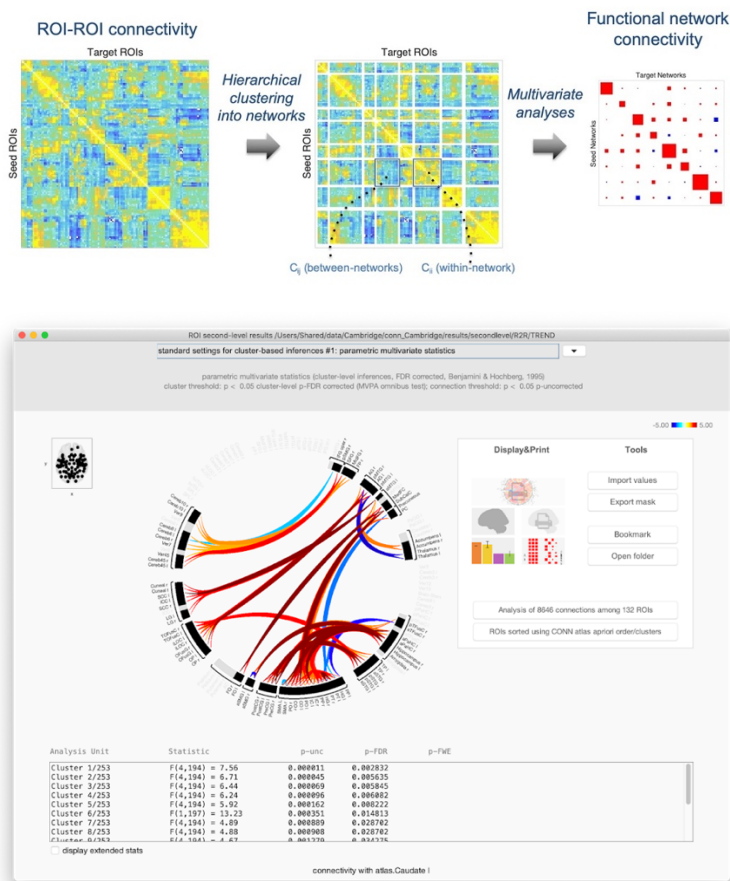


Figure 28. *Top*: schematic procedure to derive FNC multivariate statistics (see text for details); *bottom*: example report of FNC cluster-level inferences

Randomization/permutation Spatial Pairwise Clustering (SPC) statistics

Cluster-level inferences based on randomization/permutation ROI-to-ROI analyses in CONN use the general approach known as Spatial Pairwise Clustering (Zalesky et al. 2012). It starts with the entire ROI-to-ROI matrix of T- or F- statistics estimated using a General Linear Model, forming a two-dimensional statistical parametric map. ROIs in this matrix are sorted either manually by the user (e.g. from an atlas), or automatically using a hierarchical clustering procedure (optimal leaf ordering for hierarchical clustering, Bar-Joseph et al. 2001) based on ROI-to-ROI anatomical proximity or functional similarity metrics. Then this statistical parametric map is thresholded using an a priori "height" threshold (e.g. $T > 3$ or $p < 0.001$). The resulting suprathreshold areas define a series of non-overlapping clusters (groups of neighboring connections using an 8-connectivity criterion on upper triangular part of the symmetrized suprathreshold matrix). Each cluster is then characterized by its mass (sum of F- or T-squared statistics over all connections within each cluster), and these values are compared to a distribution of expected cluster mass values under the null hypothesis, which is numerically estimated using multiple (1,000 or higher) randomization/permutation iterations of the original data. For each of these iterations, the new statistical parametric map of T- or F- values is computed and thresholded in the same way as in the original data, and the properties of the resulting suprathreshold clusters are combined to numerically estimate the probability density under the null hypothesis for our choice of cluster metric. The results are summarized, for each individual cluster or group of connections, by ***uncorrected cluster-level p-values***, representing the likelihood of a randomly-selected cluster of connections having this or larger mass under the null hypothesis, ***cluster-level FWE-corrected p-values***, defined as the likelihood under the null hypothesis of finding one or more clusters with this or larger mass across the entire set of ROI-to-ROI connections, and ***cluster-level FDR-corrected p-values***, defined as the expected proportion of false discoveries among clusters having this or larger mass across the entire set of ROI-to-ROI connections.

A second standard criterion ("**standard settings for cluster-based inferences #2: Spatial Pairwise Clustering statistics**" in CONN's ROI-to-ROI results explorer gui) for thresholding ROI-to-ROI parametric maps that also appropriately controls family-wise error rates, uses randomization/permutation analyses with a combination of an uncorrected $p < 0.01$ height threshold in order to initially define clusters of interest, and a FDR-corrected $p < 0.05$ cluster-level threshold to select among the resulting clusters those deemed significant (clusters with larger mass than what we could reasonably expect under the null hypothesis)

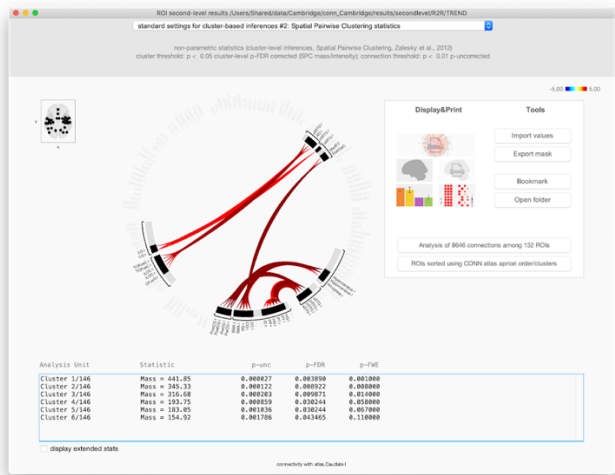
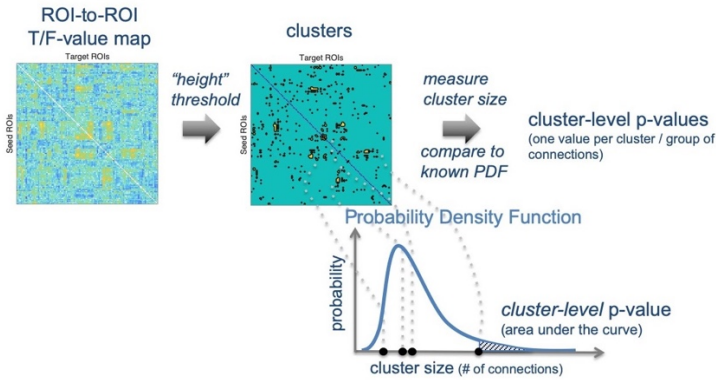


Figure 29. **Top:** schematic procedure to derive SPC cluster-level statistics (see text for details); **bottom:** example report of SPC cluster-level inferences

Threshold Free Cluster Enhancement (TFCE) statistics

Cluster-level inferences based on Threshold Free Cluster Enhancement analyses (Smith and Nichols 2007) in CONN can be also used in the context of ROI-to-ROI connectivity matrices. Similarly to SPC analyses, TFCE starts with the entire ROI-to-ROI matrix of T- or F- statistics estimated using a General Linear Model, with ROIs again sorted either manually by the user (e.g. from an atlas), or automatically using a hierarchical clustering procedure (optimal leaf ordering for hierarchical clustering, Bar-Joseph et al. 2001) based on ROI-to-ROI anatomical proximity or functional similarity metrics. Instead of thresholding this map using a priori height threshold, TFCE analyses proceed by computing the associated TFCE score map, combining the strength of the statistical effect for each connection with the extent of all clusters or groups of neighboring connections that would appear at this location when thresholding the original statistical parametric maps at any arbitrary height threshold. Then, as before, the expected distribution of TFCE values under the null hypothesis is numerically estimated using multiple (1,000 or higher) randomization/permutation iterations of the original data, and used to compute for each cluster in the original analysis a **peak-level FWE-corrected p-value** (defined as the likelihood under the null hypothesis of observing at least one or more connections with this or larger TFCE scores over the entire ROI-to-ROI connectivity matrix). Similarly, and following the approach in Chumbley et al. 2010, each local-extremum/peak in the TFCE map is compared to the null hypothesis distribution of local-peak TFCE values to estimate a **peak-level uncorrected p-value**, representing the likelihood under the null hypothesis of one randomly-selected peak in the TFCE map having this or larger scores, and associated **peak-level FDR-corrected p-values**, defined as the expected proportion of false discoveries among peaks having this or larger TFCE scores across the entire ROI-to-ROI matrix

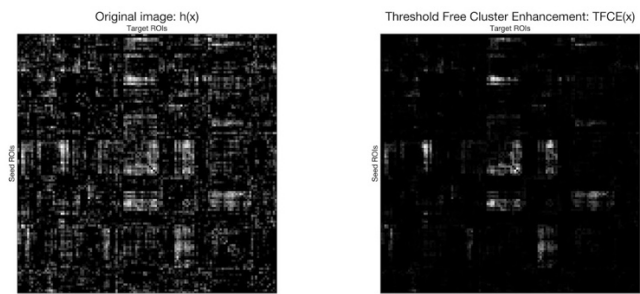


Figure 30. Example of TFCE operation. Original matrix (left) and associated TFCE-score matrix (right)

Yet another standard criterion ("**standard settings for cluster-based inferences #3: Threshold Free Cluster Enhancement statistics**" in CONN's ROI-to-ROI results explorer gui) for thresholding ROI-to-ROI parametric maps while appropriately controlling the family-wise error rate, uses TFCE analyses with a FWE-corrected $p < 0.05$ connection-level threshold to select among the resulting maps those groups of connections deemed significant (including only connections having TFCE scores larger than what we could reasonably expect under the null hypothesis).

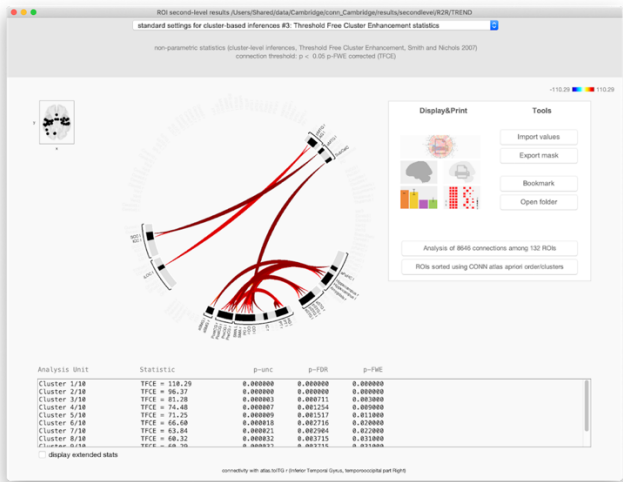


Figure 31. Example report of TFCE cluster-level inferences

Alternatives to cluster-level inferences: connection-, ROI-, and network- level inferences

In addition to the three main approaches listed above, CONN also implements alternative methods focusing on other units of inferential analyses, ranging from individual connections to entire networks of connections

The first approach ("***alternative settings for connection-based inferences, parametric univariate statistics***" in CONN's ROI-to-ROI results explorer gui) allows users to make inferences about individual connections, rather than focusing on groups of connections. In order to control family-wise error rates, this approach simply uses the standard Benjamini and Hochberg's FDR algorithm to compute for each individual connection (between all pairs of ROIs) a ***connection-level FDR-corrected p-value***, defined as the expected proportion of false discoveries among all connections with effects larger than this one across the entire ROI-to-ROI matrix. The default criterion uses a connection-level FDR-corrected $p < 0.05$ threshold to select among all connections those deemed significant (with larger effects than what we could reasonable expect under the null hypothesis)

The second approach ("***alternative settings for ROI-based inferences, parametric multivariate statistics***" in CONN's ROI-to-ROI results explorer gui) allows users to make inferences about individual ROIs. This approach uses the same general strategy as FNC but instead of defining sets/clusters of connections based on a data-driven clustering approach, it explicitly defines a different set/cluster of connections for each row of the ROI-to-ROI matrix, grouping all connections that arise from the same ROI as a new set/cluster. It then performs a multivariate parametric General Linear Model analysis for all connections included in each of these new sets/clusters of connections. This results in a F- statistic for each individual ROI and an associated ***uncorrected ROI-level p-value***, defined as the likelihood under the null hypothesis of a randomly selected ROI showing equal or larger effects than those observed at this ROI, and a ***FDR-corrected ROI-level p-value*** (Benjamini and Hochberg, 1995), defined as the expected proportion of false discoveries among all

ROIs with similar or larger effects across the entire set of ROIs included in the original analysis. The default criterion in CONN for this approach uses a ROI-level FDR-corrected $p < 0.05$ threshold in order to select among all ROIs those deemed significant (with larger effects than what we could reasonably expect under the null hypothesis), together with a post-hoc uncorrected $p < 0.01$ height (connection-level) threshold to help characterize the pattern of individual connections that show some of the largest effects from each significant ROI

The third approach ("***alternative settings for network-based inferences, Network Based Statistics***" in CONN's ROI-to-ROI results explorer gui) allows users to make inferences about entire networks of ROIs. CONN uses here the approach known as Network Based Statistics (Zalesky et al. 2010). Similar to SPC, it starts with the entire ROI-to-ROI matrix of T- or F- statistics estimated using a General Linear Model, forming a two-dimensional statistical parametric map. Unlike SPC or TFCE, the order of ROIs in this matrix is not relevant to these analyses. This statistical parametric map is then thresholded using an a priori "height" threshold (e.g. $T > 3$ or $p < 0.001$). The resulting suprathreshold connections define a graph among all nodes/ROIs. This graph is then broken down into components / networks, defined as connected subgraphs, and from here the procedure continues in the same way as SPC, but using networks instead of clusters as a basis to group multiple connections into a set. Each network is then characterized by its network mass (sum of F- or T-squared statistics over all connections within each cluster), and these values are compared to a distribution of expected network mass values under the null hypothesis, which is numerically estimated using multiple (1,000 or higher) randomization / permutation iterations of the original data. Results are summarized, for each individual network or group of connections, by ***uncorrected network-level p-values***, representing the likelihood of a randomly-selected network having this or larger mass under the null hypothesis, ***network-level FWE-corrected p-values***, defined as the likelihood under the null hypothesis of finding one or more networks with this or larger mass across the entire set of ROI-to-ROI connections, and ***network-level FDR-corrected p-values***, defined as the expected proportion of false discoveries among networks having this or larger

mass across the entire set of ROI-to-ROI connections. The default criterion in CONN for this approach uses a combination of a uncorrected $p < 0.001$ height threshold in order to initially define networks of interest, and a FDR-corrected $p < 0.05$ network-level threshold to select among the resulting networks those deemed significant (networks with larger mass than what we could reasonably expect under the null hypothesis)

Last, in addition to the default and alternative methods described above it is also possible ("**advanced family-wise error control settings**" in CONN's voxel-based or ROI-to-ROI results explorer gui) to use different choices and combinations of thresholds for any of the general procedures listed above, allowing a wide variety of possible analysis strategies. All default criteria in CONN are defined to ensure proper **analysis-wise** error control (i.e. appropriately control for all multiple comparisons within each second-level analysis) while affording reasonable sensitivity in most scenarios, but researchers are encouraged to explore different settings and use the method most appropriate to the specificities of their study. In order to avoid "p-hacking" (trying multiple combinations of analysis options but only reporting the one that works best) we strongly recommend researchers select the desired inferential approach and parameter choices a priori (e.g. based on pilot data, prior studies, or the literature most related to their sub-field), and/or attempt to appropriately control for all different analyses that have been run by applying an additional Bonferroni- or FDR- correction to the observed *cluster-level* p-values (e.g. to convert from "analysis-wise" FWE control to "study-wise" FWE control)

5.3. How to use cluster-level inferences

CONN's cluster-level inferences can be run using any of the following options:

Option 1: using CONN's gui

If you have analyzed your data in CONN, follow the instructions in the "How to run CONN General Linear Model analyses" section to define your second-level analysis. After that simply click on 'Results explorer' to have CONN launch the corresponding (voxel-level or ROI-to-ROI) results explorer window. The menu at the top of this window allows you to choose between the different standard criteria appropriate for this analysis (e.g. choose *standard settings for cluster-based inferences #1: Random Field Theory parametric statistics* option for cluster-level inferences based on Random Field Theory (RFT) parametric statistics in voxel-based analyses; alternatively choose the option labeled 'advanced Family-Wise Error control settings' for advanced options). Any significant clusters, using the selected cluster-level inferential approach, will be displayed and their associated cluster-level properties and statistics listed in table form below.

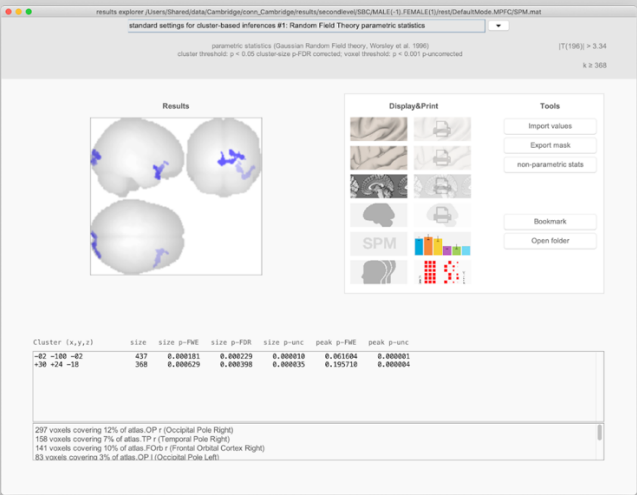


Figure 32. Example of CONN's second-level analysis results explorer gui

Option 2: using CONN's commands

If you have already defined and run the corresponding second-level GLM analysis (either from the gui or using CONN batch/modular commands; otherwise start with the instructions in the “How to run CONN General Linear Model analyses” section to define your second-level analysis) you may use the following MATLAB syntax:

```
conn display /myresults/SPM.mat
```

to launch the results explorer window for the voxel-based, surface-based, or ROI-to-ROI analyses saved in the directory /myresults (see [doc conn display](#) for additional options).

As before, the menu at the top of the results explorer window will allow you to choose between the different standard criteria appropriate for each analysis.

Bibliography

- Achard, S., & Bullmore, E. (2007). Efficiency and cost of economical brain functional networks. *PLoS Comput. Biol.* 3, e17
- Andersson JLR, Hutton C, Ashburner J, Turner R, & Friston K (2001). Modelling geometric deformations in EPI time series. *NeuroImage* 13:90-919
- Ashburner, J. & Friston, K. (1997). Multimodal image coregistration and partitioning—a unified framework. *Neuroimage*, 6(3), 209-217.
- Ashburner, J. and Friston, K.J.. (2005). Unified segmentation. *NeuroImage*, 26:839–851
- Bar-Joseph, Z., Gifford, D. K., & Jaakkola, T. S. (2001). Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(suppl_1), S22-S29.
- Behzadi, Y., Restom, K., Liao, J., & Liu, T. T. (2007). A component based noise correction method (CompCor) for BOLD and perfusion based fMRI. *Neuroimage*, 37(1), 90-101
- Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1), 289-300.
- Bullmore, E. T., Suckling, J., Overmeyer, S., Rabe-Hesketh, S., Taylor, E., & Brammer, M. J. (1999). Global, voxel, and cluster tests, by theory and permutation, for a difference between two groups of structural MR images of the brain. *IEEE transactions on medical imaging*, 18(1), 32-42.
- Calhoun, V. D., Wager, T. D., Krishnan, A., Rosch, K. S., Seymour, K. E., Nebel, M. B., ... & Kiehl, K. (2017). The impact of T1 versus EPI spatial normalization templates for fMRI data analyses. *Human brain mapping*, 38(11), 5331-5342.
- Calhoun, V. D., Adali, T., Pearlson, G. D., & Pekar, J. J. (2001). A method for making group inferences from functional MRI data using independent component analysis. *Human brain mapping*, 14(3), 140-151

Chai, X. J., Castañón, A. N., Öngür, D., & Whitfield-Gabrieli, S. (2012). Anticorrelations in resting state networks without global signal regression. *Neuroimage*, 59(2), 1420-1428

Chumbley, J., Worsley, K., Flandin, G., & Friston, K. (2010). Topological FDR for neuroimaging. *Neuroimage*, 49(4), 3057-3064.

Ciric, R., Wolf, D. H., Power, J. D., Roalf, D. R., Baum, G. L., Ruparel, K., ... & Gur, R. C. (2017). Benchmarking of participant-level confound regression strategies for the control of motion artifact in studies of functional connectivity. *Neuroimage*, 154, 174-187

Collignon, A., Maes, F., Delaere, D., Vandermeulen, D., Suetens, P., & Marchal, G. (1995, June). Automated multi-modality image registration based on information theory. In *Information processing in medical imaging* 3(6), 263-274

Deshpande, G., LaConte, S., Peltier, S., & Hu, X. (2009). Integrated local correlation: a new measure of local coherence in fMRI data. *Human brain mapping*, 30(1), 13-23

Esteban, O., Markiewicz, C. J., Blair, R. W., Moodie, C. A., Isik, A. I., Erramuzpe, A., ... & Oya, H. (2019). fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nature methods*, 16(1), 111-116.

Eklund, A., Nichols, T. E., & Knutsson, H. (2016). Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the national academy of sciences*, 113(28), 7900-7905.

Friston, K. J., Williams, S., Howard, R., Frackowiak, R. S., & Turner, R. (1996). Movement-related effects in fMRI time-series. *Magnetic resonance in medicine*, 35(3), 346-355.

Glover, G. H., Li, T. Q., & Ress, D. (2000). Image-based method for retrospective correction of physiological motion effects in fMRI: RETROICOR. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 44(1), 162-167.

Griffanti, L., Douaud, G., Bijsterbosch, J., Evangelisti, S., Alfaro-Almagro, F., Glasser, M. F., ... & Beckmann, C. F. (2017). Hand classification of fMRI ICA noise components. *Neuroimage*, 154, 188-205.

Hagler Jr, D. J., Saygin, A. P., & Sereno, M. I. (2006). Smoothing and cluster thresholding for cortical surface-based group analysis of fMRI data. *Neuroimage*, 33(4), 1093-1103.

- Hallquist, M. N., Hwang, K., & Luna, B. (2013). The nuisance of nuisance regression: spectral misspecification in a common approach to resting-state fMRI preprocessing reintroduces noise and obscures functional connectivity. *Neuroimage*, 82, 208-225
- Henson, R. N. A., Buechel, C., Josephs, O., & Friston, K. J. (1999). The slice-timing problem in event-related fMRI. *NeuroImage*, 9, 125
- Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3), 626-634
- Jafri, M. J., Pearlson, G. D., Stevens, M., & Calhoun, V. D. (2008). A method for functional network connectivity among spatially independent resting-state components in schizophrenia. *Neuroimage*, 39(4), 1666-1681.
- Latora, V., & Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical Review Letters* 87: 198701-4
- Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). Multivariate analysis. *Probability and Mathematical Statistics*, London: Academic Press.
- Martuzzi, R., Ramani, R., Qiu, M., Shen, X., Papademetris, X., & Constable, R. T. (2011). A whole-brain voxel based measure of intrinsic connectivity contrast reveals local changes in tissue connectivity with anesthetic without a priori assumptions on thresholds or regions of interest. *Neuroimage*, 58(4), 1044-1050
- Murphy, K., Birn, R. M., Handwerker, D. A., Jones, T. B., & Bandettini, P. A. (2009). The impact of global signal regression on resting state correlations: are anti-correlated networks introduced?. *Neuroimage*, 44(3), 893-905.
- Norman, K. A., Polyn, S. M., Detre, G. J., & Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends in cognitive sciences*, 10(9), 424-430
- Power, J. D., Mitra, A., Laumann, T. O., Snyder, A. Z., Schlaggar, B. L., & Petersen, S. E. (2014). Methods to detect, characterize, and remove motion artifact in resting state fMRI. *Neuroimage*, 84, 320-341
- Rao, C. R. (1951). An asymptotic expansion of the distribution of Wilks' criterion. *Bulletin of the International Statistical Institute*, 33(2), 177-180.
- Saad, Z. S., Reynolds, R. C., Jo, H. J., Gotts, S. J., Chen, G., Martin, A., & Cox, R. W. (2013). Correcting brain-wide correlation differences in resting-state FMRI. *Brain connectivity*, 3(4), 339-352

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics bulletin*, 2(6), 110-114.

Smith, S. M., & Nichols, T. E. (2009). Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. *Neuroimage*, 44(1), 83-98.

Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter / Kongelige Danske Videnskabernes Selskab* 5: 1-34.

Strang, G. (1993). *Introduction to linear algebra* (Vol. 3). Wellesley, MA: Wellesley-Cambridge Press

Studholme, Hill & Hawkes (1998). A normalized entropy measure of 3-D medical image alignment. *Proc. Medical Imaging 1998*, vol. 3338, San Diego, CA, pp. 132-143

Whitfield-Gabrieli, S., & Nieto-Castanon, A. (2012). Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks. *Brain connectivity*, 2(3), 125-141

Worsley, K. J., Marrett, S., Neelin, P., Vandal, A. C., Friston, K. J., & Evans, A. C. (1996). A unified statistical approach for determining significant signals in images of cerebral activation. *Human brain mapping*, 4(1), 58-73.

Yang, H., Long, X. Y., Yang, Y., Yan, H., Zhu, C. Z., Zhou, X. P., ... & Gong, Q. Y. (2007). Amplitude of low frequency fluctuation within visual areas revealed by resting-state functional MRI. *Neuroimage*, 36(1), 144-152.

Zalesky, A., Fornito, A., & Bullmore, E. T. (2010). Network-based statistic: identifying differences in brain networks. *Neuroimage*, 53(4), 1197-1207.

Zalesky, A., Fornito, A., & Bullmore, E. T. (2012). On the use of correlation as a measure of network connectivity. *Neuroimage*, 60(4), 2096-2106.

Zou, Q. H., Zhu, C. Z., Yang, Y., Zuo, X. N., Long, X. Y., Cao, Q. J., ... & Zang, Y. F. (2008). An improved approach to detection of amplitude of low-frequency fluctuation (ALFF) for resting-state fMRI: fractional ALFF. *Journal of neuroscience methods*, 172(1), 137-141.