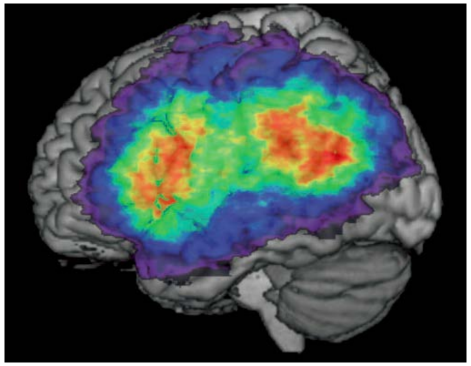
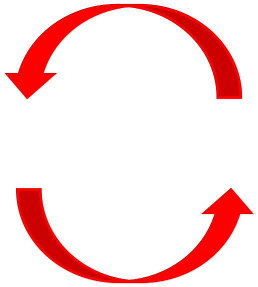
MRRI
MOSS REHABILITATION RESEARCH INSTITUTE

Lesion-Symptom Mapping Workshop

Speakers:

Frank Garcea

Harrison Stoll

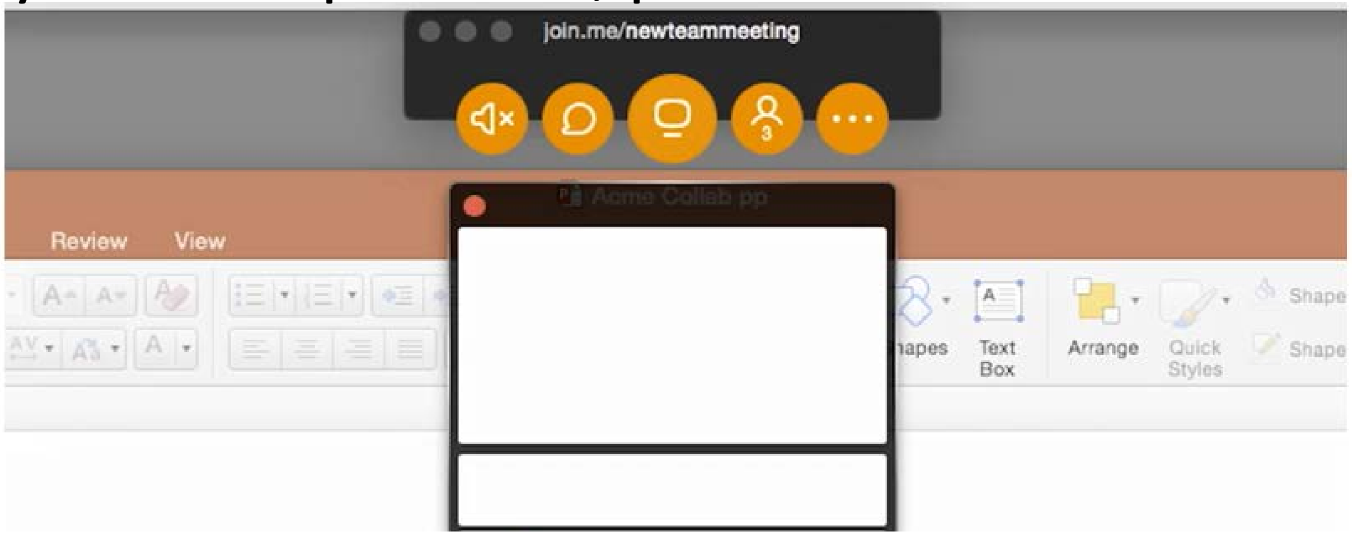
Austin Wild

Organizer:

Aaron Wong

Join the discussion

- For those joining us remotely, we have muted incoming audio to reduce background noise
- If you have questions, please use the chat window



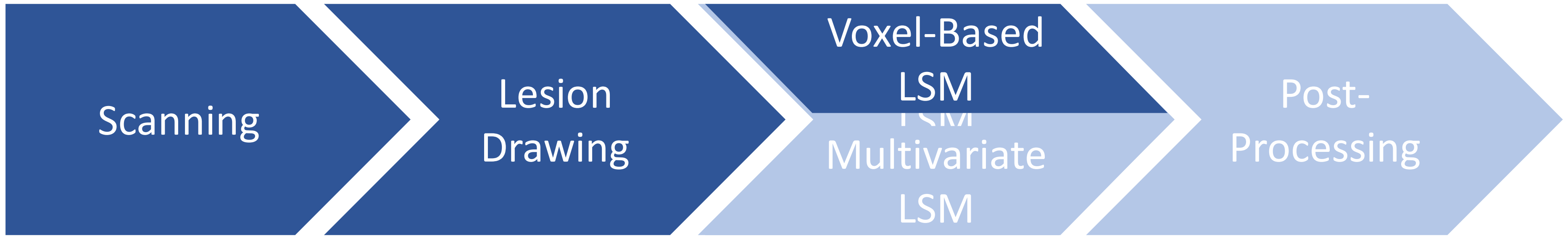
Join the discussion

- For those joining us remotely, we have muted incoming audio to reduce background noise
- If you have questions, please use the chat window
- We will also take questions after the session via email or twitter

✉ wongaaro@einstein.edu
@WongAaronL

✉ garceafr@einstein.edu
@frankgarcea

Lesion-Symptom Mapping Pipeline



Software packages:

- **SVR-LSM GUI (Matlab): https://github.com/atdemarco/svrlsmgui**
- SVR-LSM (R): https://rdrr.io/github/dorianps/LESYMAP/man/lsm_svr.html

To follow along:

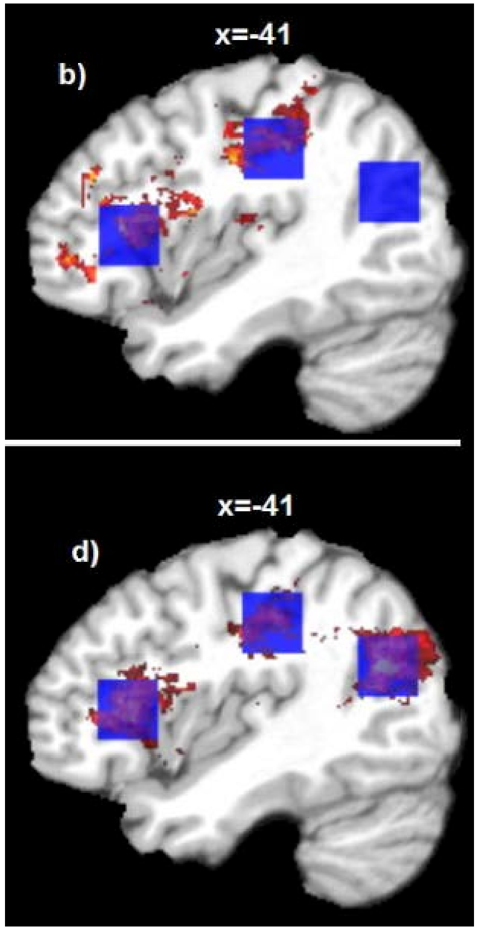
- MRIcron: https://www.nitrc.org/frs/?group_id=152
- Materials: https://mrri.org/lesion-symptom-mapping-workshop-series/

VLSM: A review and critique

- Compare behavioral scores of those with lesions and those without
  - Compute test statistic (e.g., t-value) to derive a p-value
- Major limitations:
  - Assuming each voxel is independent of one another
    - Voxels are typically correlated with one another (e.g., vasculature)
  - Running thousands of tests and increasing potential for false positives
    - Correction techniques might be removing some of the true signal
  - Different voxels have different lesion to no lesion ratios
    - Those with more lesions have higher power, biasing results

Multivariate LSM: Overview

- Consider multiple voxels at once when determining relationship between lesion location and behavior
  - Reduce # of tests
  - Consider relationship between voxels when computing significance
  - No longer have to worry about uneven distribution
- Approaches:
  - Support Vector Regression (SVR-VLSM)
  - Sparse Canonical Correlation Analysis for Neuroimaging (SCCAN)

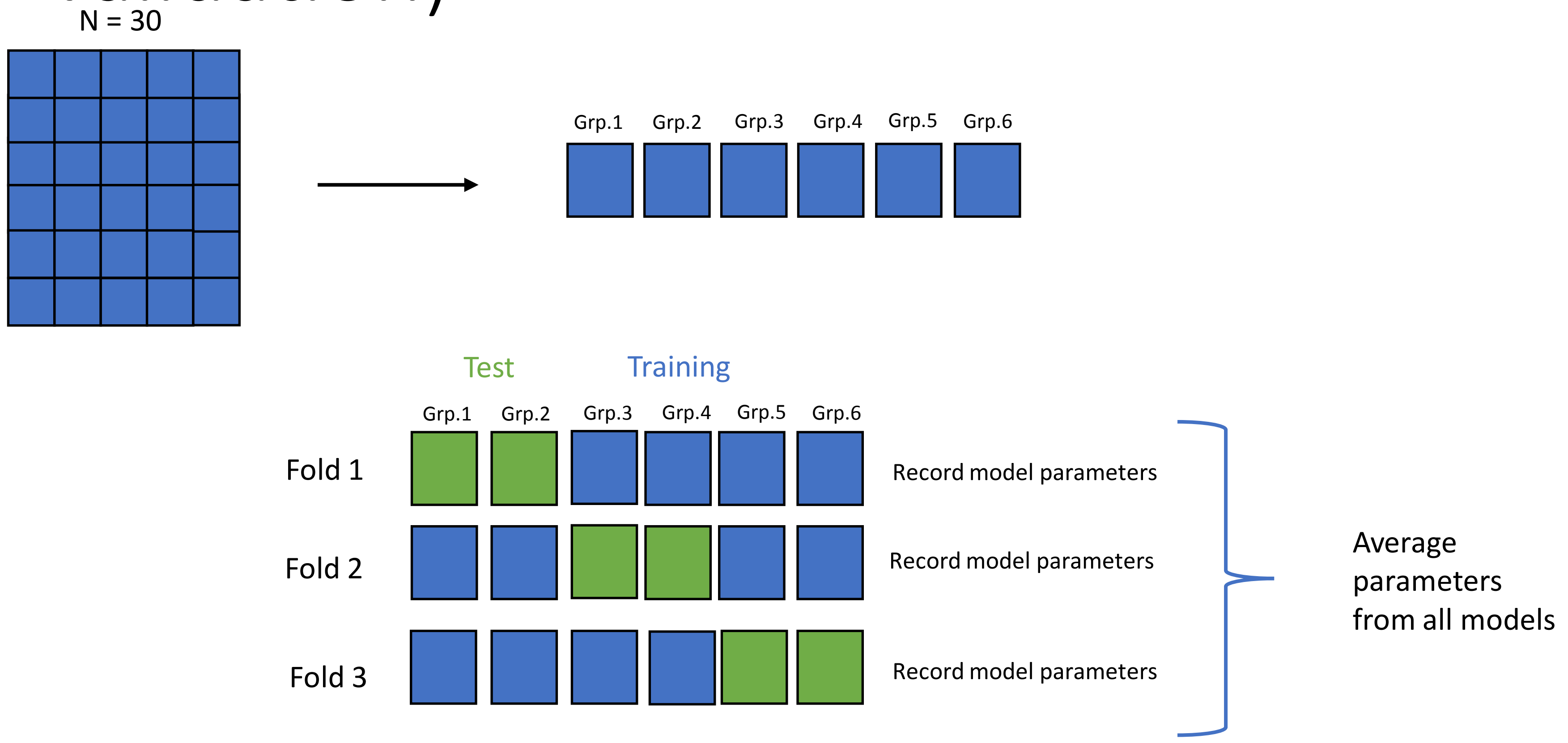


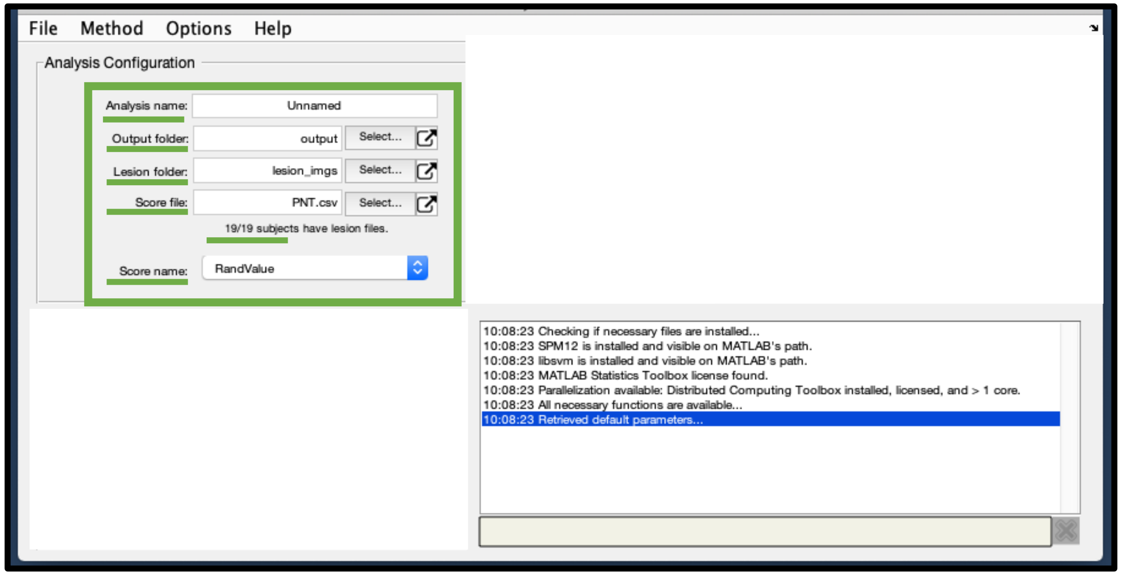
(Zhang et al. 2014)

SVR-LSM: Conceptual overview

- Regression equation with voxels as IV and behavior as DV
  - Output is a beta value for each voxel
- Machine-learning based approach to help solve model
  - High amount of collinearity between neighboring voxels
  - High number of IVs relative to DVs
- Need to constrain model (hyperparameters) and transform it to a nonlinear space
  - **Critical point: our beta values are nonlinear**

Model Validation (*k*-fold cross-validation)





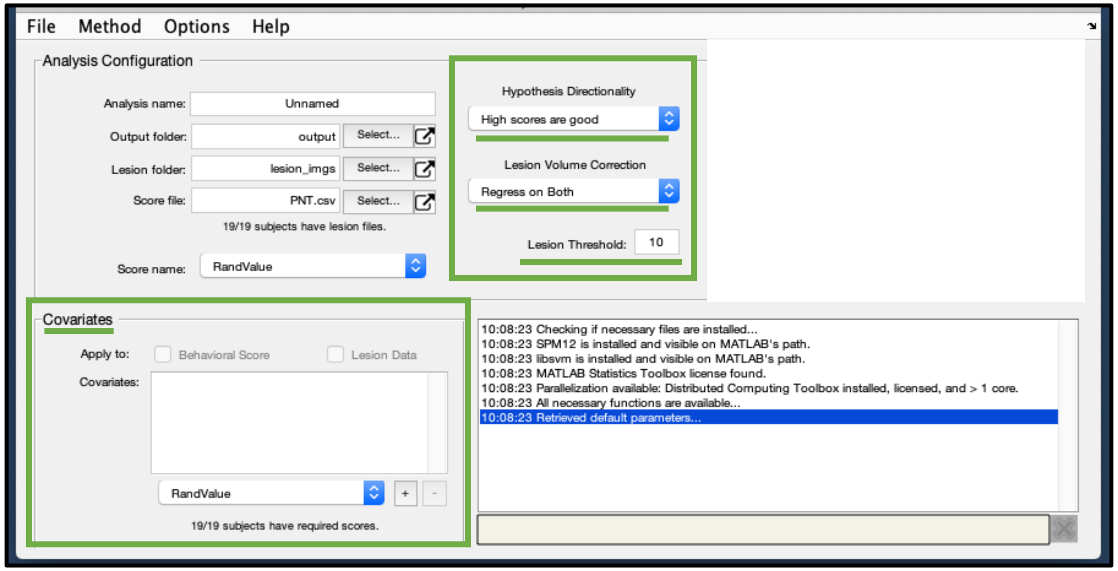
Analysis name: Unnamed
Output folder: output
Lesion folder: lesion_imgs
Score file: PNT.csv
19/19 subjects have lesion files.
Score name: RandValue

```
10:08:23 Checking if necessary files are installed...
10:08:23 SPM12 is installed and visible on MATLAB's path.
10:08:23 libsvm is installed and visible on MATLAB's path.
10:08:23 MATLAB Statistics Toolbox license found.
10:08:23 Parallelization available: Distributed Computing Toolbox installed, licensed, and > 1 core.
10:08:23 All necessary functions are available...
10:08:23 Retrieved default parameters...
```

(DeMarco et al. 2018)

SVR-LSM: Factors to consider

- Interested in high or low values
- Controlling for Total Lesion Volume (TLV)
  - DTLVC: Take value of voxel and divide by square root of TLV
  - Regress on Lesion: Regress TLV out of voxels (use residuals in model)
  - Regress on Behavior: Regress TLV out of behavior (use residuals in model)
  - Regress on both: Regress TLV out of both behavior and voxels (use residuals in model)
- Lesion Threshold value
- Other covariates



Analysis name: Unnamed
Output folder: output
Lesion folder: lesion_imgs
Score file: PNT.csv
19/19 subjects have lesion files.
Score name: RandValue

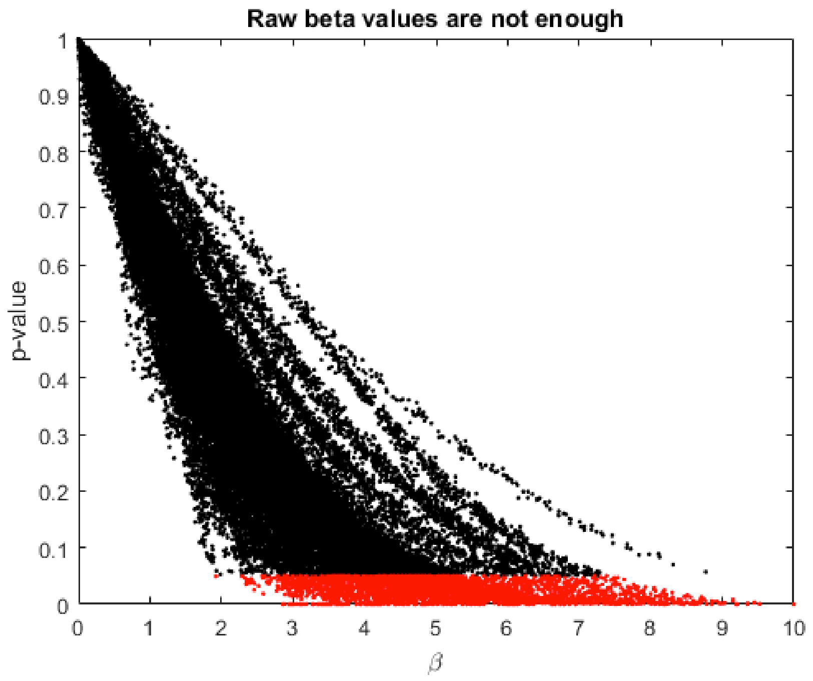
Hypothesis Directionality: High scores are good
Lesion Volume Correction: Regress on Both
Lesion Threshold: 10

Apply to: Behavioral Score Lesion Data
Covariates:
RandValue
19/19 subjects have required scores.

```
10:08:23 Checking if necessary files are installed...
10:08:23 SPM12 is installed and visible on MATLAB's path.
10:08:23 libsvm is installed and visible on MATLAB's path.
10:08:23 MATLAB Statistics Toolbox license found.
10:08:23 Parallelization available: Distributed Computing Toolbox installed, licensed, and > 1 core.
10:08:23 All necessary functions are available...
10:08:23 Retrieved default parameters...
```

SVR-LSM: Conceptual overview

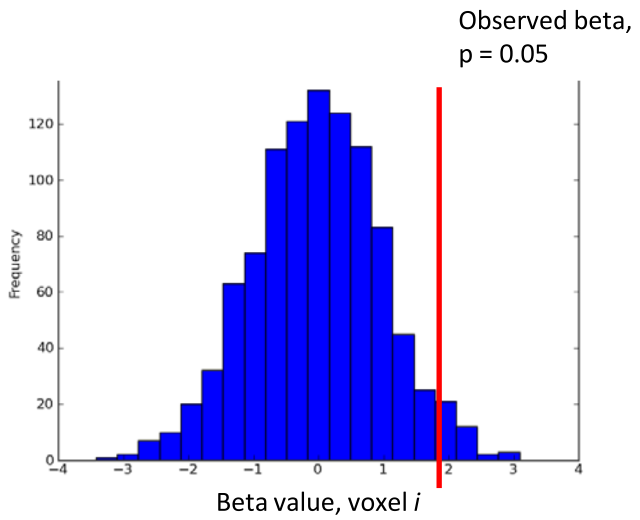
- How do we determine significance?

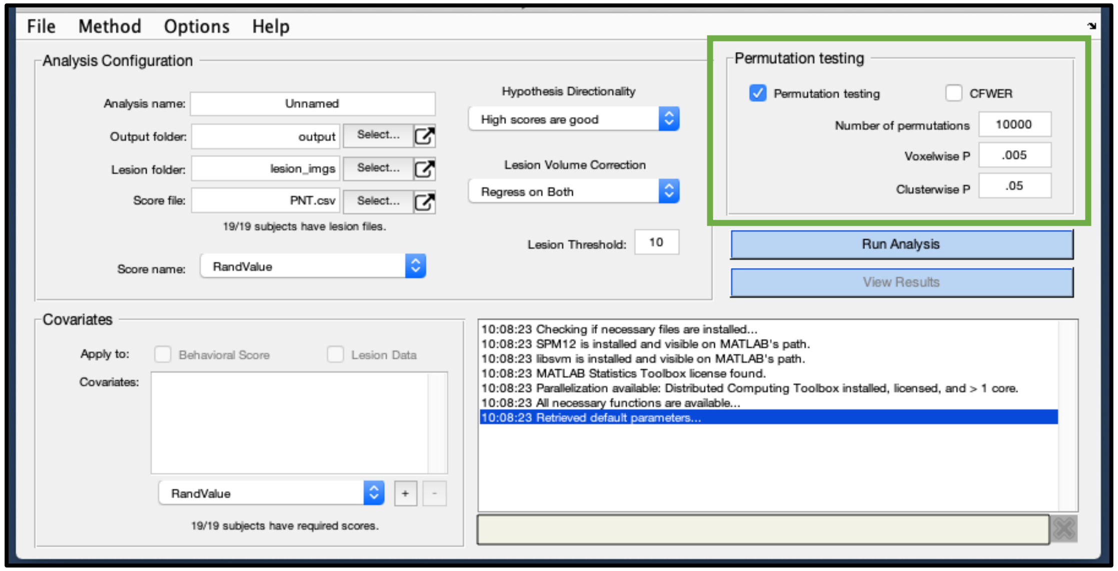


(Mirman 2018)

SVR-LSM: Conceptual overview

- How do we determine significance?
  - Run X number of permutations (10k standard)
  - Build null distribution
  - Test if a given voxel's beta value is significantly different from null distribution
  - Creates a problem of many tests





Analysis Configuration

Analysis name: Unnamed

Output folder: output

Lesion folder: lesion_imgs

Score file: PNT.csv

19/19 subjects have lesion files.

Score name: RandValue

Hypothesis Directionality: High scores are good

Lesion Volume Correction: Regress on Both

Lesion Threshold: 10

Permutation testing

[✓] Permutation testing

[ ] CFWER

Number of permutations: 10000

Voxelwise P: .005

Clusterwise P: .05

Covariates

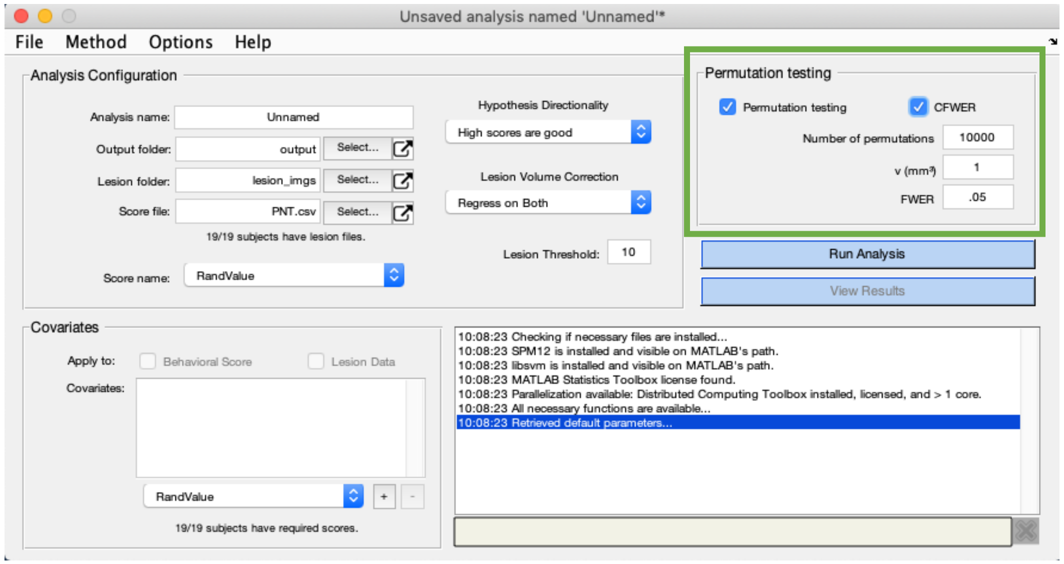
Apply to: [ ] Behavioral Score [ ] Lesion Data

Covariates:

RandValue

19/19 subjects have required scores.

```
10:08:23 Checking if necessary files are installed...
10:08:23 SPM12 is installed and visible on MATLAB's path.
10:08:23 libsvm is installed and visible on MATLAB's path.
10:08:23 MATLAB Statistics Toolbox license found.
10:08:23 Parallelization available: Distributed Computing Toolbox installed, licensed, and > 1 core.
10:08:23 All necessary functions are available...
10:08:23 Retrieved default parameters...
```



Unsaved analysis named 'Unnamed'*

Analysis Configuration

Analysis name: Unnamed

Output folder: output

Lesion folder: lesion_imgs

Score file: PNT.csv

19/19 subjects have lesion files.

Score name: RandValue

Hypothesis Directionality: High scores are good

Lesion Volume Correction: Regress on Both

Lesion Threshold: 10

Permutation testing

[✓] Permutation testing [✓] CFWER

Number of permutations: 10000

v (mm³): 1

FWER: .05

Covariates

Apply to: [ ] Behavioral Score [ ] Lesion Data

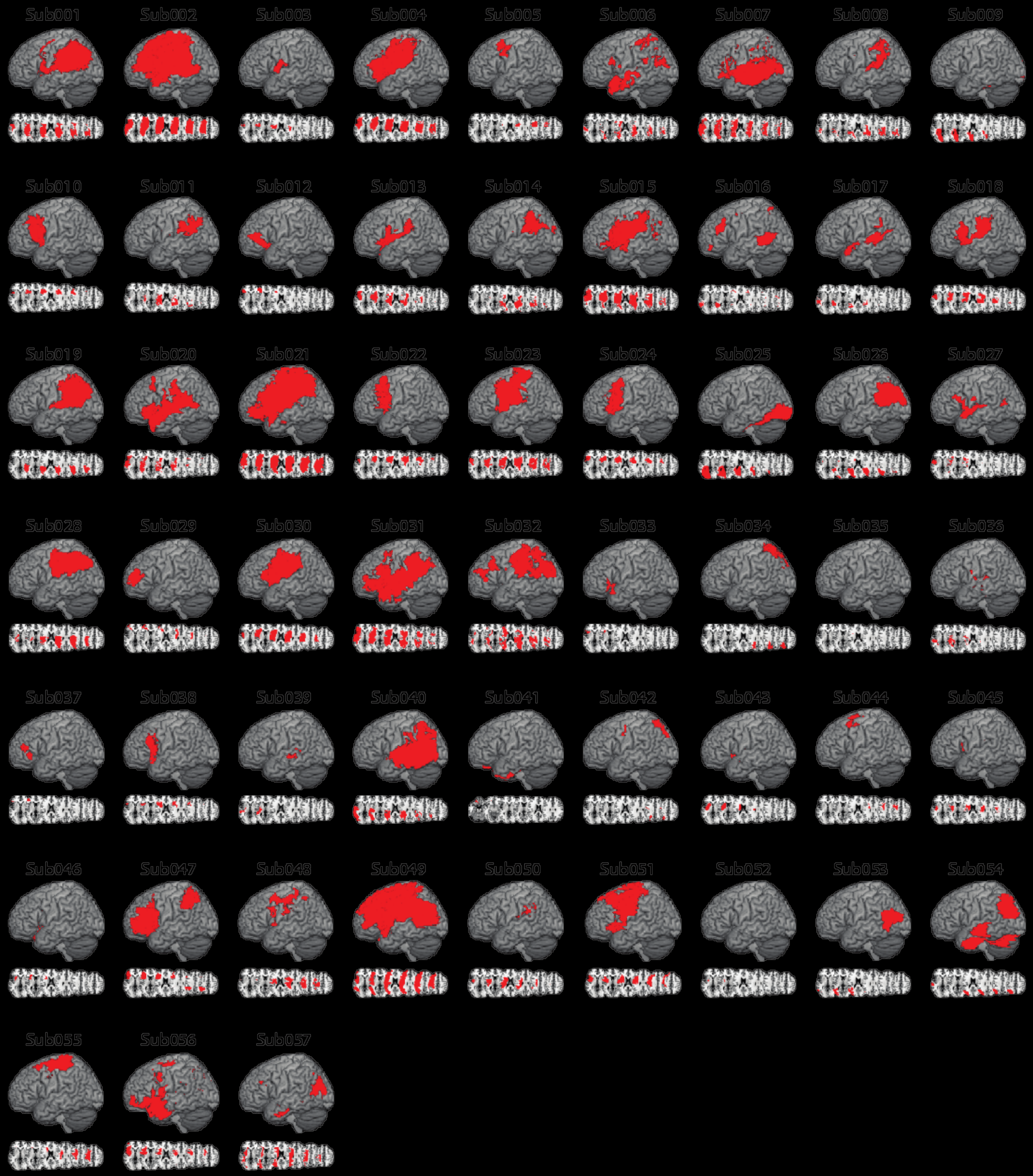
Covariates:

RandValue

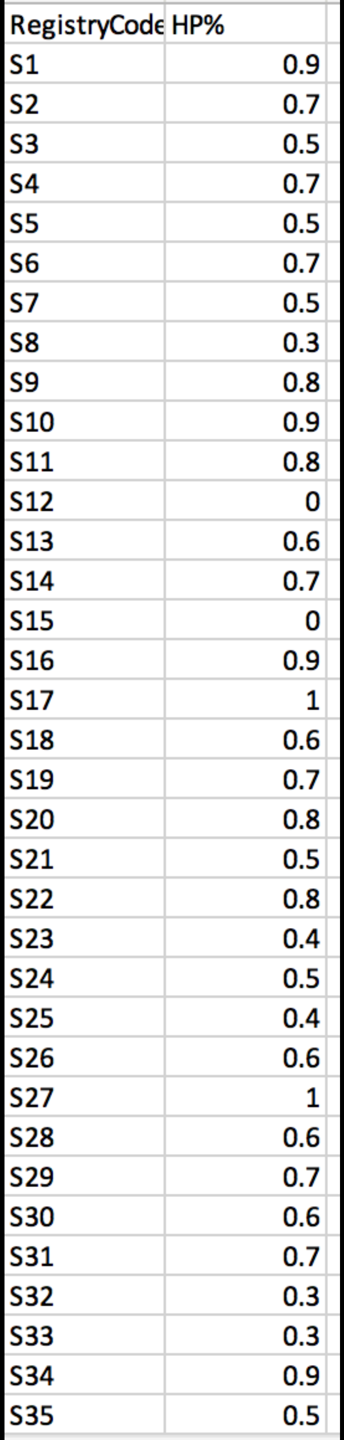
19/19 subjects have required scores.

```
10:08:23 Checking if necessary files are installed...
10:08:23 SPM12 is installed and visible on MATLAB's path.
10:08:23 libsvm is installed and visible on MATLAB's path.
10:08:23 MATLAB Statistics Toolbox license found.
10:08:23 Parallelization available: Distributed Computing Toolbox installed, licensed, and > 1 core.
10:08:23 All necessary functions are available...
10:08:23 Retrieved default parameters...
```

Figure 1. Distribution of Cortical and Subcortical Damage in each LCVA Participant.



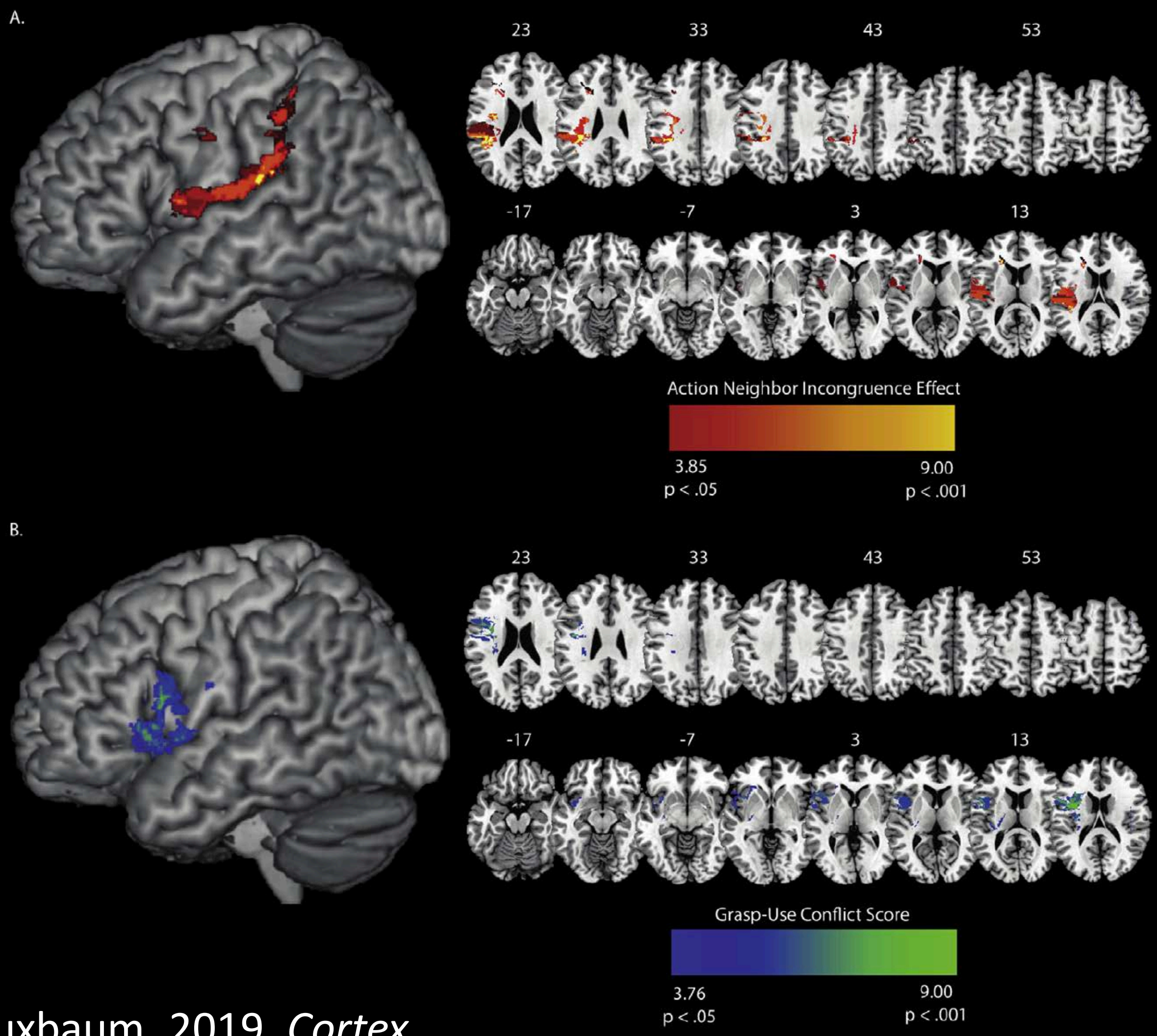
?



RegistryCode	HP%
S1	0.9
S2	0.7
S3	0.5
S4	0.7
S5	0.5
S6	0.7
S7	0.5
S8	0.3
S9	0.8
S10	0.9
S11	0.8
S12	0
S13	0.6
S14	0.7
S15	0
S16	0.9
S17	1
S18	0.6
S19	0.7
S20	0.8
S21	0.5
S22	0.8
S23	0.4
S24	0.5
S25	0.4
S26	0.6
S27	1
S28	0.6
S29	0.7
S30	0.6
S31	0.7
S32	0.3
S33	0.3
S34	0.9
S35	0.5

Garcea et al., in prep.

Post-analysis Processing



Garcea, Stoll, & Buxbaum, 2019, *Cortex*

Post-analysis Processing

1. Multiple comparison correction.
    1. SVR-LSM toolbox
    2. Custom matlab scripts

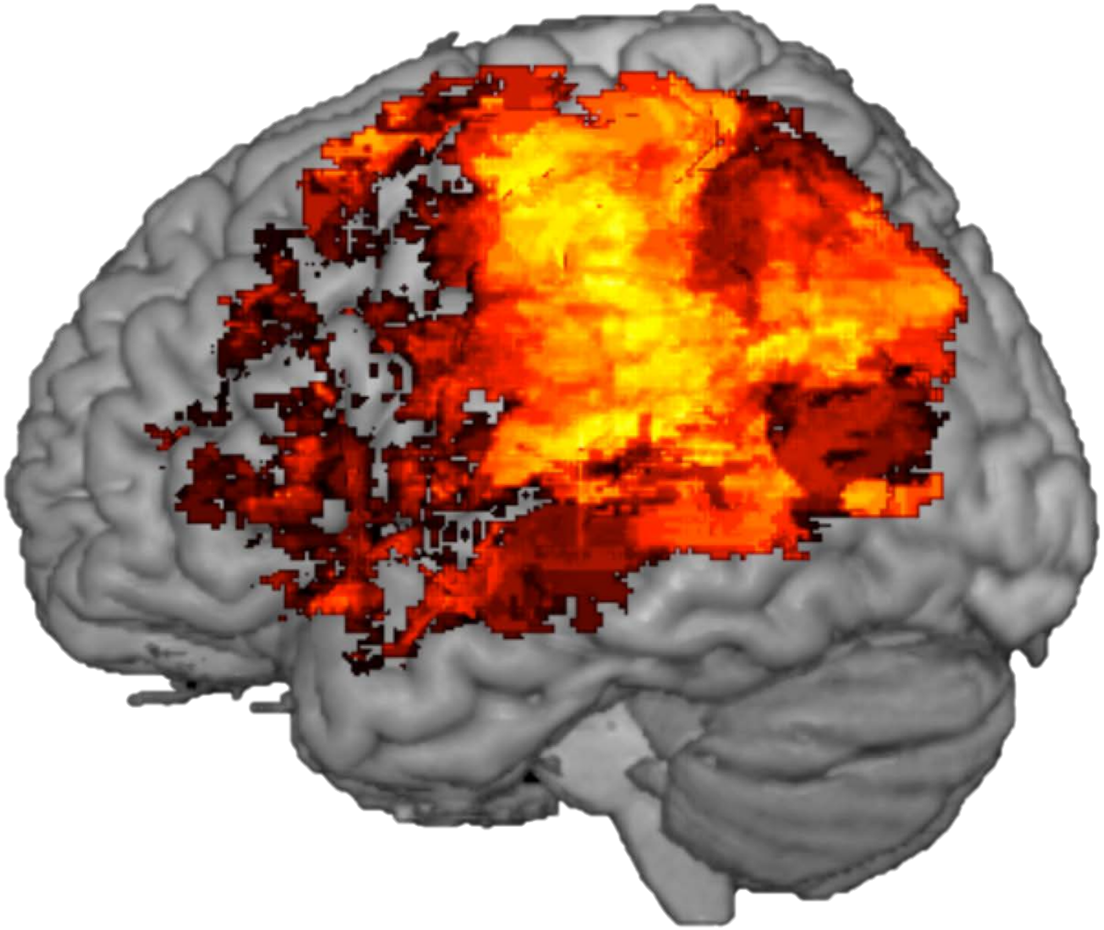
2. Cluster-size thresholding the voxelwise z-map.

3. Viewing results (and differences) before and after cluster correction.

4. Independently label your peaks and clusters (AAL template)

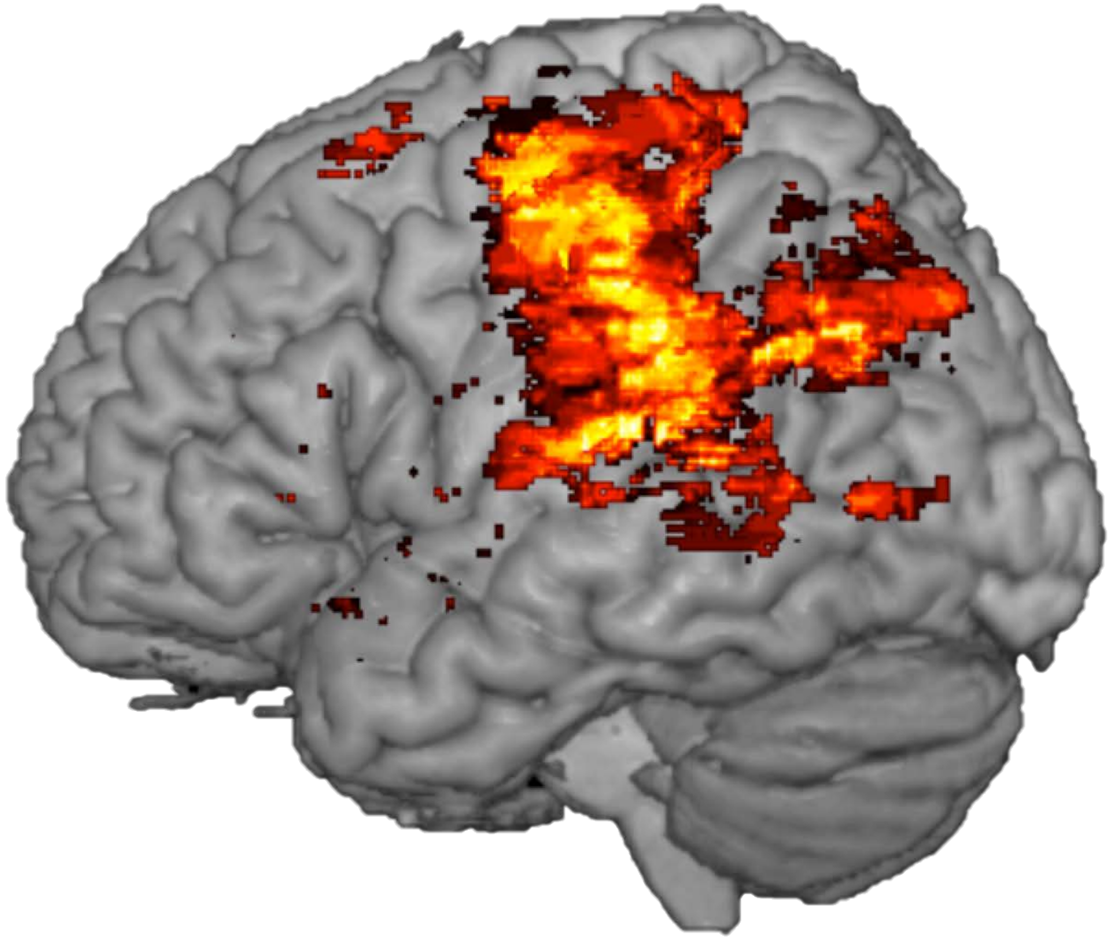
SVR-LSM: Multiple comparisons corrections

1. Raw beta values as the first output in SVR-LSM analysis.
   - Uninterpretable by itself.


Gesturing Tool Use, raw beta values (min, 0; max, 10)

SVR-LSM: Multiple comparisons corrections

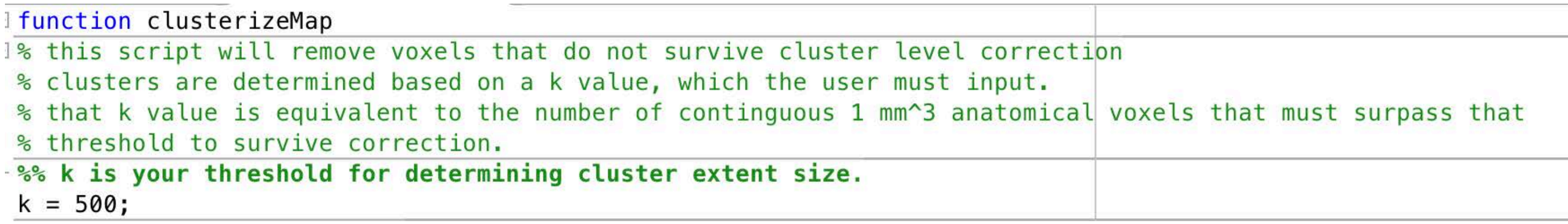
1. Raw beta values as the first output in SVR-LSM analysis.
   - Uninterpretable by itself.
2. Voxel-level correction (FWER or FDR).
   - Permutation testing to obtain null distribution in each voxel; then z-score your true data relative to null data.



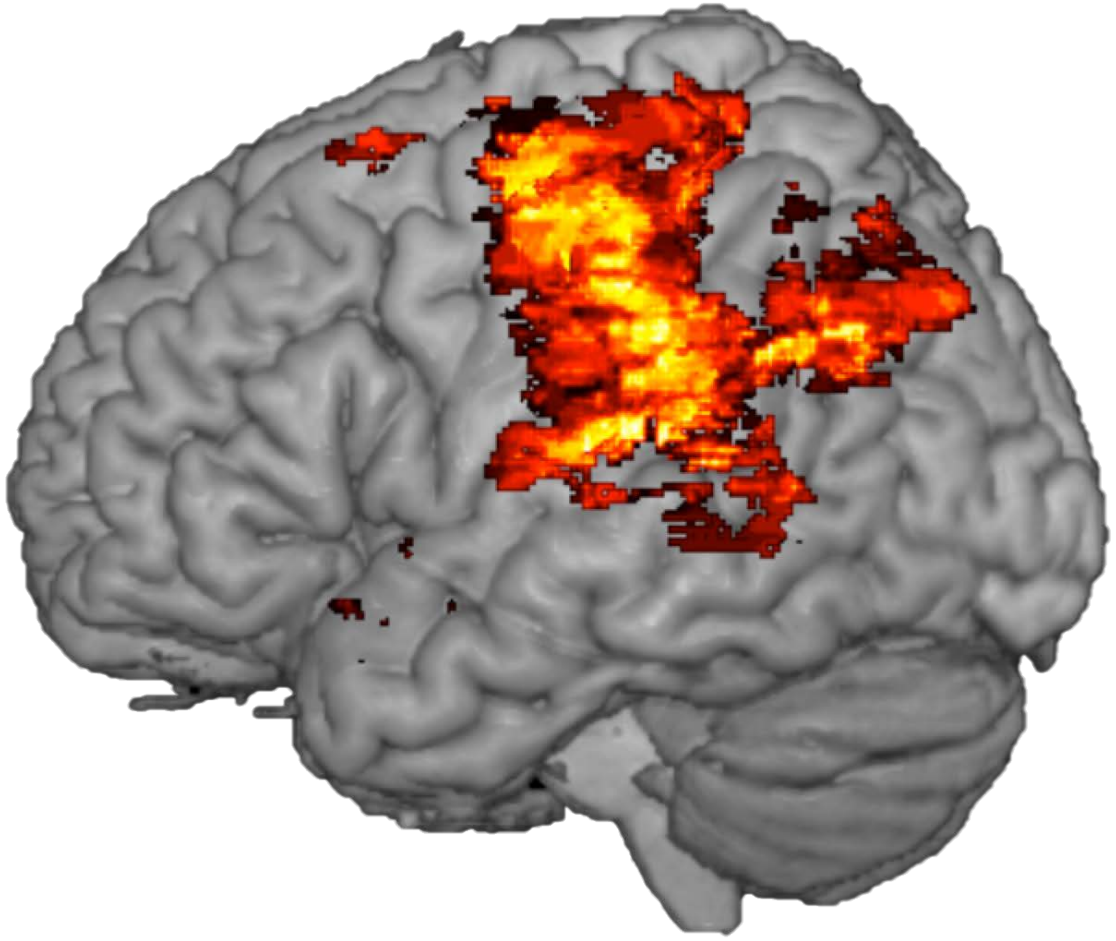
Gesturing Tool Use, voxelwise Z-value min, 1.65, $p < .05$, one-tailed

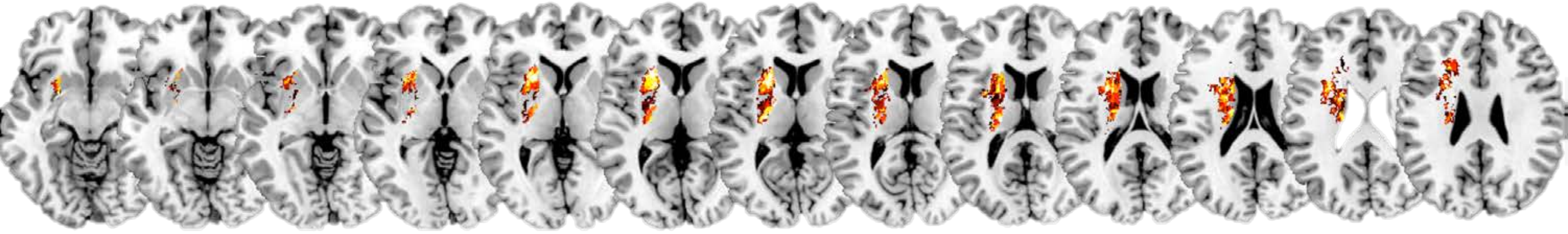
SVR-LSM: Multiple comparisons corrections

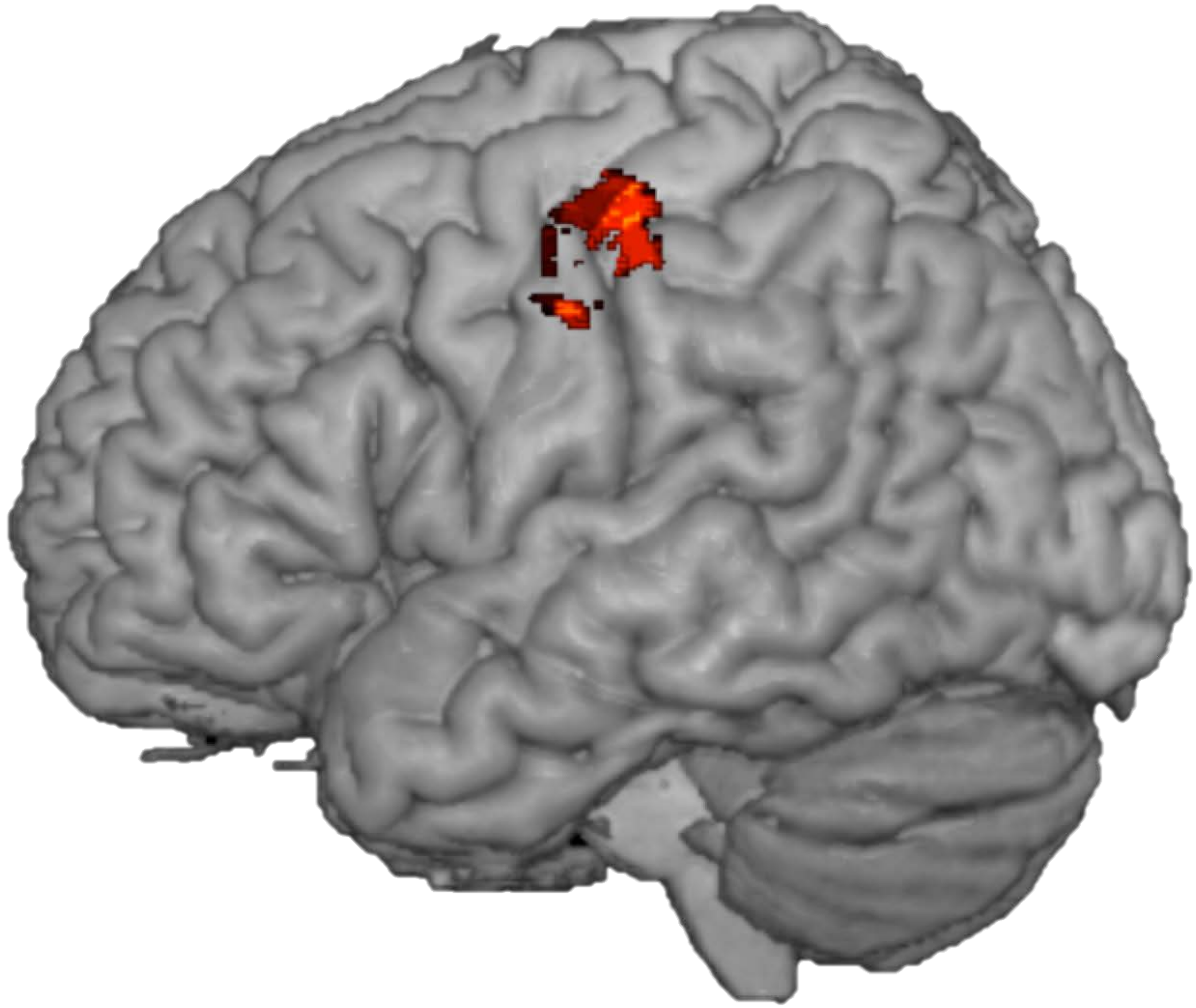
1. Raw beta values as the first output in SVR-LSM analysis.
   - Uninterpretable by itself.
2. Voxel-level correction (FWER or FDR).
   - Permutation testing to obtain null distribution in each voxel; then z-score your true data relative to null data.
3. Cluster-level correction (FWER or FDR)
   - Identify clusters of significant voxels
   - Remove clusters k < X contiguous voxels



```
function clusterizeMap
% this script will remove voxels that do not survive cluster level correction
% clusters are determined based on a k value, which the user must input.
% that k value is equivalent to the number of continguous 1 mm^3 anatomical voxels that must surpass that
% threshold to survive correction.
%% k is your threshold for determining cluster extent size.
k = 500;
```


Gesturing Tool Use, voxelwise Z-value min, 1.65 ($p < .05$, one-tailed), 500 voxel cluster min.





Flanker Task

Congruent

< < $\boxed{<}$ < <

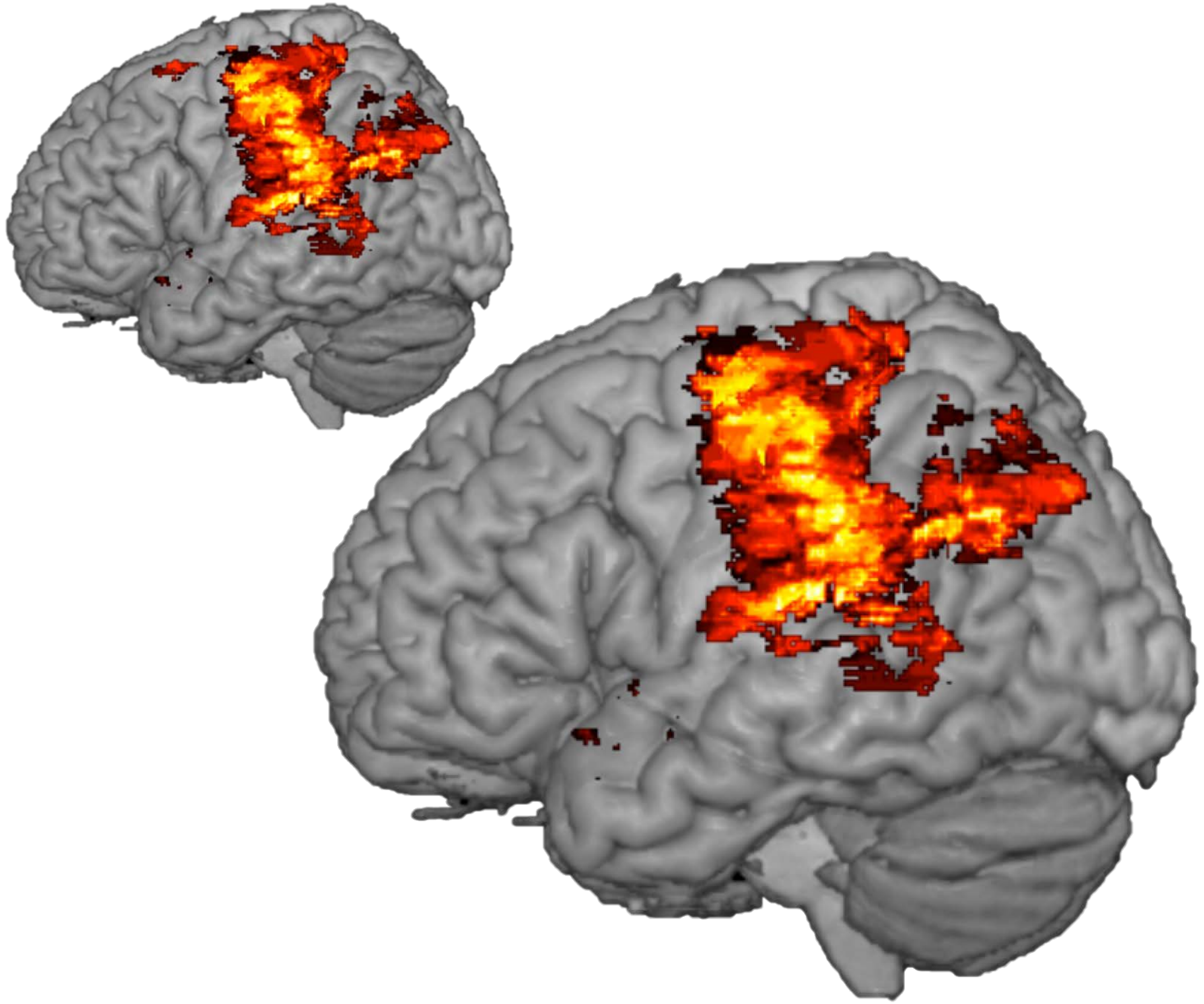
Incongruent

> > $\boxed{<}$ > >

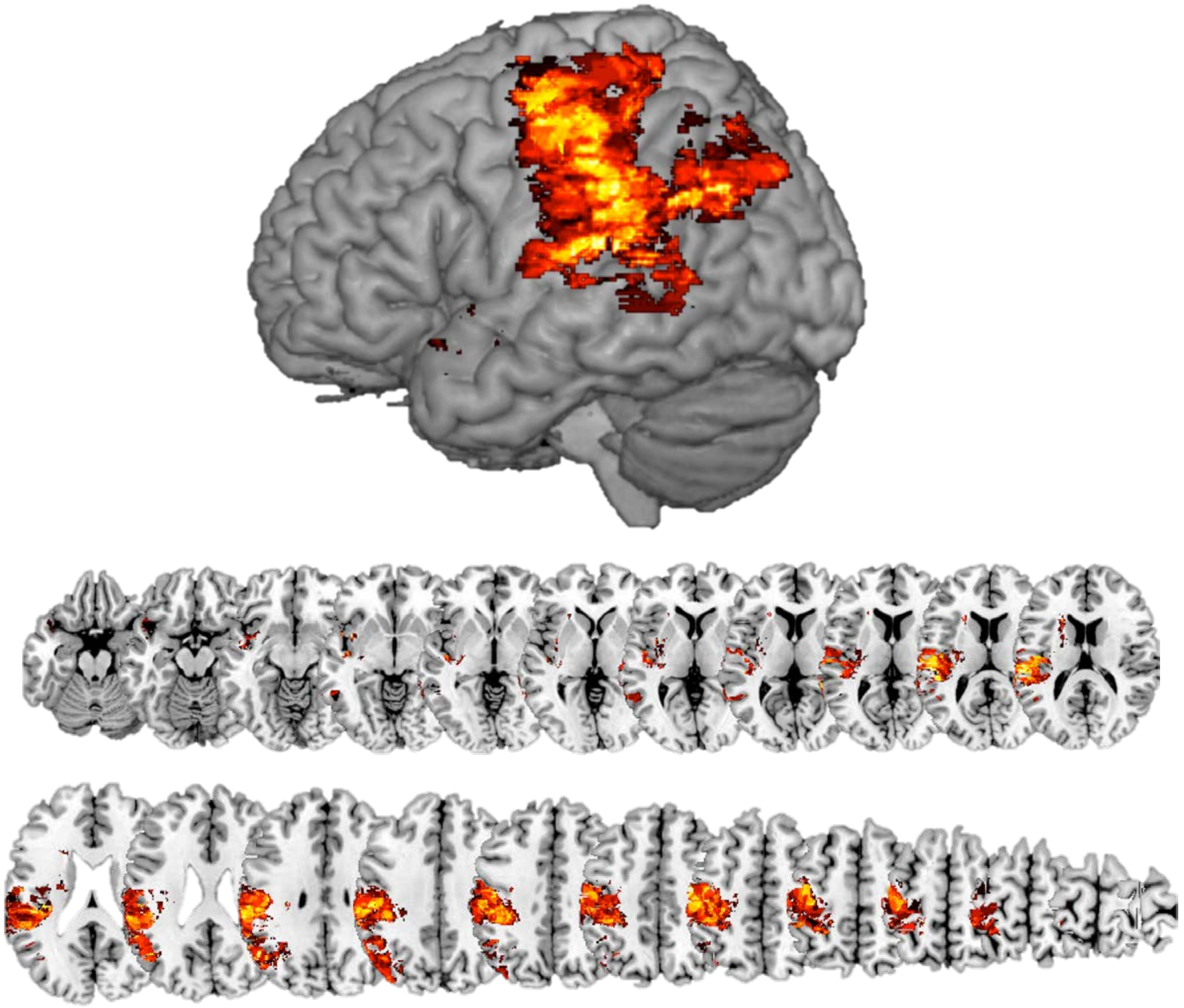
Garcea et al., in prep.

SVR-LSM: Multiple comparisons corrections

1. Raw beta values as the first output in SVR-LSM analysis.
   - Uninterpretable by itself.
2. Voxel-level correction (FWER or FDR).
   - Permutation testing to obtain null distribution in each voxel; then z-score your true data relative to null data.
3. Cluster-size threshold
   - Identify clusters of significant voxels
   - Remove clusters k < X contiguous voxels
4. Cluster-level size correction (DeMarco et al 2018)
   - Identify clusters of significant voxels
   - Calculate p values from the beta values estimated on each random permutation, then identify clusters of significant voxels
   - Find the size of the largest cluster that is significant “by chance”
   - This provides a null distribution of cluster *sizes*
   - Compare our real clusters to this distribution to identify those clusters that are significantly “large enough” in an FWER sense



Gesturing Tool Use, voxelwise Z-value min, 1.65 ($p < .05$, one-tailed), cluster corrected



Post-analysis Processing

1. Multiple comparison correction.
   1. SVR-LSM toolbox
   2. Custom matlab scripts

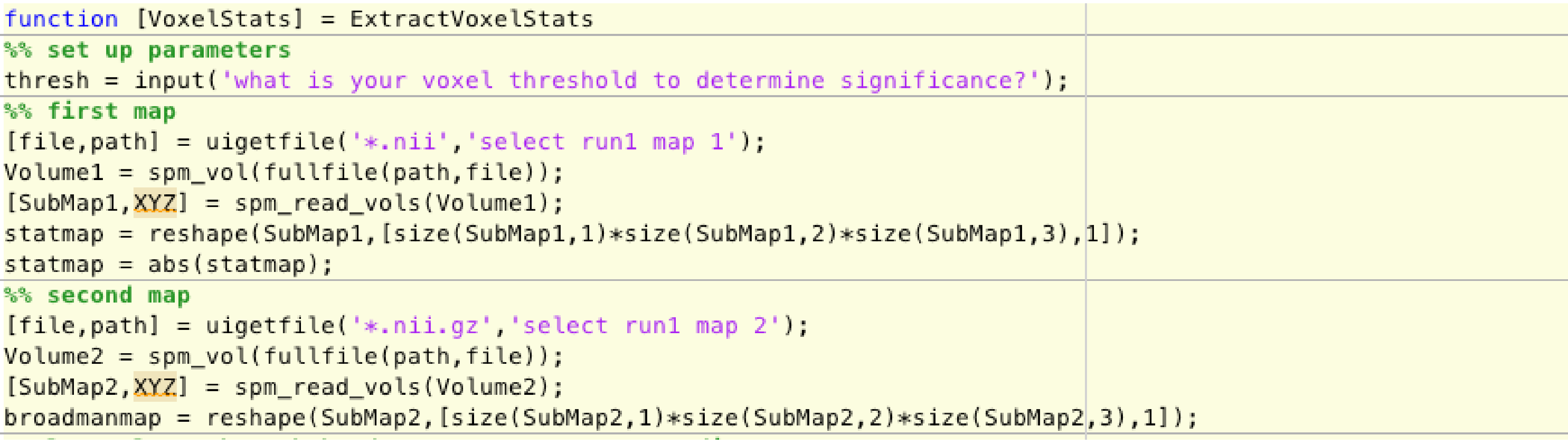
2. Cluster-size thresholding the voxelwise z-map.

3. Viewing results before and after cluster correction.

4. Independently label your peaks and clusters (AAL template)

ROIs and your whole-brain maps

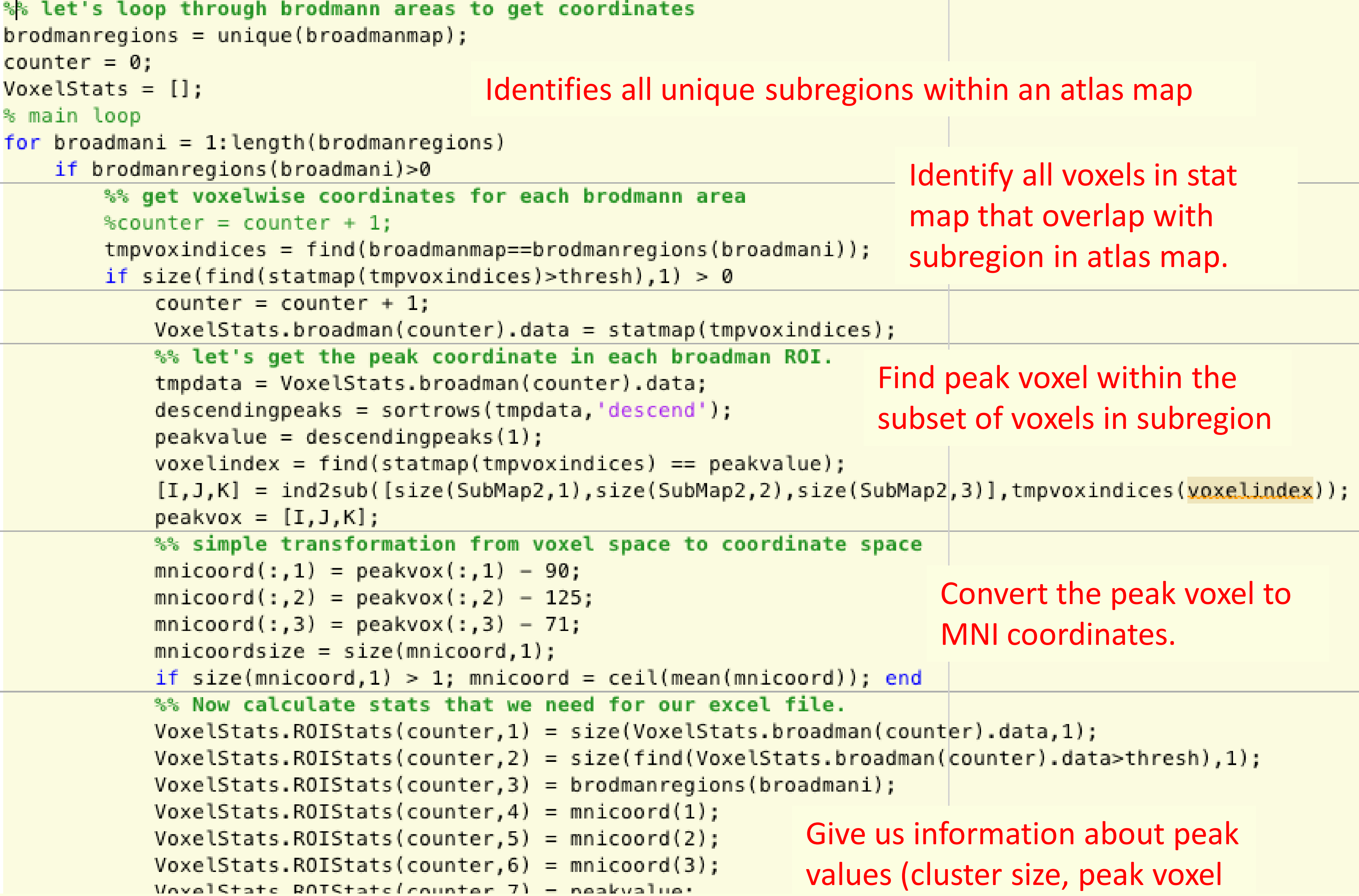
- Now that we have a whole-brain map, let’s identify peaks and voxel coordinates.
- Matlab scripts take as input a statistical map, an atlas map, and ask for a threshold of significance.



```
function [VoxelStats] = ExtractVoxelStats
%% set up parameters
thresh = input('what is your voxel threshold to determine significance?');
%% first map
[file,path] = uigetfile('*.nii','select run1 map 1');
Volume1 = spm_vol(fullfile(path,file));
[SubMap1,XYZ] = spm_read_vols(Volume1);
statmap = reshape(SubMap1,[size(SubMap1,1)*size(SubMap1,2)*size(SubMap1,3),1]);
statmap = abs(statmap);
%% second map
[file,path] = uigetfile('*.nii.gz','select run1 map 2');
Volume2 = spm_vol(fullfile(path,file));
[SubMap2,XYZ] = spm_read_vols(Volume2);
broadmanmap = reshape(SubMap2,[size(SubMap2,1)*size(SubMap2,2)*size(SubMap2,3),1]);
```

ROIs and your whole-brain maps

- Now that we have a whole-brain map, let’s identify peaks and voxel coordinates.
- Matlab scripts take as input a statistical map, an atlas map, and ask for a threshold of significance.
- Then identify the peak voxel in the statistical map (strongest supra-threshold value) in every subregion of an atlas map.



```
%% let's loop through brodmann areas to get coordinates
brodmanregions = unique(broadmanmap);
counter = 0;
VoxelStats = [];
% main loop
for broadmani = 1:length(brodmanregions)
    if brodmanregions(broadmani)>0
```

Identifies all unique subregions within an atlas map

```
        %% get voxelwise coordinates for each brodmann area
        %counter = counter + 1;
        tmpvoxindices = find(broadmanmap==brodmanregions(broadmani));
        if size(find(statmap(tmpvoxindices)>thresh),1) > 0
```

Identify all voxels in stat map that overlap with subregion in atlas map.

```
            counter = counter + 1;
            VoxelStats.broadman(counter).data = statmap(tmpvoxindices);
            %% let's get the peak coordinate in each broadman ROI.
            tmpdata = VoxelStats.broadman(counter).data;
            descendingpeaks = sortrows(tmpdata,'descend');
            peakvalue = descendingpeaks(1);
            voxelindex = find(statmap(tmpvoxindices) == peakvalue);
            [I,J,K] = ind2sub([size(SubMap2,1),size(SubMap2,2),size(SubMap2,3)],tmpvoxindices(voxelindex));
            peakvox = [I,J,K];
```

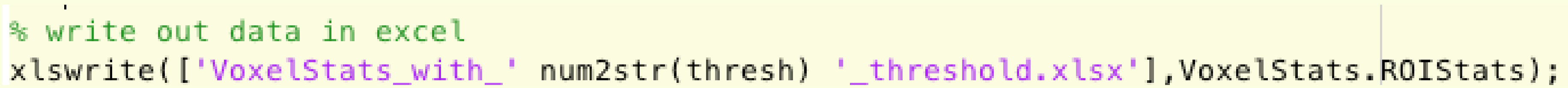
Find peak voxel within the subset of voxels in subregion

```
            %% simple transformation from voxel space to coordinate space
            mnicoord(:,1) = peakvox(:,1) - 90;
            mnicoord(:,2) = peakvox(:,2) - 125;
            mnicoord(:,3) = peakvox(:,3) - 71;
            mnicoordsize = size(mnicoord,1);
            if size(mnicoord,1) > 1; mnicoord = ceil(mean(mnicoord)); end
```

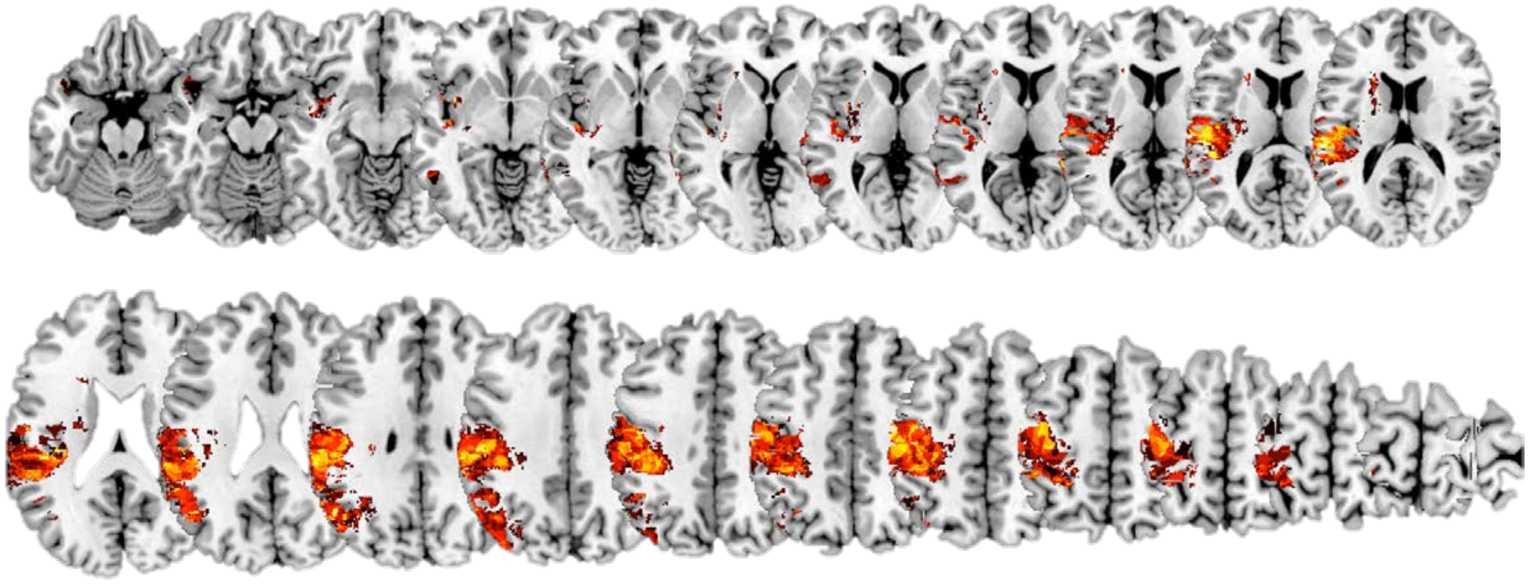
Convert the peak voxel to MNI coordinates.

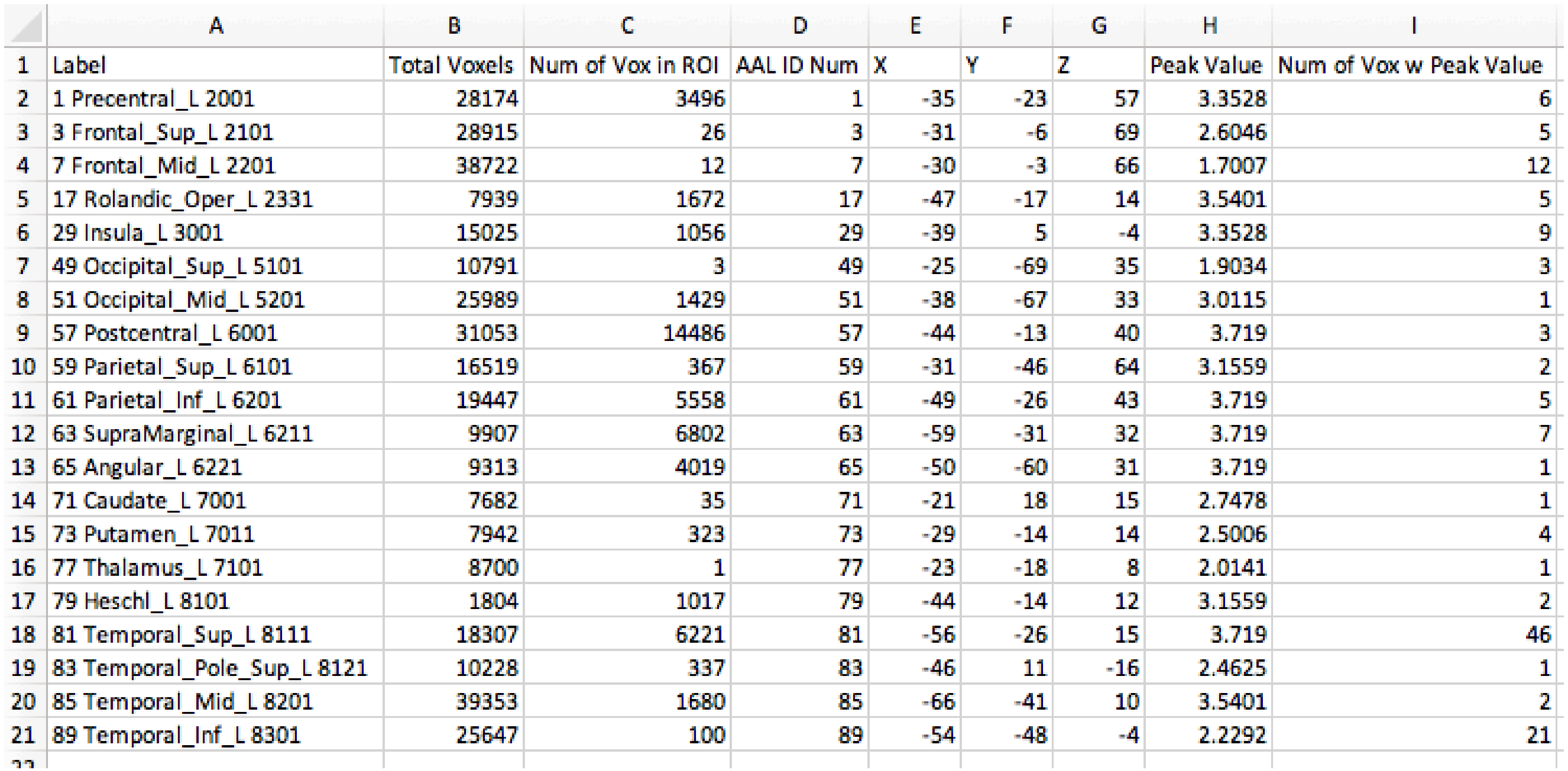
```
            %% Now calculate stats that we need for our excel file.
            VoxelStats.ROIStats(counter,1) = size(VoxelStats.broadman(counter).data,1);
            VoxelStats.ROIStats(counter,2) = size(find(VoxelStats.broadman(counter).data>thresh),1);
            VoxelStats.ROIStats(counter,3) = brodmanregions(broadmani);
            VoxelStats.ROIStats(counter,4) = mnicoord(1);
            VoxelStats.ROIStats(counter,5) = mnicoord(2);
            VoxelStats.ROIStats(counter,6) = mnicoord(3);
            VoxelStats.ROIStats(counter,7) = peakvalue;
```

Give us information about peak values (cluster size, peak voxel



```
% write out data in excel
xlswrite(['VoxelStats_with_' num2str(thresh) '_threshold.xlsx'],VoxelStats.ROIStats);
```





	A	B	C	D	E	F	G	H	I
1	Label	Total Voxels	Num of Vox in ROI	AAL ID Num	X	Y	Z	Peak Value	Num of Vox w Peak Value
2	1 Precentral_L 2001	28174	3496	1	-35	-23	57	3.3528	6
3	3 Frontal_Sup_L 2101	28915	26	3	-31	-6	69	2.6046	5
4	7 Frontal_Mid_L 2201	38722	12	7	-30	-3	66	1.7007	12
5	17 Rolandic_Oper_L 2331	7939	1672	17	-47	-17	14	3.5401	5
6	29 Insula_L 3001	15025	1056	29	-39	5	-4	3.3528	9
7	49 Occipital_Sup_L 5101	10791	3	49	-25	-69	35	1.9034	3
8	51 Occipital_Mid_L 5201	25989	1429	51	-38	-67	33	3.0115	1
9	57 Postcentral_L 6001	31053	14486	57	-44	-13	40	3.719	3
10	59 Parietal_Sup_L 6101	16519	367	59	-31	-46	64	3.1559	2
11	61 Parietal_Inf_L 6201	19447	5558	61	-49	-26	43	3.719	5
12	63 SupraMarginal_L 6211	9907	6802	63	-59	-31	32	3.719	7
13	65 Angular_L 6221	9313	4019	65	-50	-60	31	3.719	1
14	71 Caudate_L 7001	7682	35	71	-21	18	15	2.7478	1
15	73 Putamen_L 7011	7942	323	73	-29	-14	14	2.5006	4
16	77 Thalamus_L 7101	8700	1	77	-23	-18	8	2.0141	1
17	79 Heschl_L 8101	1804	1017	79	-44	-14	12	3.1559	2
18	81 Temporal_Sup_L 8111	18307	6221	81	-56	-26	15	3.719	46
19	83 Temporal_Pole_Sup_L 8121	10228	337	83	-46	11	-16	2.4625	1
20	85 Temporal_Mid_L 8201	39353	1680	85	-66	-41	10	3.5401	2
21	89 Temporal_Inf_L 8301	25647	100	89	-54	-48	-4	2.2292	21