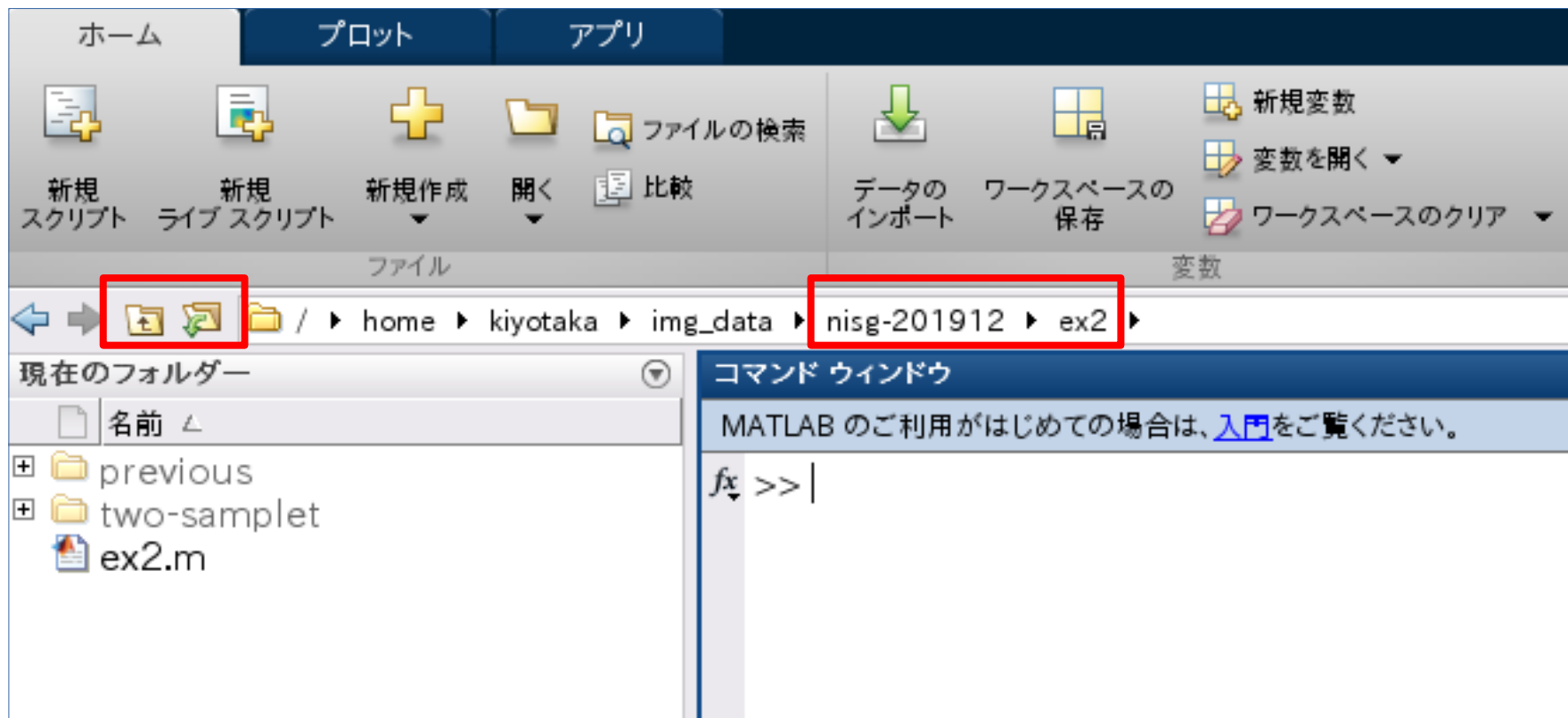


# SPM.mat と xSPM の探検

筑波大学医学医療系精神医学  
根本清貴

# この時間の作業ディレクトリ

- nisg-201912/ex2 がこの時間のディレクトリです
- Matlabでまず、ここに移動してください



# チートシート

- ex2\_talk.m にこのスライドにある全てのコマンドが記載されています
- タイプがつかなくなった時などにどうぞ

>> `edit ex2_talk.m`

- 「実行して次に進む」が便利



# SPM.mat と xSPM

- SPM.mat
  - 自分が行った統計解析に関する様々な情報が構造体としておさめられている
  - 自動で保存されるためいつでも呼び出し可
- xSPM
  - 自分が行った統計解析の結果の情報がおさめられている
  - SPMを閉じると消えてしまう

# SPM.mat と xSPM の場所

- SPM.matは、SPMでモデルを作成する際に設定したディレクトリに生成される
- xSPMは、SPMで結果を出した際に、統計に用いたSPM.matと同じディレクトリに生成される
- xSPMは構造体なので、名前をつけて保存できる
- 保存するファイル名に統計閾値を入れるのもひとつ
  - 例

```
>> save('xSPM_FWE05ext100.mat', 'xSPM')
```

# 健常者と統合失調症の群間比較

- 健常者71名と統合失調症患者71名の群間比較
- クラスターレベルでの多重比較補正の結果を表示
  - $p < 0.001$ , extent threshold 1490
- Matlabから以下を実行（SPMは起動しない）

```
>> clear all
```

```
>> cwd = pwd;
```

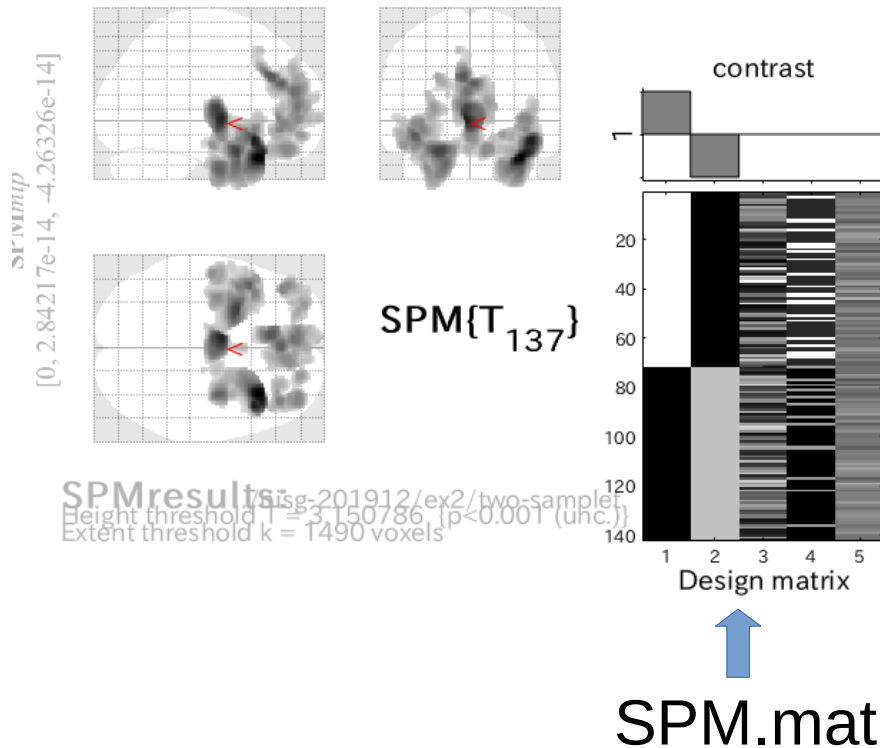
```
>> ex2
```

# cwd = pwd

- **pwd** はMatlabのコマンドで「現在のディレクトリ」の意味
- 今、**pwd** の結果を、**cwd** という変数に代入している
- **cwd**は、current working directory の略で、「今回の作業ディレクトリ」のような意味
- 主に作業しているところにすぐに戻ってこれるようにこのような設定をしておくと、後で簡単に戻ってこれる

# 健常者と統合失調症の群間比較

Control > Schizophrenia



xSPM



Statistics: *p-values adjusted for search volume*

set-level		cluster-level				peak-level				mm mm mm		
$p$	$c$	$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	$k_E$	$p_{\text{uncorr}}$	$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	$T$	$(Z)$	$p_{\text{uncorr}}$		
0.0005		0.000	0.000	5283	0.000	0.000	0.002	6.19	5.80	0.000	34	21
						0.001	0.009	5.73	5.41	0.000	44	18
						0.027	0.092	4.79	4.60	0.000	36	-9
		0.000	0.000	8486	0.000	0.001	0.009	5.63	5.33	0.000	-4	-7
						0.006	0.040	5.19	4.95	0.000	-26	5
						0.021	0.092	4.86	4.66	0.000	-44	14
		0.000	0.000	7347	0.000	0.015	0.083	4.94	4.73	0.000	-8	32
						0.024	0.092	4.82	4.63	0.000	0	38
						0.052	0.144	4.61	4.43	0.000	-4	48
		0.011	0.026	1642	0.003	0.117	0.215	4.36	4.22	0.000	39	38
						0.117	0.215	4.36	4.21	0.000	39	51
						0.127	0.215	4.34	4.19	0.000	40	36
		0.016	0.030	1490	0.004	0.124	0.215	4.34	4.20	0.000	-39	39
						0.275	0.348	4.07	3.95	0.000	-28	56
						0.696	0.502	3.64	3.55	0.000	-30	47

*table shows 3 local maxima more than 8.0mm apart*

Height threshold:  $T = 3.15$ ,  $p = 0.001$  (0.986 Degrees of freedom = [1.0, 137.0])  
Extent threshold:  $k = 1490$  voxels,  $p = 0.004$  (0.016) 16.9 16.4 16.3 mm mm mm; 11.3 10.9 10.9  
Expected voxels per cluster,  $\langle k \rangle = 150.778$  Volume: 1386899 = 410933 voxels = 270.9 resels  
Expected number of clusters,  $\langle c \rangle = 0.02$  Voxel size: 1.5 1.5 1.5 mm mm mm; (resel = 1341.2)  
FWEp: 4.617, FDRp: 5.190, FWEc: 1490, FDRc: 1490



# こんな経験ありませんか？

- 座標の領域名を求めるの、地味に大変…
- 結果を見ると、自分が考えている領域が入っているはずなのに、それがない…
- もっとたくさん座標があるはずだから、それを全部見たい…
- こういうときに、xSPMが役立つ

# SPM.matの中身

- MatlabでSPM.matの中身を見してみる

```
>> SPM %必ず大文字 小文字だとspm12が起動する
```

```
xY: [1×1 struct]
```

```
nscan: 142
```

```
xX: [1×1 struct]
```

```
xC: [1×3 struct]
```

```
xGX: [1×1 struct]
```

```
xM: [1×1 struct]
```

```
xsDes: [1×1 struct]
```

… 以下略

# SPM.mat の要素の確認

- 使用した画像ファイル一覧

>> **SPM.xY.P**

- Design matrix

>> **SPM.xX.X**

# xSPM

>> **xSPM**

xSPM =

フィールドをもつ struct:

swd: 'img\_data/nisg-201912/ex2/two-samplet'

title: 'Control > Schizophrenia'

**Z: [1×24248 double]**

n: 1

STAT: 'T'

df: [1.0000 137.0000]

STATstr: 'T\_{137}'

Ic: 1

Im: []

pm: []

Ex: []

u: 3.1508

k: 1490

**XYZ: [3×24248 double]**

**XYZmm: [3×24248 double]**

S: 410933

R: [-2 -5.7408 392.8758 270.8770]

FWHM: [11.2830 10.9299 10.8756]

M: [4×4 double]

iM: [4×4 double]

DIM: [3×1 double]

VOX: [1.5000 1.5000 1.5000]

Vspm: [1×1 struct]

thresDesc: 'p<0.001 (unc.)'

VRpv: [1×1 struct]

units: {'mm' 'mm' 'mm'}

Pp: [1×115 double]

Pc: [1×39 double]

uc: [4.6166 5.1898 1490 1490]

# xSPM.Z

- 現在有意になったすべてのボクセルの統計値がおさめられている
- Zとあるが、この値は、xSPM.STATで示されているものの統計値
  - 現在の xSPM.STAT='T' であるからT値
- Z: [1×24248 double]
  - 1行24248列で入っている

# 24248はどこから？

- 有意なボクセル数の合計

>> 5283 + 8486 +  
7347 + 1642 + 1490

ans =

24248

Statistics: *p-values adjusted for search*

set-level		cluster-level				p	
p	c	p <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>	k <sub>E</sub>	p <sub>uncorr</sub>	p <sub>FWE-corr</sub>	q <sub>FDR-corr</sub>
0.0005		0.000	0.000	5283	0.000	0.000	0.002
						0.001	0.009
						0.027	0.092
		0.000	0.000	8486	0.000	0.001	0.009
						0.006	0.040
						0.021	0.092
		0.000	0.000	7347	0.000	0.015	0.083
						0.024	0.092
						0.052	0.144
		0.011	0.026	1642	0.003	0.117	0.215
						0.117	0.215
						0.127	0.215
		0.016	0.030	1490	0.004	0.124	0.215
						0.275	0.348
						0.696	0.502

table shows 3 local maxima more than 8

Height threshold: T = 3.15, p = 0.001 (0.986 Degrees of freedom)  
Extent threshold: k = 1490 voxels, p = 0.004 (0.016)  
Expected voxels per cluster, <k> = 150.778 Volume: 1386  
Expected number of clusters, <c> = 0.02 Voxel size: 1.5  
FWEp: 4.617, FDRp: 5.190, FWEc: 1490, FDRc: 1490

## xSPM.Zの101～105列

```
>> t = xSPM.Z(:,101:105)
```

```
t =
```

```
3.3274 3.2489 3.2935 3.4063 3.4368
```

- これがあるボクセルはどこ? → xSPM.XYZ に入っている

# xSPM.XYZ の101～105列

```
>> XYZ = xSPM.XYZ(:,101:105)
```

XYZ =

80	81	78	79	80
85	85	86	86	86
18	18	18	18	18

- ベクトルは縦ベクトルの集まりであることに注意  
[80 85 18] [81 85 18] [78 86 18] ...
- このボクセルのMNI座標は？ → xSPM.XYZmm



# xSPM.XYZmm の101～105列

```
>> XYZmm = xSPM.XYZmm(:,101:105)
```

XYZmm =

-28.5	-30.0	-25.5	-27.0	-28.5
0.0	0.0	1.5	1.5	1.5
-46.5	-46.5	-46.5	-46.5	-46.5

- ベクトルは縦ベクトルの集まりであることに注意

[-28.5 0.0 -46.5] [-30.0 0.0 -46.5] [-25.5 1.5 -46.5] ...

# 今得られた3つの関係

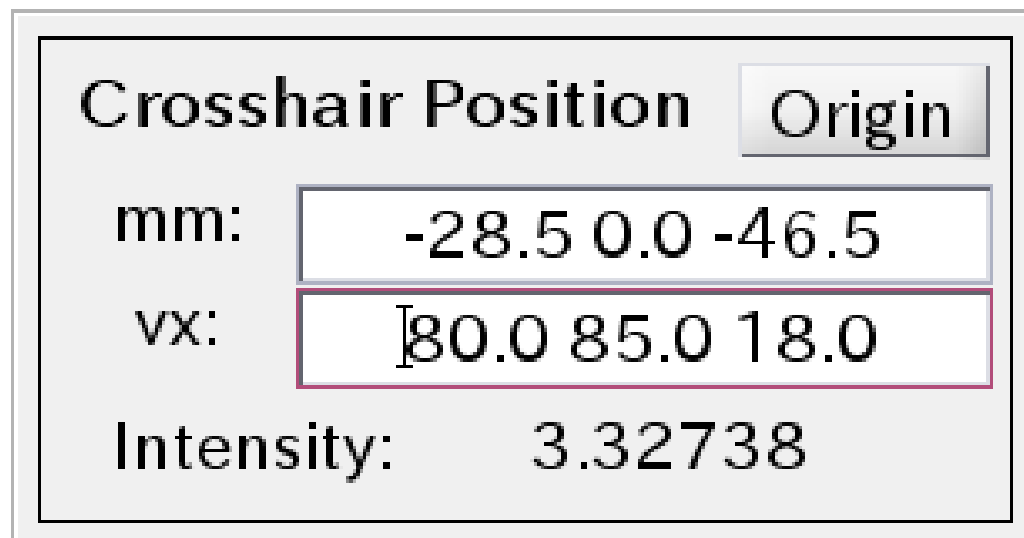
>> [XYZ' XYZmm' t'] %行列の転置

Voxel			MNI coordinates			T
80	85	18	-28.5	0.0	-46.5	3.33
81	85	18	-30.0	0.0	-46.5	3.25
78	86	18	-25.5	1.5	-46.5	3.29
79	86	18	-27.0	1.5	-46.5	3.41
80	86	18	-28.5	1.5	-46.5	3.44

# SPMで見るこれらの関係

```
>> spm_image('Display','spmT_0001.nii')
```

- vxに 80 85 18 と入力して、Enter



A screenshot of the 'Crosshair Position' dialog box in SPM. The dialog has a title bar and a 'Origin' button. It contains three input fields: 'mm:' with the value '-28.5 0.0 -46.5', 'vx:' with the value '80.0 85.0 18.0' (highlighted with a red border), and 'Intensity:' with the value '3.32738'.

Voxel			MNI coordinates			T
80	85	18	-28.5	0.0	-46.5	3.33

# 座標のインデックスを確認

- 特定の座標のt値を求めたい
- このような時は、座標のインデックスが有用
- 書式

`[xyz i]=spm_XYZreg('NearestXYZ',[知りたい座標],xSPM.XYZmm)`

- 座標は縦ベクトルであることに注意
- SPMではMNI座標は小数点で入っている
  - 'NearestXYZ'の機能により、整数で入れると、それに一番近い座標を探してそのインデックスを返す

# 特定の座標のインデックスを確認

- $[-29 \ 0 \ -47]$  のt値を求めたい

```
>> [xyz i] = spm_XYZreg('NearestXYZ', [-29; 0; -47], xSPM.XYZmm)
```

```
xyz =
```

```
-28.5000
```

```
0.0000
```

```
-46.5000
```

```
i =
```

```
101 %インデックスが101とわかる
```

# インデックスを使って値を求める

- $[-29 \ 0 \ -47]$  のt値を求めたい(続き)

```
>> tvalue=xSPM.Z(:,i) % i=101
```

```
tvalue =
```

```
3.3274
```

# SPMに搭載されているアトラス

- SPMには様々なアトラスが搭載可能な状態となっている
- 現在搭載されているアトラスは、以下で確認できる

```
>> L = spm_atlas('list')
```

```
L =
```

```
file:
```

```
'spm12/tpm/labels_Neuromorphometrics.xml'
```

```
name: 'Neuromorphometrics'
```

- 現在は、このNeuromorphometricsアトラスのみ利用できる

# アトラスの場所

- SPMでは、'tpm' の下にアトラスがある
- `spm('dir')` がSPMのディレクトリなので、  
`cd(spm('dir'))` でSPMのディレクトリに移動できる

```
>> cd(spm('dir'))
```

```
>> cd tpm
```

```
>> spm_image %引数がないとファイル選択GUIが出るので、  
そのまま labels_Neuromorphometrics.nii を選択
```

```
>> cd(cwd) %最初に設定したディレクトリに戻る
```



# cd の2つの使い方

- Matlabでは `cd` は2つの使い方がある

- ディレクトリを直接指定する時

- シェルと同様

```
cd /path/to/dir
```

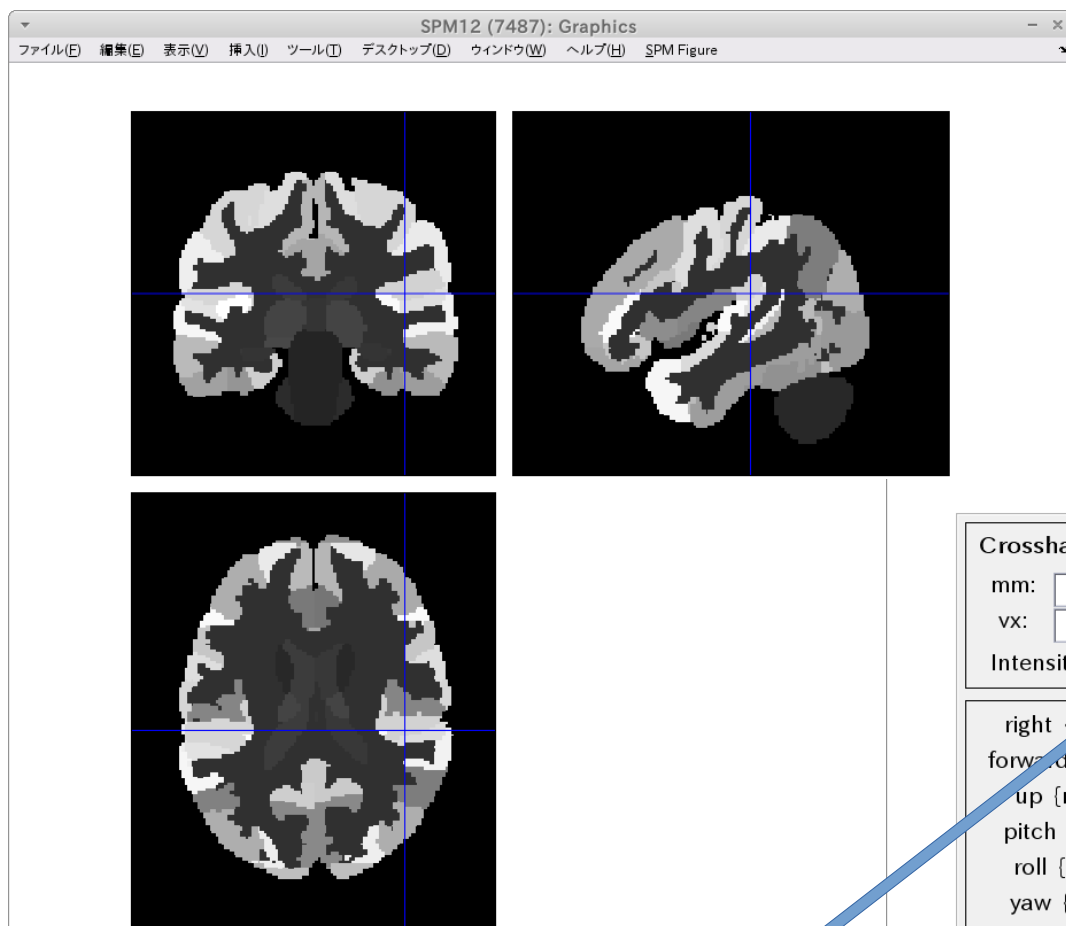
- ディレクトリのパスが変数に入っている時

- `cd(変数)` とする

```
cwd = '/path/to/dir'
```

```
cd(cwd)
```

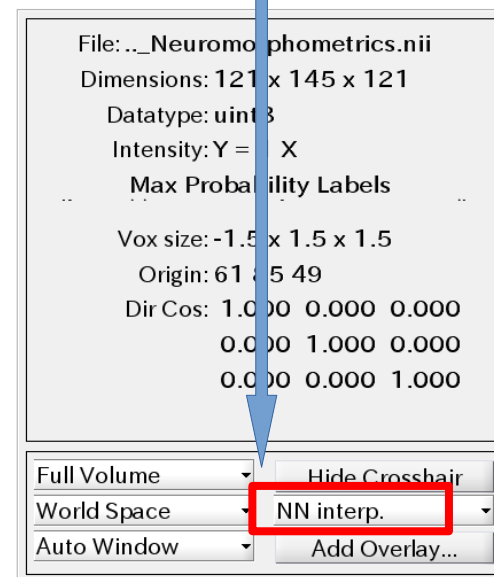
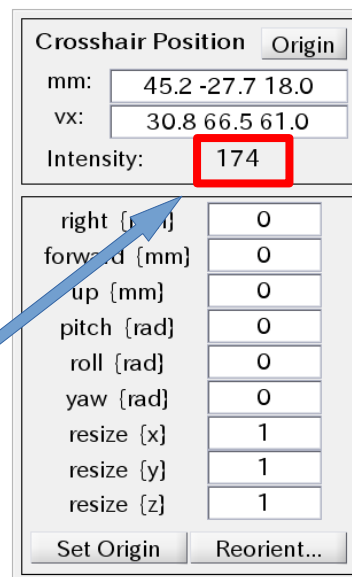
# labels\_Neuromorphometrics



アトラスの「ラベル label」

\*このラベルの領域名は、  
`spm_atlas('select','neuromorphometrics',174)`  
で求められる

- Interpolationを"NN"  
(Nearest Neighbor) に変更
- アトラスは、すべて整数で値が入っている
- Nearest Neighborでないと、小数が出てきてしまう



# spm\_atlasを使った領域名の同定

- `spm_atlas('query', アトラス名, [座標の縦ベクトル])`
- `spm_atlas` の後の第1引数に `'query'` をつけることで、領域名を調べることができる (query: 問い合わせ)
- アトラス名は、現在は `'Neuromorphometrics'` 一択
  - 最初のNは大文字でも小文字でもOKなので、本スライドでは小文字で統一

# 特定の座標の領域名を求める

- $[-29 \ 0 \ -47]$ の領域名を求める

- `>>`

```
spm_atlas('query','neuromorphom  
etrics',[-29; 0; -47])
```

```
ans =
```

```
    'Left ITG inferior temporal  
gyrus'
```

# 複数の座標の領域名を求める

- ちょっと工夫をすると、複数の座標の領域名を一気に求めることもできる
- XYZmm に入っている5つの座標の領域名を求めたい
- そのために必要なコマンド
  - size
  - for

# size

- 行列の大きさを表示するコマンド

```
>> size(XYZmm)
```

```
ans =
```

```
     3     5
```

- 3行5列であることがわかる
- 行数だけ抜き出したかったら `size(XYZmm,1)`
- 列数だけ抜き出したかったら `size(XYZmm,2)`

# for

- 繰り返しのコマンド
- Matlabでは繰り返しの回数を最初に指定する

**for** **i=1:繰り返し回数** %**i**でなくてもよい

    繰り返す内容

**end**

# 複数の座標から領域名を求める

```
>> for j=1:size(XYZmm,2) %mniの列数だけ処理
spm_atlas('query','neuromorphometrics',XYZmm(:,j))
end
```

- 行列XYZmmの列数を求めなさい（今の場合は5）
- jを1から5まで1ずつ増やしながら、以下を実行しなさい

```
spm_atlas('query','neuromorphometrics',XYZmm(:,1))
spm_atlas('query','neuromorphometrics',XYZmm(:,2))
spm_atlas('query','neuromorphometrics',XYZmm(:,3))
spm_atlas('query','neuromorphometrics',XYZmm(:,4))
spm_atlas('query','neuromorphometrics',XYZmm(:,5))
```



# セル配列

- 今出された結果を変数に格納できたら使いやすい
- Matlabは文字列に関しては、通常、行列に入れるときに同じ文字数でないと入れられない
- 「セル配列」を使うと文字数が異なっても入れられる
- セル配列は `{ }` でくくることで作成できる

# 複数の座標から領域名を求めて セル配列に格納する

```
>> region={}; %空のセル配列を作成

>> for j=1:size(XYZmm,2) %列数だけ処理

region{j,1}=spm_atlas('query','neuromorp
hometrics',XYZmm(:,j));

end

>> region %領域が格納されたことがわかる
```

# ゆっくり理解

- `region{j,1}=spm_atlas('query','neuro-morphometrics',XYZmm(:,j));`
- この意味は、「regionというセル配列の、j行1列に、変数XYZmmの第j列の領域名を代入する」ということ

# SPMの結果に表示されている 座標から領域名を求める

- SPMの結果で表示される座標は、TabDat.dat という構造体の12列目のセル配列に入っている
- 以下で取り出すことができる

```
>> XYZmm = cell2mat(TabDat.dat(:,12)')
```

- 先程のものを組み合わせれば、領域名を求められる

```
>> region={};
```

```
>> for j=1:size(XYZmm,2)
```

```
region{j,1}=spm_atlas('query','neuromorphometrics',XYZmm  
(:,j));
```

```
end
```

```
>> region
```

# セル配列をcsv形式で書き出す

- セル配列は table を経由して csv形式として書き出せる

```
>> T = cell2table(region)
```

```
>> writetable(T, 'region.csv')
```

- 現在作業しているディレクトリに書き出されることに注意

# TabDat.dat についてもう少し

- TabDat.dat に、SPMの結果のテーブルの実際の値がそのままおさめられている

Statistics: *p-values adjusted for search volume*

set-level		cluster-level				peak-level					mm mm mm		
$p$	$c$	$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	$k_E$	$p_{\text{uncorr}}$	$p_{\text{FWE-corr}}$	$q_{\text{FDR-corr}}$	$T$	$(Z_{\equiv})$	$p_{\text{uncorr}}$			
0.000	5	0.000	0.000	5283	0.000	0.000	0.002	6.19	5.80	0.000	34	21	-29
						0.001	0.009	5.73	5.41	0.000	44	18	-15
						0.027	0.092	4.79	4.60	0.000	36	-9	-35
		0.000	0.000	8486	0.000	0.001	0.009	5.63	5.33	0.000	-4	-7	-0
						0.006	0.040	5.19	4.95	0.000	-26	5	-23
						0.021	0.092	4.86	4.66	0.000	-44	14	-12
		0.000	0.000	7347	0.000	0.015	0.083	4.94	4.73	0.000	-8	32	30
						0.024	0.092	4.82	4.63	0.000	0	38	25
						0.052	0.144	4.61	4.43	0.000	-4	48	19
		0.011	0.026	1642	0.003	0.117	0.215	4.36	4.22	0.000	39	38	-8
						0.117	0.215	4.36	4.21	0.000	39	51	-0
						0.127	0.215	4.34	4.19	0.000	40	36	1
		0.016	0.030	1490	0.004	0.124	0.215	4.34	4.20	0.000	-39	39	-3
						0.275	0.348	4.07	3.95	0.000	-28	56	9
						0.696	0.502	3.64	3.55	0.000	-30	47	4

# TabDat

- TabDatが、SPMの結果のテーブル全体をおさめた構造体

>> **TabDat**

TabDat =

フィールドをもつ struct:

tit: 'p-values adjusted for search volume'

hdr: {3×12 cell}

fmt: {1×12 cell}

str: 'table shows 3 local maxima more than 8.0mm apart'

ftr: {9×2 cell}

dat: {15×12 cell}

# おみやげスクリプト

- `get_spm_names.m`
  - SPMの結果に表示されている座標の領域名と座標をCSVに書き出す
- `get_spm_all_names.m`
  - 統計解析で有意になったすべてのボクセルについて以下をCSVに書き出す
    - クラスタ番号、座標、領域名、T値
- `batch`ディレクトリをMatlabのパスに通す（既にパスが通っているディレクトリにスクリプトをコピーしても可）
- SPMの解析をした後、SPMを閉じる前にMATLABのコマンドウィンドウからスクリプトを呼び出すことで実行
- スクリプトが実行されたディレクトリ内に、"`region_names_実行した日付.csv`" "`region_all_names_実行した日付.csv`"のファイル名で保存される



# おみやげスクリプトの実行

>> **pwd** % nishg-201912/ex2 であることを確認

>> **addpath ../batch**

%nishg-201912/batch をパスに追加

>> **get\_spm\_names**

%region\_names\_日付.csv が生成

>> **get\_spm\_all\_names**

%region\_all\_names\_日付.csv が生成

- エクスプローラー (Win) や Finder (Mac) でCSVを確認

# リソース

- SPM.matの詳しい情報
  - SPM data structures
  - <http://people.duke.edu/~njs28/spmdatastructure.htm>
- xSPMについての情報
  - Matlab内から
    - >> **help spm\_getSPM**

Questions?