

Protecting the Aging Brain – Diet Study

Dataset:

I have acquired two distinct datasets from OpenNeuro.

The first dataset comprises scans of 12 participants who underwent imaging under three distinct conditions:

1. On a standard non-ketogenic diet.
2. After following a standard diet and fasting for 12 hours.
3. After being on a ketogenic diet for one week.

The second dataset involves scans of 30 participants under three conditions:

1. After adhering to a standard diet and fasting for 12 hours (pre-bolus).
2. Following the 12-hour fast, with a bolus of glucose drink.
3. Following the 12-hour fast, with a bolus of ketogenic drink.

Due to space limitations, I've downloaded only 7 subjects from the second dataset, as the scanned data files are quite large. I've organized the data into separate folders for each dataset. Within each dataset's folder, I've created individual folders for every participant, and within those, I've placed their respective T1 and BOLD images.

Dataset Preprocessing:

Our initial goal is to eliminate any irrelevant noise present in the datasets. To achieve this, we embark on preprocessing the datasets.

Motion Correction:

As a part of this preprocessing stage, we focus on addressing minor movements, including subtle head or body shifts. These minute movements can result in undesired blurring, distortion, and misalignment within the images. Our aim is to rectify these issues and enhance the quality of the data.

Command:

```
3dvolreg -prefix output;
```

Motion Correcting for all subjects:

```
#!/bin/bash
```

```
subs=$(ls -d dataset2/sub*)

for sub in $subs; do
    cd ${sub}
    gzs=$(ls *gz)
    for gz in $gzs; do
        3dvolreg -prefix $gz -overwrite $gz;
    done
    cd ../../
done

Done
```

Extract First Volume:

Selected from the motion-corrected dataset, the initial volume represents the image captured at the beginning of the scanning session, subsequent to the application of motion correction procedures.

Command

```
3dTcat -prefix output mc_data[0]
```

Extracting first volume for all subjects:

```
#!/bin/bash

subs=$(ls -d dataset2/sub*)

for sub in $subs; do
    cd ${sub}
    gzs=$(ls *.gz)
    for gz in $gzs; do
        3dTcat -prefix vol_${gz} ${gz}[0]
    done
    cd ../../
done

done
```

Registration:

During this phase, our objective is to bring the datasets into alignment by mapping them onto a shared coordinate system.

Performing registration across all participants

```
#!/bin/bash

subs=$(ls -d dataset2/sub*)

for sub in $subs; do
    cd ${sub}
    gzs=$(ls vol*)
    for gz in $gzs; do
        antsRegistration --dimensionality 3 --float 0 --output
[to_${gz}/.nii/},affine_${gz}] --interpolation Linear --transform
Affine[0.1] --metric MI[../ref.nii.gz,$gz,1,32, Regular, 0.25]
--convergence [1000x500x250x100,1e-6,10] --shrink-factors 8x4x2x1
--smoothing-sigmas 3x2x1x0vox -v
    done
    cd ../../
Done
```

Bandpass Filtering:

In this stage, our aim is to isolate a targeted frequency range within a signal while reducing the influence of frequencies lying outside this designated range.

Command:

```
3dTproject -input input_filename -prefix output_filename -passband
0.01 0.1
```

Performing bandpass filtering across all participants

```
#!/bin/bash

subs=$(ls -d dataset2/sub*)

for sub in $subs; do
    cd ${sub}
    gzs=$(ls sub*.nii)
    for gz in $gzs; do
        3dTproject -input $gz -prefix bp_${gz} -passband 0.01 0.1
    done
    cd ../../
done
```

RSFC Metrics:

During this phase, these RSFC metrics offer understanding into the coordinated activity and communication patterns within the brain's inherent networks.

ReHo:

At this stage, ReHo helps to evaluate how closely the time series of a specific voxel aligns with those of its neighboring voxels.

Command

```
3dReHo -prefix reho_outputfile -inset bp_file
```

Generating ReHo for all BP dataset

```
#!/bin/bash

subs=$(ls -d dataset2/sub*)

for sub in $subs; do
    cd ${sub}
    bps=$(ls bp*.nii)
    for bp in $bps; do
        3dReHo -prefix reho_${bp} -inset $bp
    done
    cd ../../
done
```

ReHo Transformation:

```
subs=$(ls -d dataset1/sub*)

for sub in $subs; do
    cd ${sub}
    rehos=$(ls reho*.gz)
    for reho in $rehos; do
        matname=${reho/reho_bp_/to_vol_}
        matname=${matname/.nii.gz/0GenericAffine.mat}
        echo $matname
    done
done
```

```

    antsApplyTransforms -d 3 -e 0 -i $reho ../ref.nii.gz -o
inref_${reho} -t $matname
done
cd ../../
done

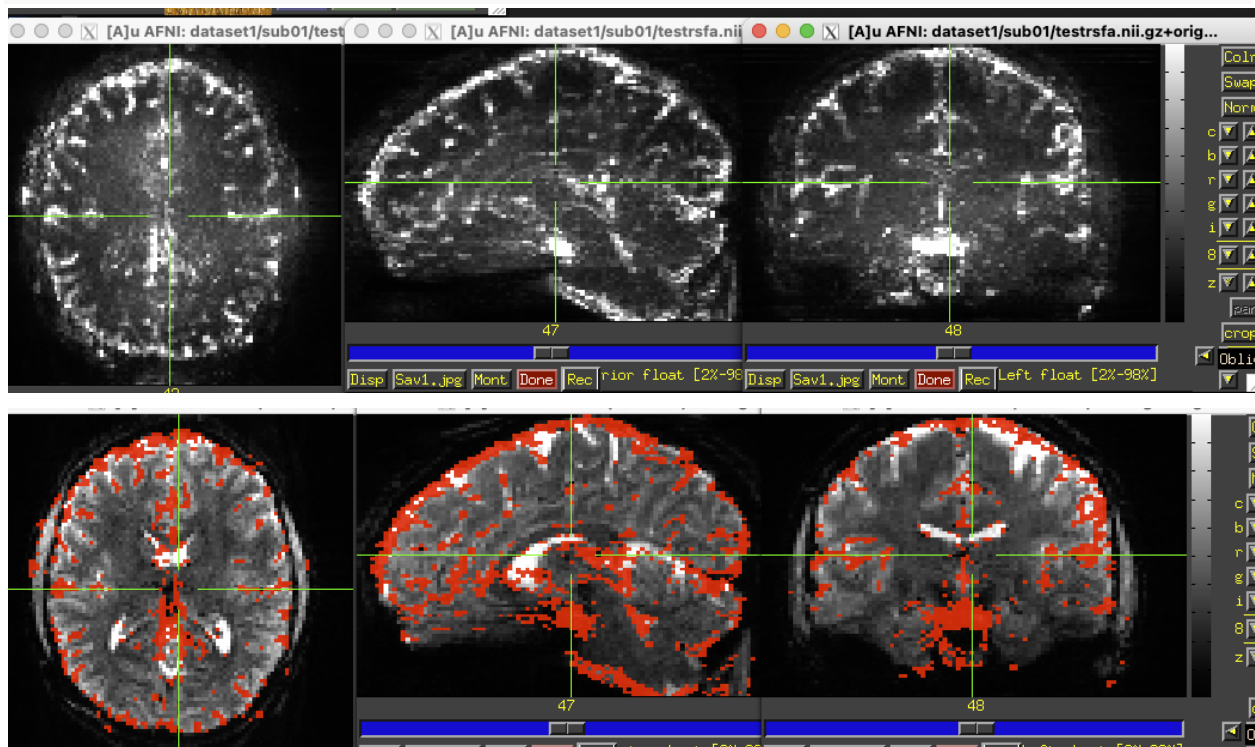
```

Stacking 3d to 4d ReHo Image:

```

3dTcat -prefix all_reho_fast.nii.gz sub01/reho sub02/reho ...
sub11/reho

```



Generated fALFF and RSFA using 3dRSFC

```

subs=$(ls -d dataset1/sub*)

for sub in $subs; do
  cd ${sub}
  bps=$(ls bp*.gz)
  for bp in $bps; do
    rsfc_name=rsfc_metric_${bp/.nii.gz/}
    3dRSFC -prefix $rsfc_name 0.01 0.1 $bp
    3dAFNItoNIFTI -prefix rsfc_metric_RSFA_${bp/.nii.gz/}

```

```

${rsfc_name}_RSFA+orig
    3dAFNItoNIFTI -prefix rsfc_metric_fALFF_${bp/.nii.gz/}
${rsfc_name}_fALFF+orig
    rm *.HEAD
    rm *.BRIK
done
cd ../../
done

```

fALFF Transformation:

```

import os
from nipype.interfaces.ants import ApplyTransforms
dataset_dir = 'dataset1'

# List all subdirectories
subs = [subdir for subdir in os.listdir(dataset_dir) if
os.path.isdir(os.path.join(dataset_dir, subdir))]
# subs = ["sub01"]
for sub in subs:
    print(sub)
    sub_dir = os.path.join(dataset_dir, sub)
    os.chdir(sub_dir)

    # List reho*gz files
    falffs = [falff for falff in os.listdir('.') if
falff.startswith('rsfc_metric_fALFF') and falff.endswith('.nii')]
    # rehos = ["reho_bp_sub-001_ses-ket_task-rest_run-1_bold.nii.gz"]
    for falff in falffs:
        # print(falff)
        # Generate the matname
        matname = falff.replace('rsfc_metric_fALFF_bp_',
'to_vol_').replace('.nii', '0GenericAffine.mat')
        # print(matname)

    # Create an instance of the ApplyTransforms interface
    at = ApplyTransforms()
    at.inputs.dimension = 3

```

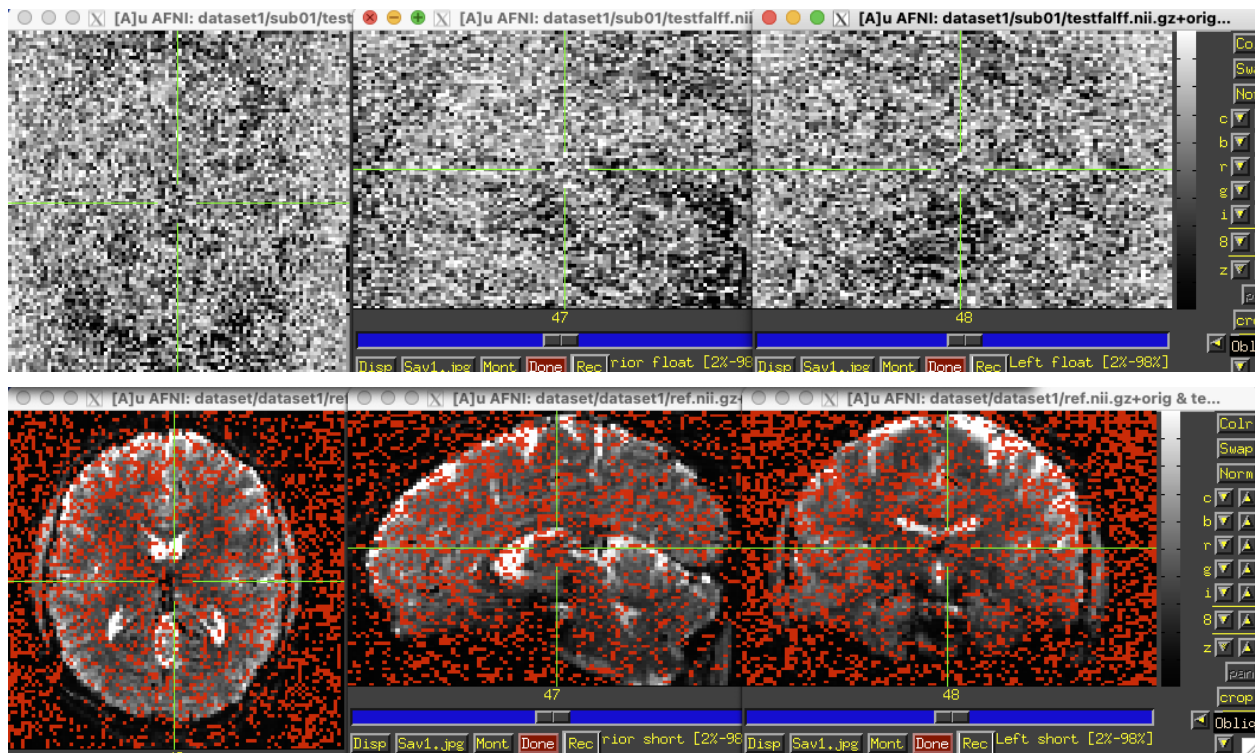
```

at.inputs.input_image = falff
at.inputs.reference_image = '../ref.nii.gz'
at.inputs.output_image = f'inref_{falff}.gz'
at.inputs.transforms = [matname]

# Execute the interface
result = at.run()

os.chdir('../..')

```



Somehow fALFF images are a bit weird.

Next we perform,

fALFF, RSFA and Global Signal Transformation

Stacking 3d to 4d Image for fALFF, RSFA and Global Signal Image

Statistical Analysis for DataSet-1

Generated T-Map

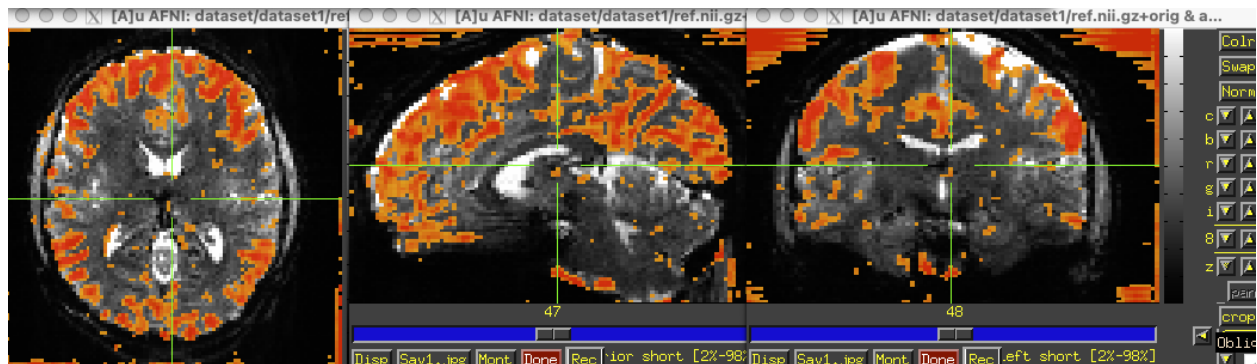
Standard diet minus 12-hour fast


```
3dttest++ -setA all_reho_std.nii.gz[0] all_reho_std.nii.gz[1].. -setB
all_reho_fast.nii.gz[0] all_reho_fast.nii.gz[1].. -paired -prefix
ttest_reho_std_minus_fast.nii.gz
```

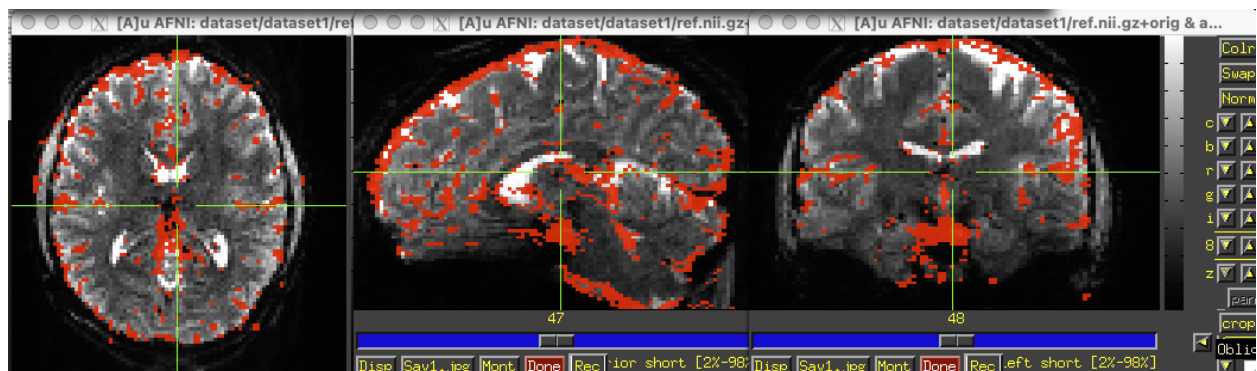
Similarly, create plots for various scenarios such as...

1. Standard diet minus ketogenic diet
2. 12-hour fast minus ketogenic diet
3. No bolus minus glucose drink
4. No bolus minus ketogenic drink
5. Glucose drink minus ketogenic drink

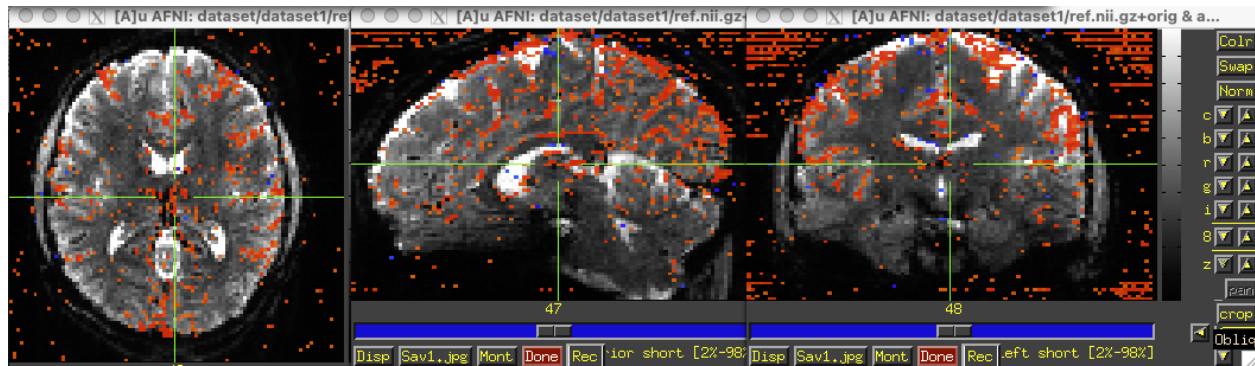
Cmd: afni all_reho_fast.nii.gz ref.nii.gz ref.nii.gz



Cmd: afni all_RFSA_fast.nii.gz ref.nii.gz ref.nii.gz



Cmd: afni all_global_signal_fast.nii.gz ref.nii.gz ref.nii.gz

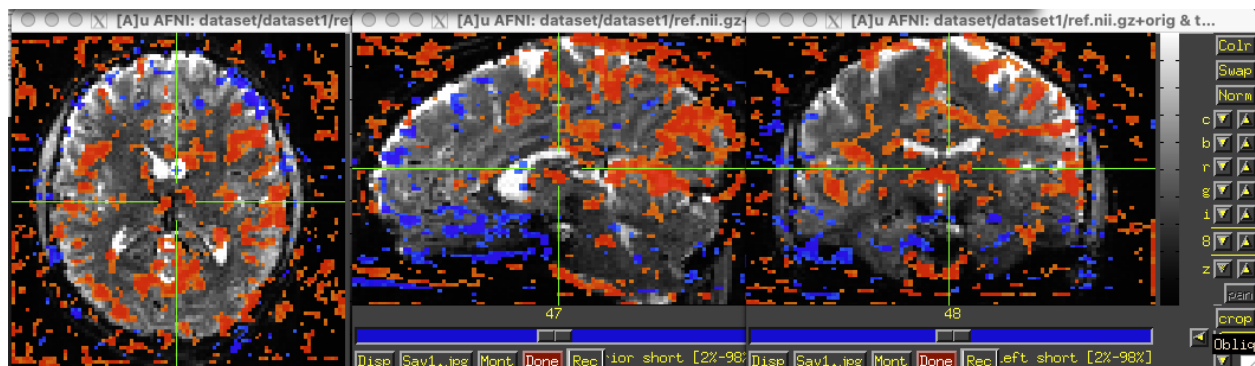


Generated Thresholded T-Map for DataSet-1:

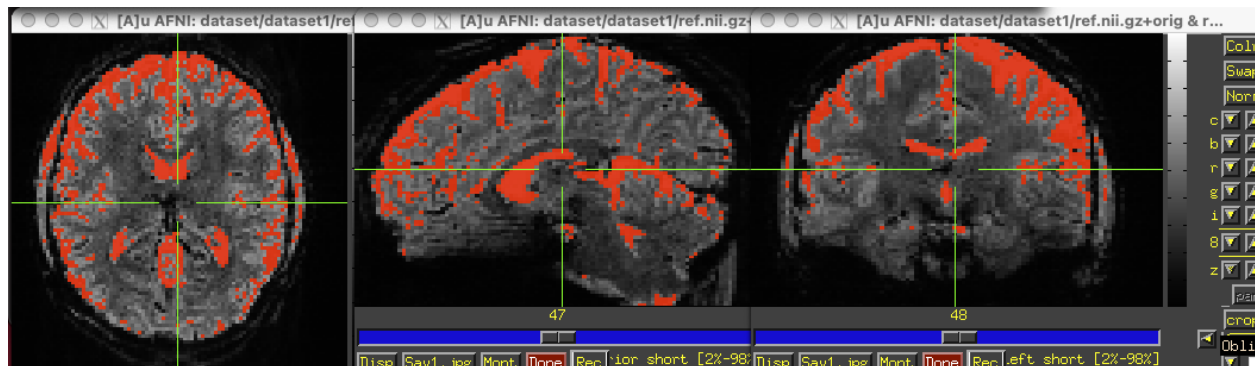
In this step, a threshold of 2 voxels is established, resulting in the separation of positive and negative values.

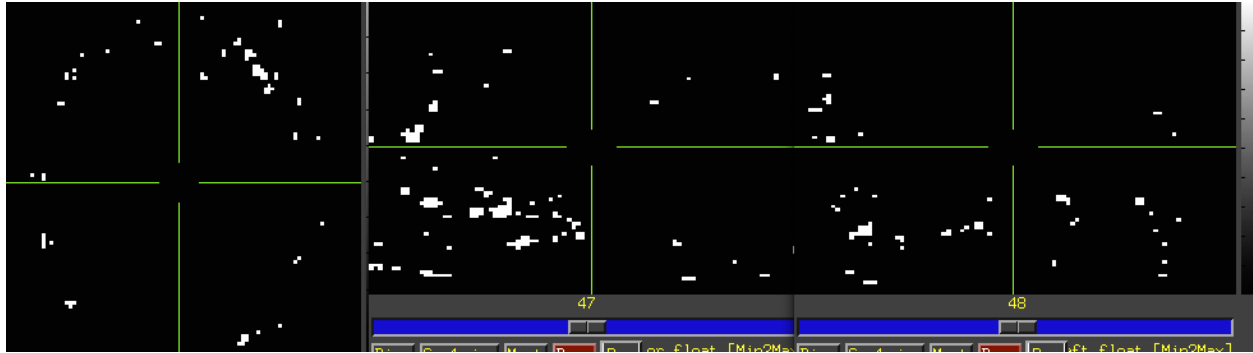
```
3dcalc -a ttest_reho_fast_minus_ket.nii.gz[1] -expr 'isnegative(a+2)'
-prefix negative_ttest_reho_fast_minus_ket_threshmap.nii.gz
```

Cmd: afni ttest_reho_fast_minus_ket.nii.gz ref.nii.gz



Cmd: afni negative_ttest_reho_fast_minus_ket_threshmap.nii.gz ref.nii.gz





Plot the Images:

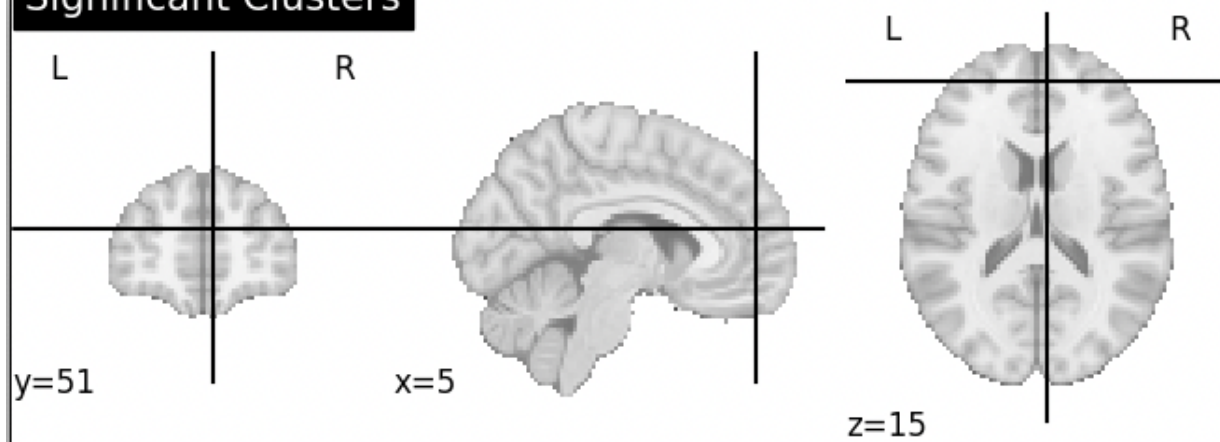
Images are visualized using the glass brain technique provided by Nilearn.

```
from nilearn import plotting

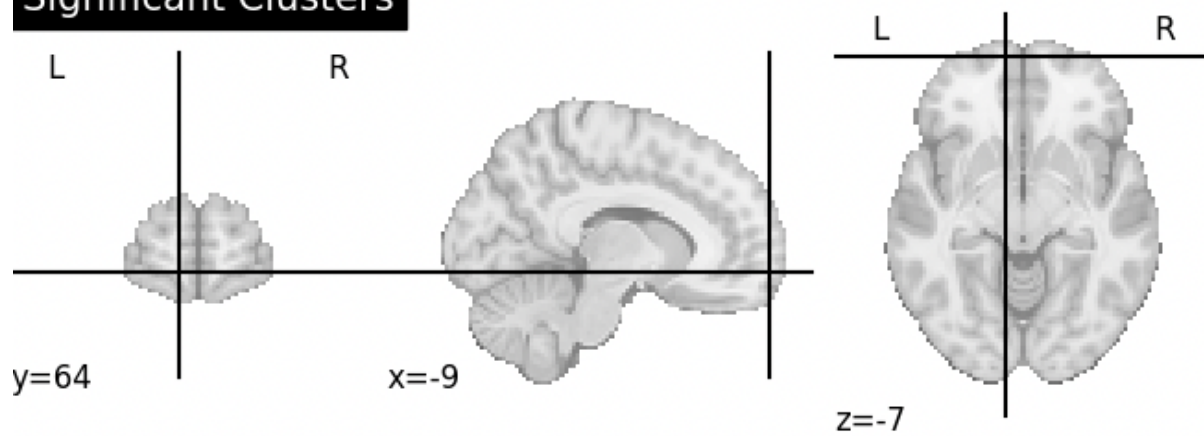
map_filenames =
["negative_ttest_RSFA_fast_minus_ket_threshmap.nii.gz", .... ,
"negative_ttest_reho_std_minus_ket_threshmap.nii.gz"]
for map_file in map_filenames:
    display = plotting.plot_stat_map(map_file, threshold=3,
colorbar=True, display_mode='ortho', title="Significant Clusters")
    plotting.show()
```

Significant Cluster Cluster for DataSet-1

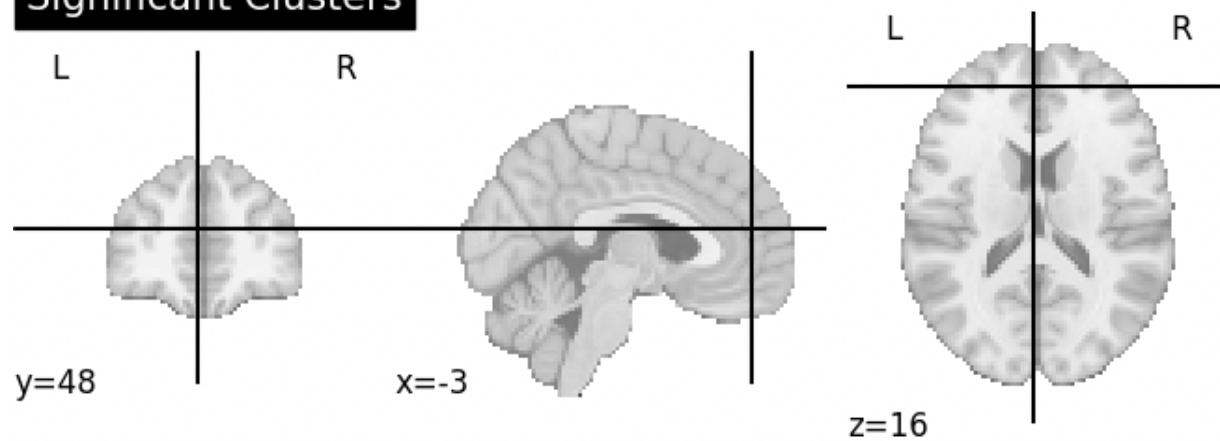
Significant Clusters



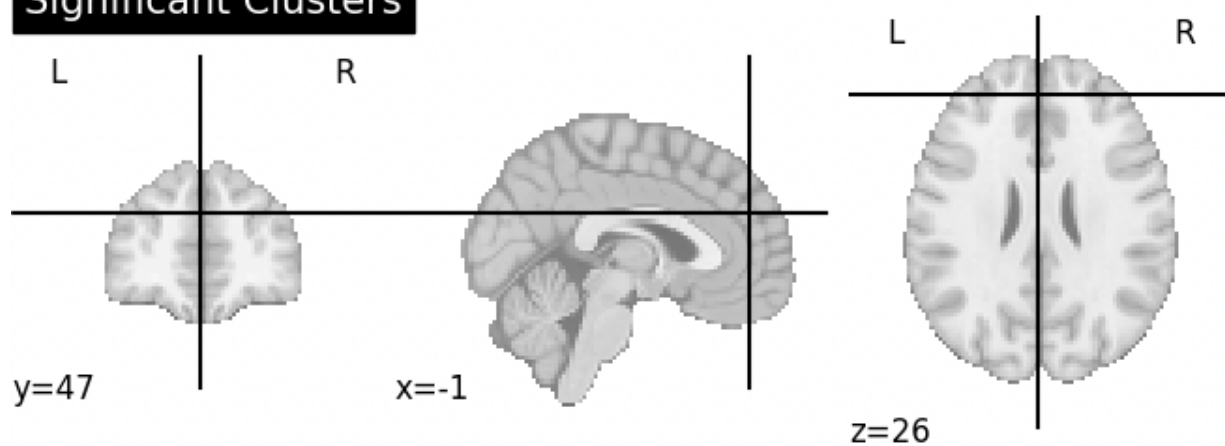
Significant Clusters



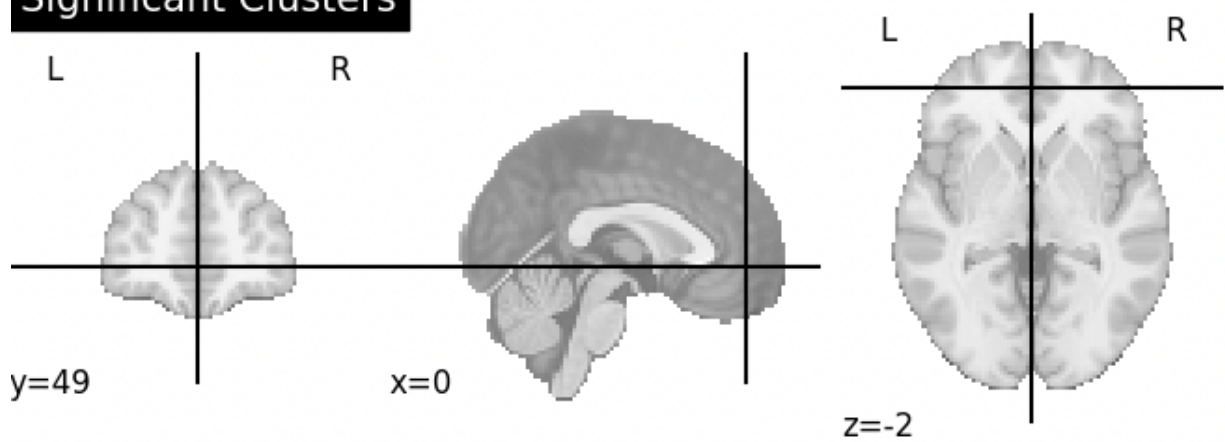
Significant Clusters



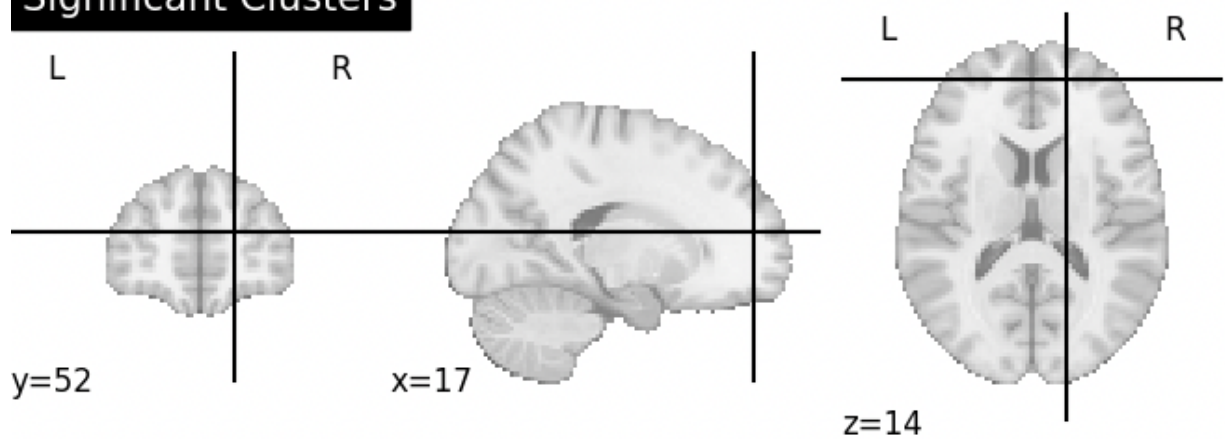
Significant Clusters



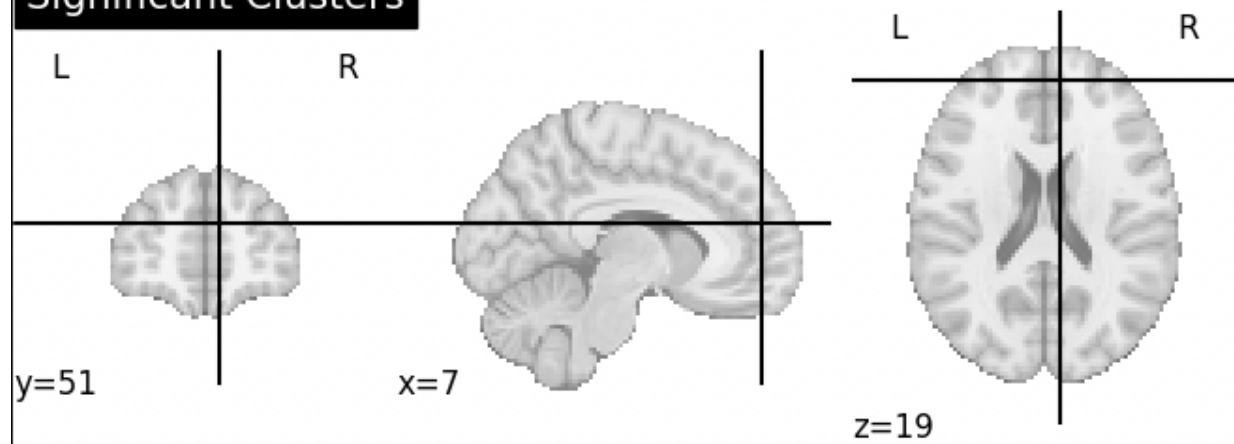
Significant Clusters



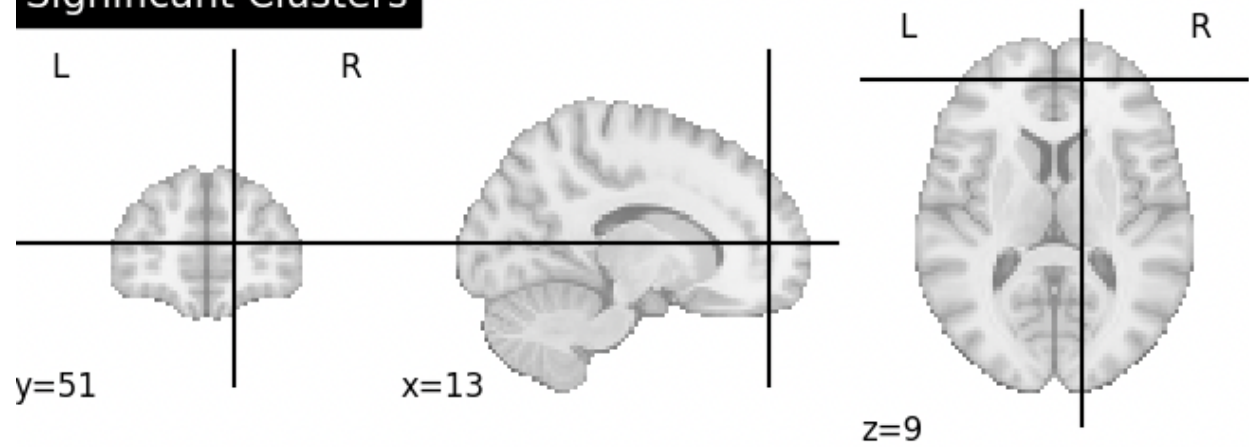
Significant Clusters



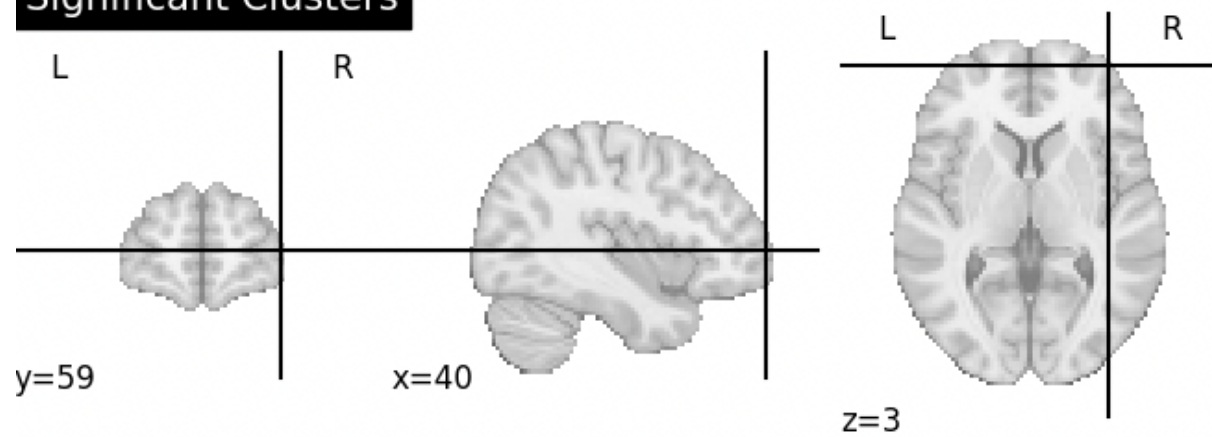
Significant Clusters



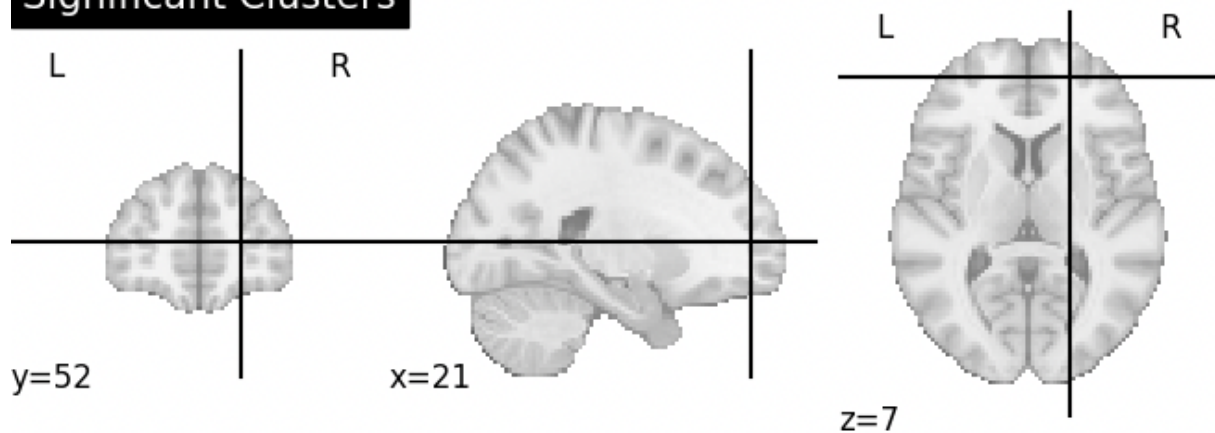
Significant Clusters



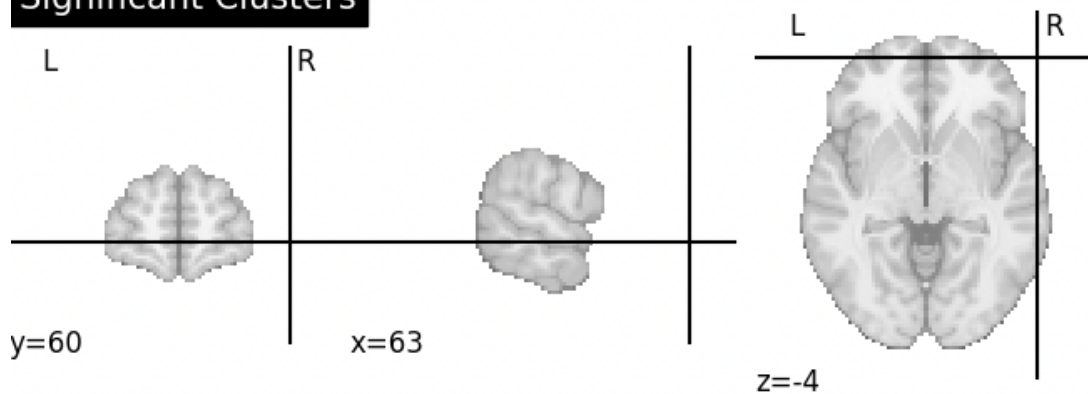
Significant Clusters



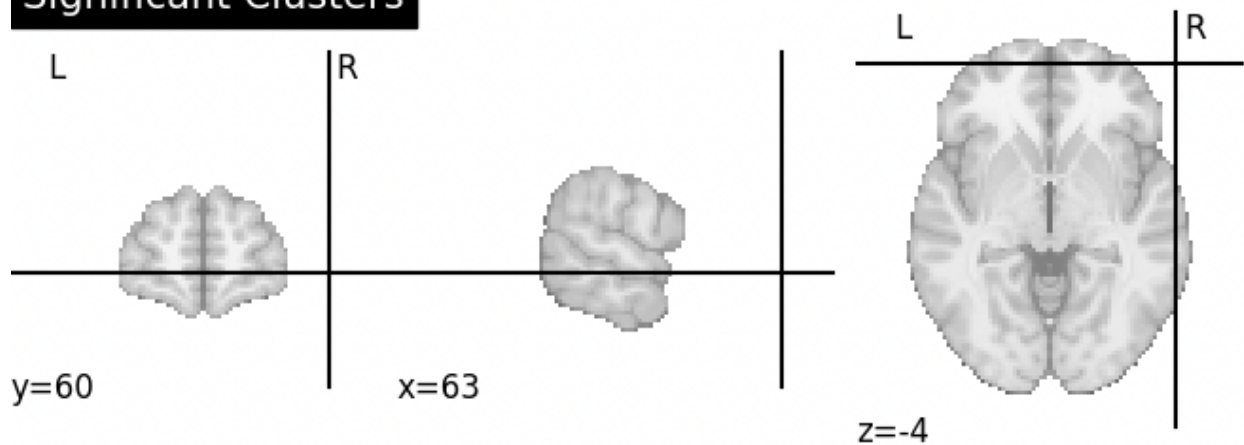
Significant Clusters



Significant Clusters



Significant Clusters



Statistical Analysis for DataSet-2

Generated T-Map

For following:

1. No bolus minus glucose drink
2. No bolus minus ketogenic drink
3. Glucose drink minus ketogenic drink

```
#!/bin/bash
#Reho
#No bolus minus glucose drink
3dttest++ -setA all_reho_glc1.nii[0] all_reho_glc1.nii[1]
all_reho_glc1.nii[2] all_reho_glc1.nii[3] all_reho_glc1.nii[4]
all_reho_glc1.nii[5] -setB all_reho_glc2.nii[0] all_reho_glc2.nii[1]
all_reho_glc2.nii[2] all_reho_glc2.nii[3] all_reho_glc2.nii[4]
all_reho_glc2.nii[5] -paired -prefix ttest_reho_case1.nii.gz

#No bolus minus ketogenic drink
3dttest++ -setA all_reho_bhb1.nii[0] all_reho_bhb1.nii[1]
all_reho_bhb1.nii[2] all_reho_bhb1.nii[3] all_reho_bhb1.nii[4]
all_reho_bhb1.nii[5] -setB all_reho_bhb2.nii[0] all_reho_bhb2.nii[1]
all_reho_bhb2.nii[2] all_reho_bhb2.nii[3] all_reho_bhb2.nii[4]
all_reho_bhb2.nii[5] -paired -prefix ttest_reho_case2.nii.gz

#Glucose drink minus ketogenic drink
3dttest++ -setA all_reho_glc2.nii[0] all_reho_glc2.nii[1]
all_reho_glc2.nii[2] all_reho_glc2.nii[3] all_reho_glc2.nii[4]
all_reho_glc2.nii[5] -setB all_reho_bhb2.nii[0] all_reho_bhb2.nii[1]
all_reho_bhb2.nii[2] all_reho_bhb2.nii[3] all_reho_bhb2.nii[4]
all_reho_bhb2.nii[5] -paired -prefix ttest_reho_case3.nii.gz

#fALFF
#No bolus minus glucose drink
3dttest++ -setA all_fALFF_glc1.nii.gz[0] all_fALFF_glc1.nii.gz[1]
all_fALFF_glc1.nii.gz[2] all_fALFF_glc1.nii.gz[3]
all_fALFF_glc1.nii.gz[4] all_fALFF_glc1.nii.gz[5] -setB
all_fALFF_glc2.nii.gz[0] all_fALFF_glc2.nii.gz[1]
all_fALFF_glc2.nii.gz[2] all_fALFF_glc2.nii.gz[3]
all_fALFF_glc2.nii.gz[4] all_fALFF_glc2.nii.gz[5] -paired -prefix
ttest_fALFF_case1.nii.gz

#No bolus minus ketogenic drink
3dttest++ -setA all_fALFF_bhb1.nii.gz[0] all_fALFF_bhb1.nii.gz[1]
all_fALFF_bhb1.nii.gz[2] all_fALFF_bhb1.nii.gz[3]
all_fALFF_bhb1.nii.gz[4] all_fALFF_bhb1.nii.gz[5] -setB
all_fALFF_bhb2.nii.gz[0] all_fALFF_bhb2.nii.gz[1]
```



```
all_fALFF_bhb2.nii.gz[2] all_fALFF_bhb2.nii.gz[3]
all_fALFF_bhb2.nii.gz[4] all_fALFF_bhb2.nii.gz[5] -paired -prefix
ttest_fALFF_case2.nii.gz
```

#Glucose drink minus ketogenic drink

```
3dttest++ -setA all_fALFF_glc2.nii.gz[0] all_fALFF_glc2.nii.gz[1]
all_fALFF_glc2.nii.gz[2] all_fALFF_glc2.nii.gz[3]
all_fALFF_glc2.nii.gz[4] all_fALFF_glc2.nii.gz[5] -setB
all_fALFF_bhb2.nii.gz[0] all_fALFF_bhb2.nii.gz[1]
all_fALFF_bhb2.nii.gz[2] all_fALFF_bhb2.nii.gz[3]
all_fALFF_bhb2.nii.gz[4] all_fALFF_bhb2.nii.gz[5] -paired -prefix
ttest_fALFF_case3.nii.gz
```

#RSFA

#No bolus minus glucose drink

```
3dttest++ -setA all_RSFA_glc1.nii.gz[0] all_RSFA_glc1.nii.gz[1]
all_RSFA_glc1.nii.gz[2] all_RSFA_glc1.nii.gz[3]
all_RSFA_glc1.nii.gz[4] all_RSFA_glc1.nii.gz[5] -setB
all_RSFA_glc2.nii.gz[0] all_RSFA_glc2.nii.gz[1]
all_RSFA_glc2.nii.gz[2] all_RSFA_glc2.nii.gz[3]
all_RSFA_glc2.nii.gz[4] all_RSFA_glc2.nii.gz[5] -paired -prefix
ttest_RSFA_case1.nii.gz
```

#No bolus minus ketogenic drink

```
3dttest++ -setA all_RSFA_bhb1.nii.gz[0] all_RSFA_bhb1.nii.gz[1]
all_RSFA_bhb1.nii.gz[2] all_RSFA_bhb1.nii.gz[3]
all_RSFA_bhb1.nii.gz[4] all_RSFA_bhb1.nii.gz[5] -setB
all_RSFA_bhb2.nii.gz[0] all_RSFA_bhb2.nii.gz[1]
all_RSFA_bhb2.nii.gz[2] all_RSFA_bhb2.nii.gz[3]
all_RSFA_bhb2.nii.gz[4] all_RSFA_bhb2.nii.gz[5] -paired -prefix
ttest_RSFA_case2.nii.gz
```

#Glucose drink minus ketogenic drink

```
3dttest++ -setA all_RSFA_glc2.nii.gz[0] all_RSFA_glc2.nii.gz[1]
all_RSFA_glc2.nii.gz[2] all_RSFA_glc2.nii.gz[3]
all_RSFA_glc2.nii.gz[4] all_RSFA_glc2.nii.gz[5] -setB
all_RSFA_bhb2.nii.gz[0] all_RSFA_bhb2.nii.gz[1]
all_RSFA_bhb2.nii.gz[2] all_RSFA_bhb2.nii.gz[3]
all_RSFA_bhb2.nii.gz[4] all_RSFA_bhb2.nii.gz[5] -paired -prefix
```

```
ttest_RSFA_case3.nii.gz
```

Generated Thresholded T-Map for DataSet-2:

In this step, a threshold of 2 voxels is established, resulting in the separation of positive and negative values.

```
#!/bin/bash
```

```
#ReHo Metric
```

```
3dcalc -a ttest_reho_case1.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_reho_case1_threshmap.nii.gz
```

```
3dcalc -a ttest_reho_case1.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_reho_case1_threshmap.nii.gz
```

```
3dcalc -a ttest_reho_case2.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_reho_case2_threshmap.nii.gz
```

```
3dcalc -a ttest_reho_case2.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_reho_case2_threshmap.nii.gz
```

```
3dcalc -a ttest_reho_case3.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_reho_case3_threshmap.nii.gz
```

```
3dcalc -a ttest_reho_case3.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_reho_case3_threshmap.nii.gz
```

```
#fALFF Metric
```

```
3dcalc -a ttest_fALFF_case1.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_fALFF_case1_threshmap.nii.gz
```

```
3dcalc -a ttest_fALFF_case1.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_fALFF_case1_threshmap.nii.gz
```

```
3dcalc -a ttest_fALFF_case2.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_fALFF_case2_threshmap.nii.gz
```

```
3dcalc -a ttest_fALFF_case2.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_fALFF_case2_threshmap.nii.gz
```

```
3dcalc -a ttest_fALFF_case3.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_fALFF_case3_threshmap.nii.gz
```

```
3dcalc -a ttest_fALFF_case3.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_fALFF_case3_threshmap.nii.gz
```

#RSFA Metric

```
3dcalc -a ttest_RSFA_case1.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_RSFA_case1_threshmap.nii.gz
```

```
3dcalc -a ttest_RSFA_case1.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_RSFA_case1_threshmap.nii.gz
```

```
3dcalc -a ttest_RSFA_case2.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_RSFA_case2_threshmap.nii.gz
```

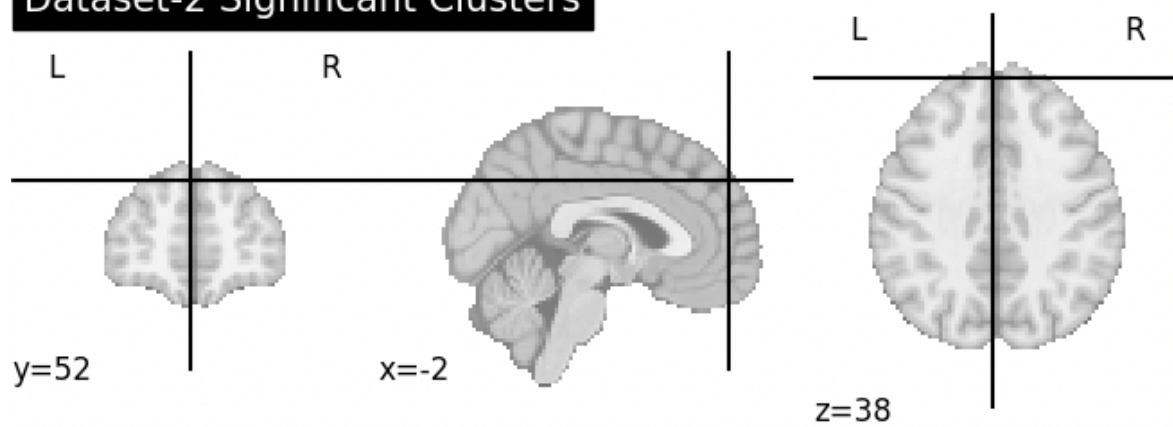
```
3dcalc -a ttest_RSFA_case2.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_RSFA_case2_threshmap.nii.gz
```

```
3dcalc -a ttest_RSFA_case3.nii.gz[1] -expr 'isnegative(a+2)' -prefix  
negative_ttest_RSFA_case3_threshmap.nii.gz
```

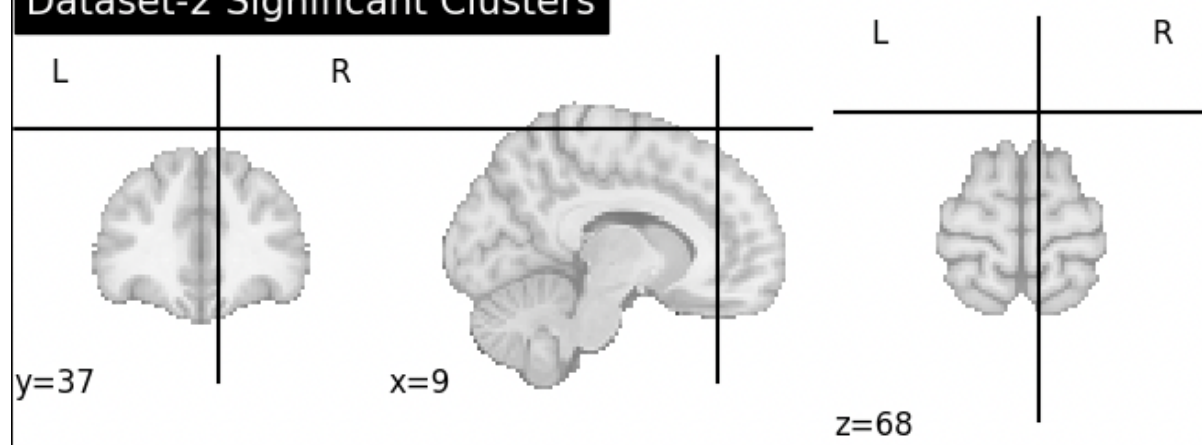
```
3dcalc -a ttest_RSFA_case3.nii.gz[1] -expr 'isnegative(a-2)' -prefix  
positive_ttest_RSFA_case3_threshmap.nii.gz
```

Significant Cluster Cluster for DataSet-2

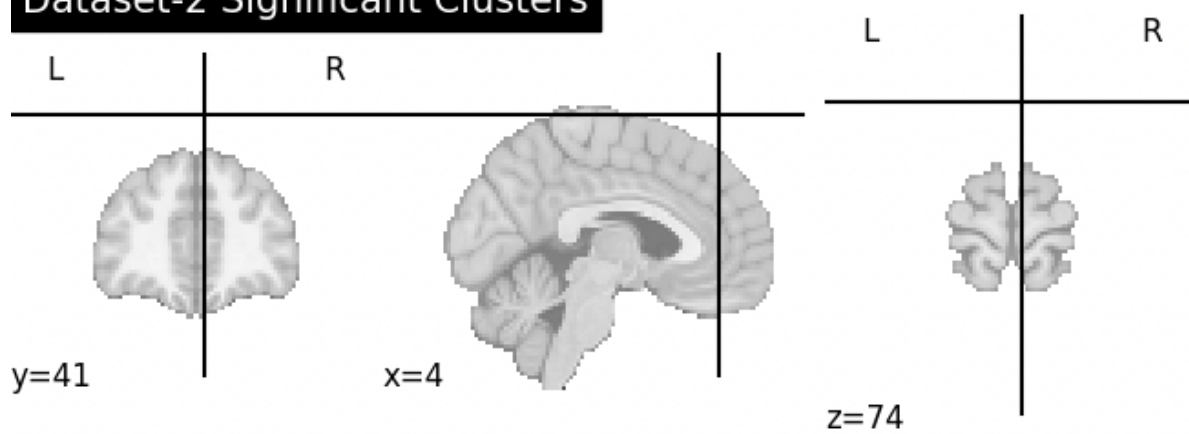
Dataset-2 Significant Clusters



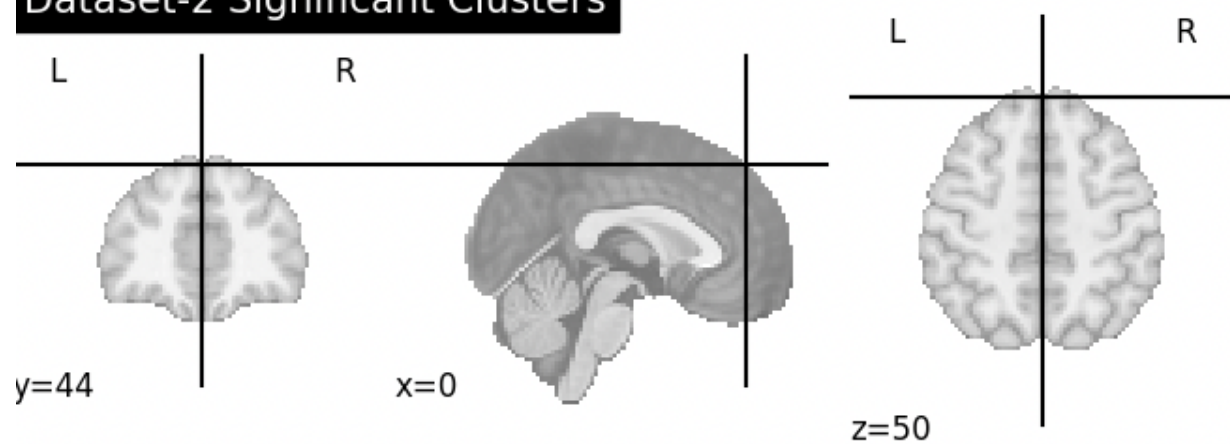
Dataset-2 Significant Clusters



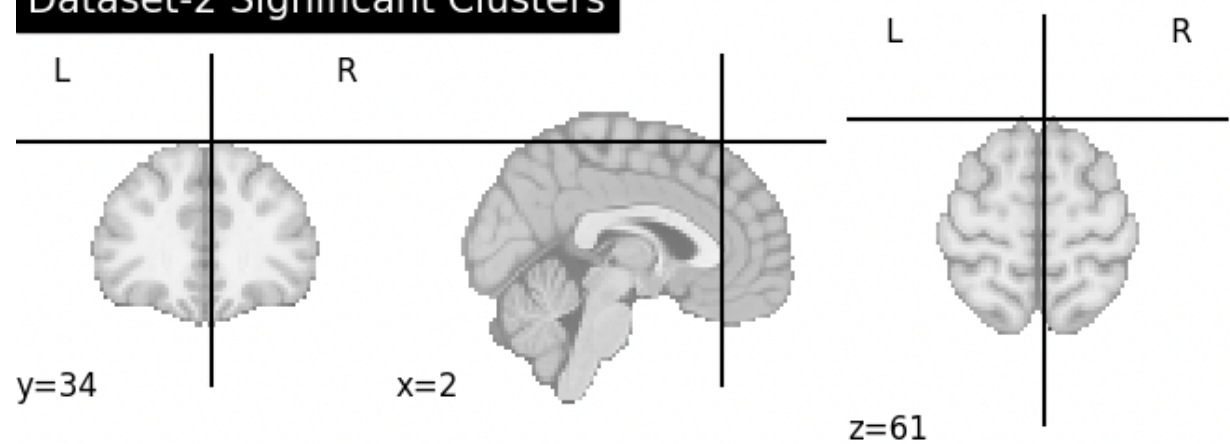
Dataset-2 Significant Clusters



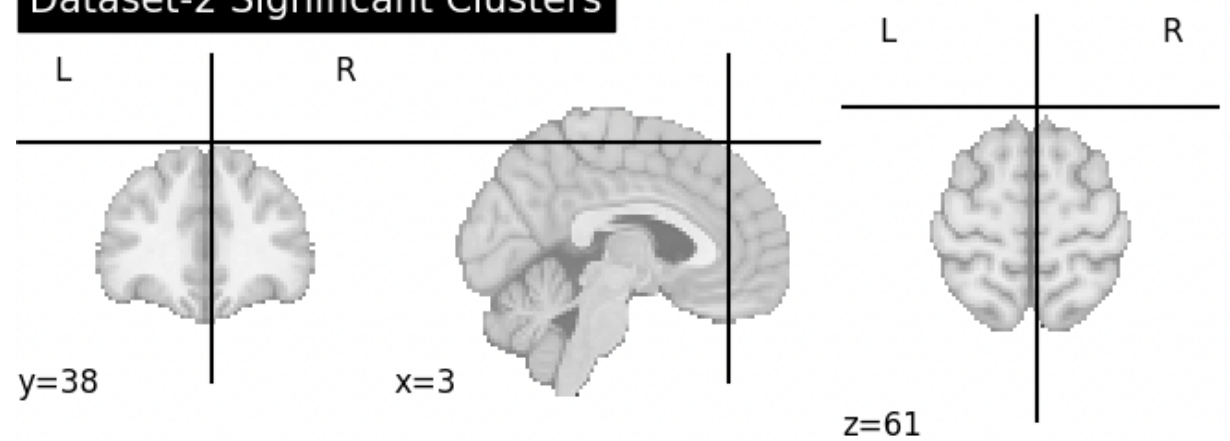
Dataset-2 Significant Clusters



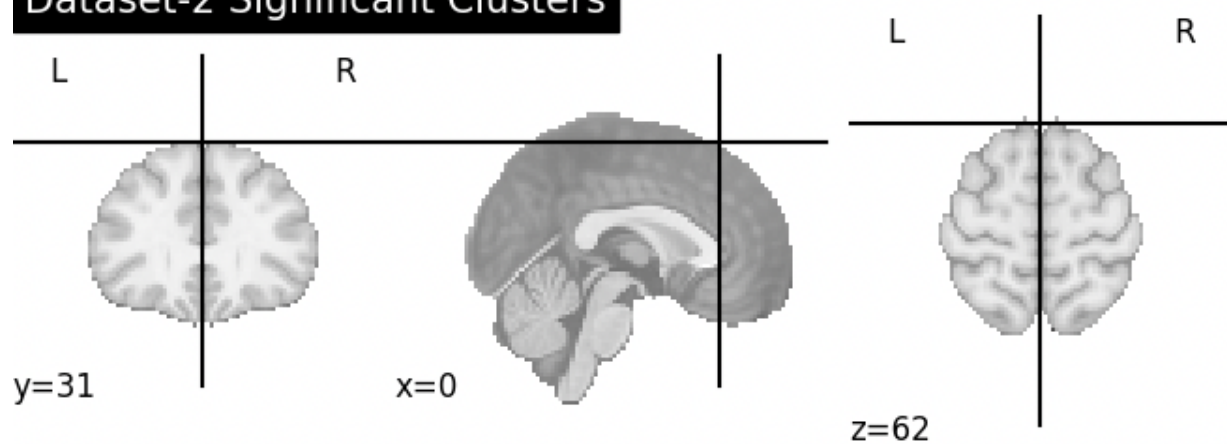
Dataset-2 Significant Clusters



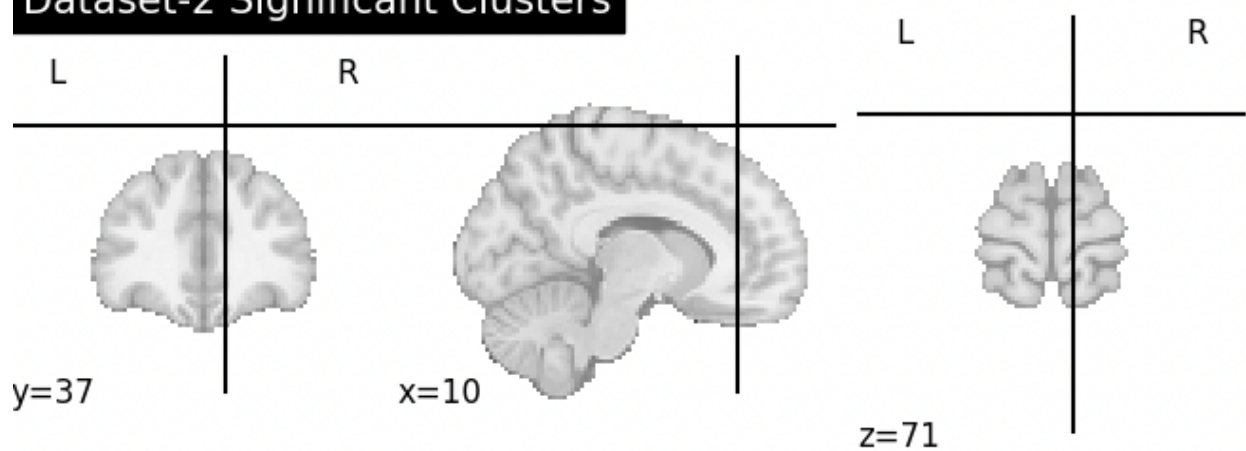
Dataset-2 Significant Clusters



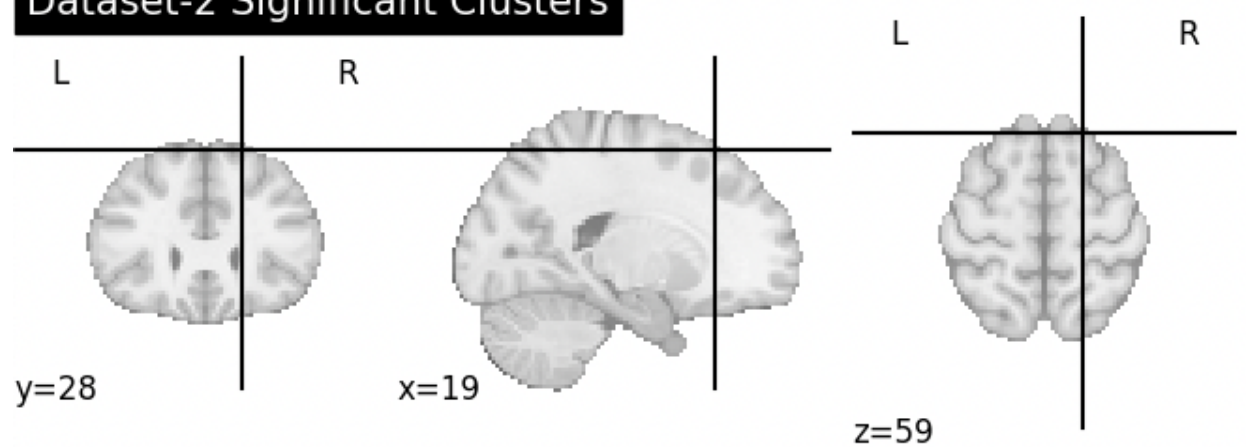
Dataset-2 Significant Clusters



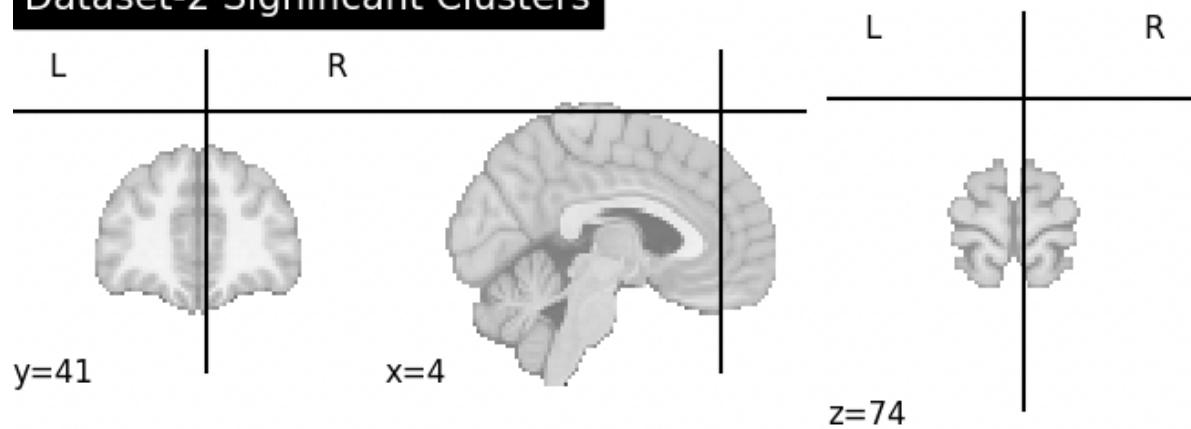
Dataset-2 Significant Clusters



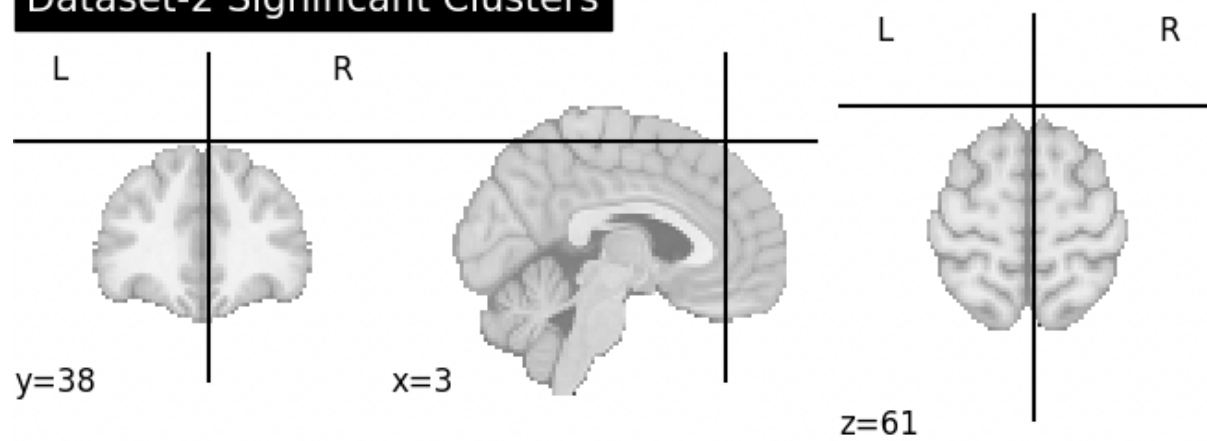
Dataset-2 Significant Clusters



Dataset-2 Significant Clusters



Dataset-2 Significant Clusters



Dataset-2 Significant Clusters

