

Ironsmith QSM

Copyright (C) 2021 Valentinos Zachariou, University of Kentucky (see LICENSE file for more details). Third party software provided with Ironsmith are subject to their own licenses and copyrights (see section 7 for details).

This software has been developed for research purposes only and is not a clinical tool.

Description:

Ironsmith is a comprehensive, fully automated pipeline for creating and processing Quantitative Susceptibility Maps (QSM), extracting QSM based iron concentrations from subcortical and cortical brain regions and evaluating the quality of QSM data using per ROI SNR measures. Ironsmith can also filter out per-ROI outlier QSM values (such as values associated with large veins) and offers a precise, CSF-only reference region for QSM reconstruction to minimize partial volume effects.

Ironsmith can perform the following tasks:

- a) Automate the process of creating QSM maps from GRE DICOM images using MEDI Toolbox **(see section 7 for details)** .
- b) Register MPR or multi-echo MPR (MEMPR) T1 images to QSM maps and then segment these into 89 ROIs **(ROI list in section 8)** using FreeSurfer.
- c) Filter outlier voxels from these ROIs (default: QSM values larger than 97th percentile of values), extract QSM based iron concentration and output the results into CSV formatted tables.
- d) Calculate SNR (GRE magnitude image based) for each ROI and output SNR values into CSV tables.
- e) Identify outlier regions on phase images using median absolute deviation (MAD); calculate percent overlap between phase-image outlier regions and each ROI; Output results into CSV tables.
- e) Non-linearly Warp QSM maps and aligned MPR/MEMPR to MNI152 1mm space. This step allows users to (1) extract QSM values from standard space ROIs not included with Ironsmith and (2) conduct voxelwise QSM analyses.
- f) Process single or multiple participants in parallel (multiple instances and nohup supported).
- g) Provide comprehensive user feedback and detailed error reporting. When an error or warning occurs, Ironsmith does not display cryptic messages but instead provides detailed reports of what might have gone wrong and how a user can fix the error/warning.

Release Notes (latest release):

Ironsmith QSM v1.2

Can be installed via `git clone https://github.com/vzachari/IronSmithQSM.git`
or via direct download as described in *section #2, Installation*.

Updates:

1. Phase image quality control procedure added.
- Ironsmith will automatically Identify artifacts on Relative Difference Field (RDF) images (unwrapped phase images with the background field removed) and report the perecent overlap of these with the 89 supported ROIs in a CSV formatted output file (see section #6, Outputs).
1. FreeSurfer updated to v7.1.1 in singularity container (reduces recon-all time to around 5 hours from 8 hours).
- Updated singularity container needs to be downloaded. See section 2b (Download QSM_Container.simg).
1. Fsleyes can now be accessed within the singularity container via the Ironsmith_fsleyes command.
2. Minor bug fixes.
3. Updates to README.md and README.pdf for clarity/readability.

NOTE:

Due to the FreeSurfer update, the per-ROI QSM values are not identical between Ironsmith v1.0 and v1.2.

Correlation of QSM values extracted from the 89 ROIs offered by Ironsmith, between v1.0 and v1.2, yielded an r^2 value of 0.99.

For this reason, we do not recommend updating Ironsmith mid-analyses. All data should be analyzed with either v1.0 OR v1.2 (FreeSurfer 6.0.0 or FreeSurfer 7.1.1).

1) Software requirements:

a) Operating system

Unix

Any Linux distribution that supports Singularity (see point c).

Ironsmith tested on:

Red Hat Enterprise Linux Workstation release 7.8 (Maipo)

Windows 10 WSL2

Ironsmith tested on:

Ubuntu 18.04 running on Windows 10 Subsystem for Linux V2 (WSL2)

Ubuntu 16.04 running on Windows 10 Subsystem for Linux V2 (WSL2)

b) MATLAB

Ironsmith requires Matlab to run MEDI Toolbox and supports versions R2017b to R2019b.

MEDI Toolbox requires the Matlab `Image Processing Toolbox` to be installed.

Matlab is not needed if MEDI is not required.

c) Singularity

Ironsmith tested on Singularity versions 3.5.2 and 3.5.3

Installation guide:

<https://sylabs.io/guides/3.5/admin-guide/installation.html>

d) Bash UNIX shell version 4.2.46(2) or later with GNU coreutils

e) MEDI Toolbox version 01/15/2020

Ironsmith requires MEDI Toolbox if QSM maps need to be generated

Download from:

<http://pre.weill.cornell.edu/mri/pages/qsm.html>

MEDI Toolbox is not required if QSM maps are already available

Currently only MEDI Toolbox version 01/15/2020 is supported

NOTE: 10.8 GB of free space is required for each instance of IronSmith running (see section #5, parallel processing)

2) Installation:

a) Download Ironsmith QSM

Option 1: download from github

Visit <https://github.com/vzachari/IronSmithQSM>

Click on tags

Click on desired IronsmithQSM version (release notes are displayed by clicking the three dots ...)

Click on source code link (zip or tar.gz) to download

NOTE: Via this download option the IronSmithQSM folder will be **IronSmithQSM-version#** (Ex. *IronSmithQSM-1.0*).

Option 2: using git

```
git clone https://github.com/vzachari/IronSmithQSM.git && cd IronSmithQSM && git
checkout v1.0
```

NOTE: `git checkout v1.0` can be replaced with a different version number. Type `git tag -l` from within the IronSmithQSM folder for a list of available versions.

b) Download QSM_Container.simg

For Ironsmith v1.2 (09/29/2021): FreeSurfer7.1.1 (10.8GB)

From: <https://drive.google.com/file/d/1NFV2z0yIEPKGbIQVrcMe8bza8ZS21AO7/view?usp=sharing>

For Ironsmith v1.0 (06/04/2021): FreeSurfer6.0.0 (8.3GB)

From: https://drive.google.com/file/d/1Q1KC665q1Pv_GSWPWH9cfUsoJcd-dxEB/view?usp=sharing

c) Place QSM_Container.simg in IronSmithQSM/Functions

d) Download MEDI Toolbox version 01/15/2020 (~7MB)

From: <http://pre.weill.cornell.edu/mri/pages/qsm.html>

e) Unzip MEDI Toolbox (typically MEDI_toolbox.zip)

f) Place MEDI_toolbox folder (folder with README.m, UPDATES.m etc) into IronSmithQSM/Functions

NOTE: Make sure the MEDI_toolbox folder in IronSmithQSM/Functions does not have another MEDI_toolbox folder in it (e.g MEDI_toolbox/MEDI_toolbox).

g) Edit IronSmithQSM/Matlab_Config.txt with the path to the matlab executable on your system

(e.g. `/usr/local/MATLAB/R2019b/bin/matlab`)

Supported versions R2017b to R2019b.

NOTE: the Image Processing Toolbox needs to be installed for the matlab version specified in IronSmithQSM/Matlab_Config.txt.

h) Add the IronSmithQSM directory to \$PATH

Guide: <https://opensource.com/article/17/6/set-path-linux>

3) Syntax:

Ironsmith [MyInputFile] [absolute path to output folder]

Example: "Ironsmith File.csv /home/data/MyAmazingExp/QSM_Analysis"

- a) The output folder does not need to exist.
- b) The output folder does not need to be empty but Ironsmith will skip any participant specified in MyInputFile that has a corresponding folder inside the output folder.

Ex. if S0001 is specified in MyInputFile and folder S0001 exists in output folder, S0001 will be skipped.

- c) Absolute path to MyInputFile needs to be provided
(Ex. /home/data/MyAmazingExp/CSVFileVault/File.csv) if MyInputFile is not in current folder.

- d) Freesurfer_Skip is a reserved folder name under output folder and may be used by Ironsmith. See section #5 on optional features below.

4) MyInputFile format:

- a) MyInputFile has to be CSV formatted (entries separated by commas ',').
- b) MyInputFile can be created in Excel (MS Windows or MacOS) and saved as a CSV (Comma delimited) file or in a Unix text editor (e.g. Gedit, Atom, Emacs).
- c) Each row in MyInputFile corresponds to a different participant.

see Example_File.csv in IronSmithQSM folder:

If MEDI Toolbox is required to create QSM images/maps:

Column1 = Subj (nominal subject variable e.g. S0001 or 01 or Xanthar_The_Destroyer)

Column2 = MEDI_Yes <-- this is case sensitive

Column3 =

Either absolute path to directory with MPR/MEMPR files **OR** absolute path to a single NIFTI (.nii or

.nii.gz) MPR/MEMPR file

(e.g. /home/subjects/S01/MPR **OR** /home/subjects/S01/MPR/S01_MPR.nii.gz)

If a single NIFTI (.nii or .nii.gz) MPR/MEMPR file is provided:

File can have any name.

File can have multiple volumes, each corresponding to a different echo (RMS will be calculated).

File can have a single echo/volume.

If path to directory with MPR/MEMPR files is provided:

MPR/MEMPR files in provided folder can be:

a) DICOMS

Only MPR/MEMPR DICOMS should be present in the MPR folder if you want Ironsmith to process DICOMS. If NIFTI files exist together with DICOMS, they will be selected instead (see "b" below).

b) Multiple .nii/.nii.gz files each corresponding to a different echo (will be catenated and RMS will be calculated).

To make sure the correct files are selected, each NIFTI file needs to have _e# in the file name, where # is the echo number.

(e.g. S0001_MEMPR_e1.nii.gz, S0001_MEMPR_e2.nii.gz...)

This is the default **dcm2niix** output format for multiple echos.

c) Single .nii/.nii.gz file with multiple volumes, each corresponding to a different echo (RMS will be calculated).

This single NIFTI file can have any name.

d) Single .nii/.nii.gz file with a single echo/volume.

This can be rms/averaged across echos or just a single echo T1 MPAGE.

This single NIFTI file can have any name and will be used as is.

Column4 = Absolute path to folder with QSM DICOM files

(e.g. /home/subjects/S01/QSM_Dicom)

The QSM DICOM folder must include DICOMS for both GRE magnitude and phase. T2* DICOMS that are sometimes saved as part of a GRE sequence can be present in the QSM DICOM folder and will be ignored.

Preferably only DICOMS should be present in the QSM DICOM folder. However, Ironsmith can filter out the following filetypes .nii .json .txt .nii.gz .HEAD .BRIK .hdr .img

All 4 columns need to be provided, otherwise Ironsmith will exit with errors.

If QSM maps and GRE magnitude images are already available:

Column1 = Subj (nominal subject variable e.g. S0001 or 01 or Xanthar_The_Destroyer)

Column2 = MEDI_No <-- This is case sensitive

Column3 =

Either absolute path to directory with MPR/MEMPR files **OR** absolute path to a single NIFTI (.nii or .nii.gz) MPR/MEMPR file

(e.g. /home/subjects/S01/MPR **OR** /home/subjects/S01/MPR/S01_MPR.nii.gz)

If a single NIFTI (.nii or .nii.gz) MPR/MEMPR file is provided:

File can have any name.

File can have multiple volumes, each corresponding to a different echo (RMS will be calculated).

File can have a single echo/volume.

If path to directory with MPR/MEMPR files is provided:

MPR/MEMPR files in provided folder can be:

a) DICOMS

Only MPR/MEMPR DICOMS should be present in the MPR folder if you want Ironsmith to process DICOMS. If NIFTI files exist together with DICOMS, they will be selected instead (see "b" below).

b) Multiple .nii/.nii.gz files each corresponding to a different echo (will be catenated and RMS will be calculated).

To make sure the correct files are selected, each NIFTI file needs to have _e# in the file name, where # is the echo number.

(e.g. S0001_MEMPR_e1.nii.gz, S0001_MEMPR_e2.nii.gz...)

This is the default **dcm2niix** output format for multiple echos.

c) Single .nii/.nii.gz file with multiple volumes, each corresponding to a different echo (RMS will be calculated).

This single NIFTI file can have any name.

d) Single .nii/.nii.gz file with a single echo/volume.

This can be rms/averaged across echos or just a single echo T1 MPAGE.

This single NIFTI file can have any name and will be used as is.

Column4 = Absolute path including filename to QSM magnitude image (.nii or .nii.gz)
(e.g. /home/subjects/S01/QSM/QSM_Magnitude.nii.gz)

Column5 = Absolute path including filename to QSM map (.nii or .nii.gz)
(e.g. /home/subjects/S01/QSM/QSM_Map.nii.gz)

All 5 columns need to be provided, otherwise Ironsmith will exit with errors.

5) Optional features:

Skipping FreeSurfer segmentation

If FreeSurfer has already run and a participant has a completed FreeSurfer recon-all -all segmentation folder, Ironsmith can skip the FreeSurfer segmentation step by doing the following:

a) Copy the FreeSurfer recon-all folder (the one containing the *label*, *mri*, *scripts*, *stats*, *surf...* folders) into **/OutputFolder/Freesurfer_Skip**, where **OutputFolder** is the one specified/to be specified in the Ironsmith command.

You can create the /OutputFolder/Freesurfer_Skip folder or Ironsmith will create it for you if you have run it at least once previously for OutputFolder.

b) Rename the recon-all folder to **Subj_FreeSurfSeg_Skull**. Subj should match the one provided in MyInputFile and should correspond to the participant you want the segmentation step skipped.

Note: if Ironsmith runs FreeSurfer, it will create **Subj_FreeSurfSeg_Skull** and place it under **/OutputFolder/Subj/MPR**. This helps reduce processing time if for any reason one would like to repeat the analysis on a given participant (e.g. due to a crash or errors). Just copy/move this folder over to **/OutputFolder/Freesurfer_Skip**, delete the problematic participant folder (e.g. **/OutputFolder/Subj**) and re-run Ironsmith.

Processing participants in parallel

Parallel processing can significantly increase the speed of analyses. Running the Ironsmith command with the same MyInputFile and output folder in multiple terminal windows allows for parallel processing of participants specified in MyInputFile.

For example, running three instances of Ironsmith:

Terminal 1:

```
Ironsmith File.csv /home/data/MyAmazingExp/QSM_Analysis
```

Terminal 2:

```
Ironsmith File.csv /home/data/MyAmazingExp/QSM_Analysis
```

Terminal 3:

```
Ironsmith File.csv /home/data/MyAmazingExp/QSM_Analysis
```

Terminals 1-3 will each be running a different instance of Ironsmith (each working on a different set of participants) but all instances will be working on the same group/list of participants (from **File.csv**) and in the same output folder (**/home/data/MyAmazingExp/QSM_Analysis**) and will only create a single set of group output files (see section #6 below).

NOTE: nohup can also be used with parallel processing:

*For example, running three **nohup** instances of Ironsmith:*

Terminal 1:

```
bash (press enter to switch to bash)
```

```
nohup Ironsmith File.csv /home/data/MyAmazingExp/QSM_Analysis &> Ironsmith_Inst_1.txt &
```

```
nohup Ironsmith File.csv /home/data/MyAmazingExp/QSM_Analysis &> Ironsmith_Inst_2.txt &
```

```
nohup Ironsmith File.csv /home/data/MyAmazingExp/QSM_Analysis &> Ironsmith_Inst_3.txt &
```

To monitor the nohup progress of an Ironsmith instance, locate the Ironsmith_Inst_#.txt file (in directory where the nohup command was executed) and use the following command:

```
tail -f Ironsmith_Inst_#.txt
```

ctrl+c exits the tail -f process.

Adding participants to an existing analysis/output folder

To add participants to an existing Ironsmith analysis, simply edit MyInputFile.csv with the info of the additional participants and rerun the Ironsmith command. Ironsmith will skip all participants that have been previously processed (without errors) and will proceed to process the newly added participants. Lastly, the QSM/SNR/MAD values of the new participants will be added to the group output files in **OutputFolder**.

What if Ironsmith fails while writing data to output files (computer resets, power outage etc)?

Ironsmith can identify corrupted output files and will attempt to correct the damage by reverting these back to the last participant that completed successfully. This error is clearly communicated to the user both on screen and in log files so a corrupted/unfinished participant can be re-processed.

Viewing output NIFTI files

If you do not have a NIFTI viewer, AFNI or fsleyes can be launched from within the QSM_Container.simg by using the *Ironsmith_AFNI* or *Ironsmith_fsleyes* commands. Just type `Ironsmith_AFNI` or `Ironsmith_fsleyes` from within the folder you would like to view NIFTI files from (provided the IronSmithQSM directory (the installation folder) was added to \$PATH; see section 2: Installation).

AFNI viewer documentation:

https://afni.nimh.nih.gov/pub/dist/edu/latest/afni_handouts/afni03_interactive.pdf

Fsleyes documentation:

<https://open.win.ox.ac.uk/pages/fsl/fsleyes/fsleyes/userdoc/>

6) Outputs:

Each participant processed by Ironsmith will have a corresponding folder in **OutputFolder**. For example, if "S0001" is one of the participants processed, then **OutputFolder/S0001** will be created and populated with data.

a) All FreeSurfer based masks/ROIs are placed under:

S0001/QSM/FreeSurf_QSM_Masks/Cort_Mask_AL_QSM_RS_Erx1

S0001/QSM/FreeSurf_QSM_Masks/SubC_Mask_AL_QSM_RS_Erx1

b) All QSM maps/images created are placed under S0001/QSM/FreeSurf_QSM_Masks and are labelled as:

Subj_QSM_Map_FSL.nii.gz <-- Whole brain CSF, segmented by MEDI Toolbox, as the QSM reference structure (default MEDI)

Subj_QSM_Map_New_CSF_FSL.nii.gz <-- Lateral ventricles as the QSM reference structure

Subj_QSM_Map_New_WM_FSL.nii.gz <-- White matter as the QSM reference structure

c) All QSM maps warped to MNI space are placed under

S0001/QSM/FreeSurf_QSM_Masks/MNI152_QSM

d) QSM per ROI means (89 ROIs) are under **OutputFolder/Group** as follows:

Group_QSM_Mean.csv

Group_QSM_Mean_CSF.csv

Group_QSM_Mean_WM.csv

Group_QSM_ADJ_Mean.csv

Group_QSM_ADJ_Mean_CSF.csv

Group_QSM_ADJ_Mean_WM.csv

Group_QSM_SNR.csv

Group_QSM_MAD.csv

_Mean = Using only positive QSM voxels

_ADJ_Mean = Using only positive QSM voxels and adjusting for ROI size (*sum of all positive QSM voxels / Number of all voxels within an ROI*)

_CSF = Lateral ventricles as the QSM reference structure

_WM = White matter as the QSM reference structure

_SNR = Per ROI SNR values (see **SNR** description below)

_MAD = Percent overlap between phase image outlier regions and each of the 89 supported ROIs (see **Phase QC** description below).

NOTE: For each ROI, only QSM voxels with values less than the 97th percentile of all positive QSM values are included in averages. This percentile cutoff point for outliers can be modified by manually editing the header of the **05_Extract_QSM.sh** script file (line 38) under the Ironsmith installation folder:

```
#Percentile cutoff for outlier removal. Edit Percnt variable to change outlier cutoff  
Percnt="97"
```

SNR:

SNR is calculated as follows:

Mean signal intensity of magnitude image (root mean square of all echos) within an ROI / standard deviation of magnitude signal outside the head (away from the frequency and phase axes).

Lastly, SNR is multiplied by the Rayleigh distribution correction factor $\sqrt{(2-\pi/2)}$.

The outside of the head mask used for SNR can be found in a participants folder within the **OutputFolder**. For example:

/OutputFolder/S0001/QSM/FreeSurf_QSM_Masks/S0001_QSM_Mag_FSL_rms_OH_Mask.nii.gz

Phase QC:

Median Absolute deviation (MAD) based outlier regions on phase images are identified as follows: first, using MEDI Toolbox, a relative difference field (RDF) image is created by unwrapping (region

growing method) the input phase image and removing the background field (using projections onto dipole fields). Then, the median of an RDF image is calculated, constrained within an aligned, Freesurfer-derived whole-brain (WB) mask, eroded by one voxel. The median is calculated using the FSL function `fsfstats`. Then, using the same FSL function and WB mask, the median is subtracted from every voxel of the RDF image and the absolute value of the outcome is saved into a new intermediate map/image. The median of this intermediate map/image is then calculated to get the MAD of the RDF image. Positive and negative thresholds for outlier voxels are subsequently calculated and correspond to the median of the RDF image $\pm (5 * MAD)$. These outlier thresholds have been determined by testing the outlier detection procedure on 35 participants. Next, a MAD-based outlier mask is created using the AFNI function `3dcalc` by identifying all voxels within an RDF image (WB-mask constrained) lower or higher than the positive and negative outlier thresholds calculated in the previous step. Lastly, the percent overlap between a MAD-based outlier mask and each of the 89 anatomical ROIs supported by Ironsmith is calculated and saved in an output file labelled `Group_QSM_MAD.csv` within the output directory (`output directory/Group/ Group_QSM_MAD.csv`).

7) Ironsmith uses the following software, provided in the form of a Singularity image:

AFNI

RW Cox. AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research*, 29:162-173, 1996.

RW Cox and JS Hyde. Software tools for analysis and visualization of FMRI Data. *NMR in Biomedicine*, 10:171-178, 1997.

S Gold, B Christian, S Arndt, G Zeien, T Cizadlo, DL Johnson, M Flaum, and NC Andreasen. Functional MRI statistical software packages: a comparative analysis. *Human Brain Mapping*, 6:73-84, 1998.

dcm2niix

Li, Xiangrui, et al. "The first step for neuroimaging data analysis: DICOM to NIFTI conversion." *Journal of neuroscience methods* 264 (2016): 47-56.

FreeSurfer

Dale, A.M., Fischl, B., Sereno, M.I., 1999. Cortical surface-based analysis. I. Segmentation and surface reconstruction. Neuroimage 9, 179-194.

Dale, A.M., Sereno, M.I., 1993. Improved localization of cortical activity by combining EEG and MEG with MRI cortical surface reconstruction: a linear approach. J Cogn Neurosci 5, 162-176.

Desikan, R.S., Segonne, F., Fischl, B., Quinn, B.T., Dickerson, B.C., Blacker, D., Buckner, R.L., Dale, A.M., Maguire, R.P., Hyman, B.T., Albert, M.S., Killiany, R.J., 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. Neuroimage 31, 968-980.

FSL

M.W. Woolrich, S. Jbabdi, B. Patenaude, M. Chappell, S. Makni, T. Behrens, C. Beckmann, M. Jenkinson, S.M. Smith. Bayesian analysis of neuroimaging data in FSL. NeuroImage, 45:S173-86, 2009

S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, M. De Luca, I. Drobnjak, D.E. Flitney, R. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J.M. Brady, and P.M. Matthews. Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage, 23(S1):208-19, 2004

M. Jenkinson, C.F. Beckmann, T.E. Behrens, M.W. Woolrich, S.M. Smith. FSL. NeuroImage, 62:782-90, 2012

Neurodocker

The singularity containers provided with Ironsmith have been created using Neurodocker:

<https://github.com/ReproNim/neurodocker>

8) ROI List:

L_ = Left hemisphere

R_ = Right hemisphere

LR_ = Bilateral

_GM = Gray matter

LR_Frontal_Lobe_GM

LR_Parietal_Lobe_GM

LR_Occipital_Lobe_GM
LR_Temporal_Lobe_GM
L_CaudalAnteriorCingulate_GM
L_CaudalMiddleFrontal_GM
L_Cuneus_GM
L_DLPFC_GM
L_Entorhinal_GM
L_Frontal_GM
L_Fusiform_GM
L_InferiorParietal_GM
L_AngularGyrus_GM
L_InferiorTemporal_GM
L_Insula_GM
L_IsthmusCingulate_GM
L_LateralOccipital_GM
L_LateralOrbitofrontal_GM
L_Lingual_GM
L_MedialOrbitofrontal_GM
L_MiddleTemporal_GM
L_Occipital_GM_Mask
L_Parietal_GM_Mask
L_Temporal_GM_Mask
L_Parahippocampal_GM
L_Pericalcarine_GM
L_Postcentral_GM
L_PosteriorCingulate_GM
L_Precentral_GM
L_Precuneus_GM
L_RostralMiddleFrontal_GM
L_RostralAnteriorCingulate_GM
L_SuperiorFrontal_GM
L_SuperiorParietal_GM
L_SuperiorTemporal_GM
L_TransverseTemporal_GM
R_CaudalAnteriorCingulate_GM
R_CaudalMiddleFrontal_GM
R_Cuneus_GM
R_DLPFC_GM

R_Entorhinal_GM
R_Frontal_GM_Mask
R_Fusiform_GM
R_InferiorParietal_GM
R_AngularGyrus_GM
R_InferiorTemporal_GM
R_Insula_GM
R_IsthmusCingulate_GM
R_LateralOccipital_GM
R_LateralOrbitofrontal_GM
R_Lingual_GM
R_MedialOrbitofrontal_GM
R_MiddleTemporal_GM
R_Occipital_GM_Mask
R_Parietal_GM_Mask
R_Temporal_GM_Mask
R_Parahippocampal_GM
R_Pericalcarine_GM
R_Postcentral_GM
R_PosteriorCingulate_GM
R_Precentral_GM
R_Precuneus_GM
R_RostralMiddleFrontal_GM
R_RostralAnteriorCingulate_GM
R_SuperiorFrontal_GM
R_SuperiorParietal_GM
R_SuperiorTemporal_GM
R_TransverseTemporal_GM
LR_Accumbens_area
LR_Amygdala
LR_Caudate
LR_Hipp
LR_Pallidum
LR_Putamen
LR_Thalamus_Proper
L_Accumbens_area
L_Amygdala
L_Caudate

L_Hipp
L_Pallidum
L_Putamen
L_Thalamus_Proper
R_Accumbens_area
R_Amygdala
R_Caudate
R_Hipp
R_Pallidum
R_Putamen
R_Thalamus_Proper