Thresholding SPM for Inference

Lecture Notes

Ahmet Ademoglu, *PhD*
Bogazici University
Institute of Biomedical Engineering

Some concepts and illustrations in this lecture are adapted from the textbook,
Statistical Parametric Mapping: The Analysis of Functional Brain Images, Editors: K. Friston, J. Ashburner, S. Kiebel, T. Nichols and W. Penny, *Academic Press*, 2006.

Inference on SPM

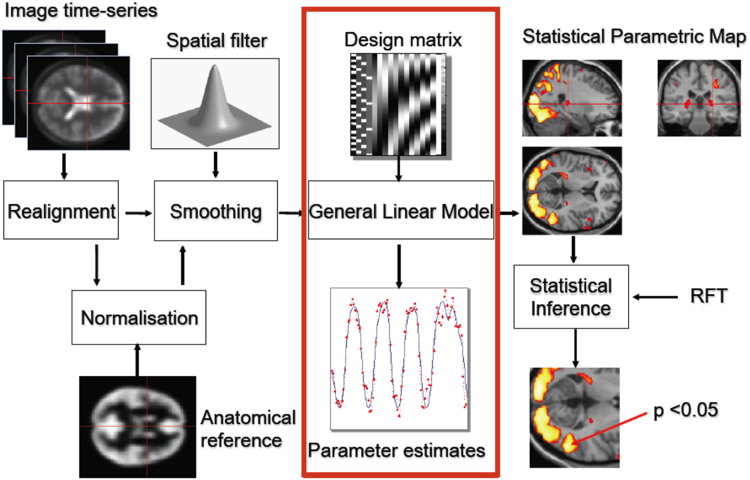


Image time-series

Spatial filter

Design matrix

Statistical Parametric Map

Realignment

Smoothing

General Linear Model

Normalisation

Statistical Inference

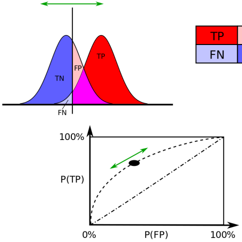
RFT

Anatomical reference

Parameter estimates

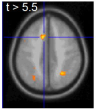
$p < 0.05$

ROC Curve: Sensitivity vs Specificity



TP	FP
FN	TN

High Threshold

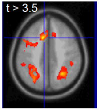


Good Specificity

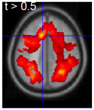
Poor Power

(risk of false negatives)

Med. Threshold



Low Threshold

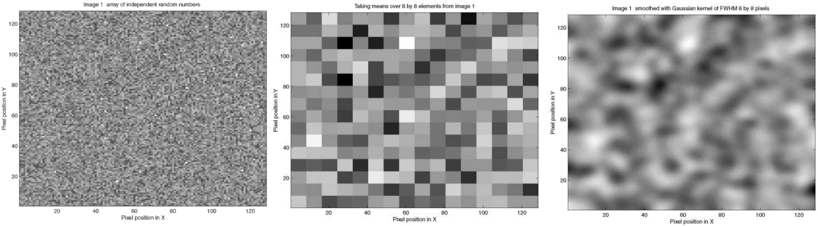


Poor Specificity

(risk of false positives)

Good Power

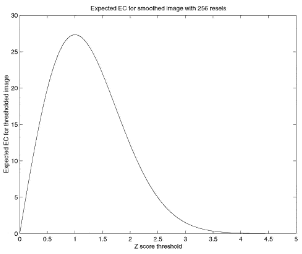
Family Wise Error (FWE) Rate



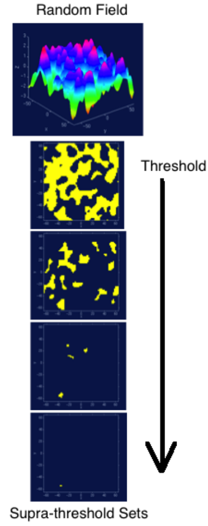
- For $p = 0.05$, the Bonferroni Correction $\alpha = 0.05/(128)^2$, $z = 4.52$
- For the averaged image $\alpha = 0.05/128$, $z = 3.55$.
- For the filtered image the z threshold is determined using Random Field Theory since the number of independent pixels are unknown.

Random Fielde Theory

Euler Characteristics χ_u
Topological Measure : # of blobs - # of holes
At high thresholds, only blobs are counted.



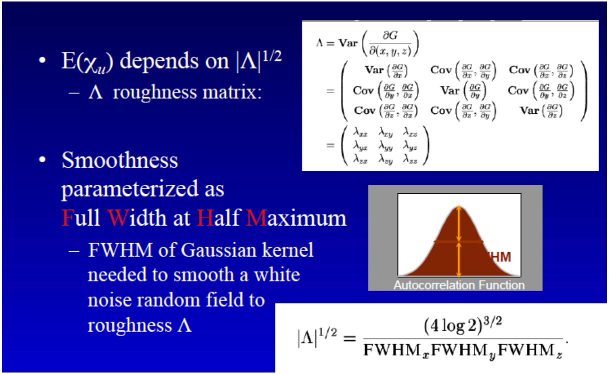
$$
\begin{aligned}
FWER &= P(Max\ voxel \geq u|H_0) \\
&= P(One\ or\ more\ blobs|H_0) \\
&\approx P(\chi_u \geq 1|H_0)
\end{aligned}
$$



Euler Characteristics, Smoothness Parameter

$$EC(\chi_t) = V|\Lambda|^{\frac{1}{2}}(2\pi)^{-2}(t^2-1)e^{-\frac{1}{2}t^2}$$

V: Volume,
$|\Lambda|^{\frac{1}{2}}$: number of resels per volume



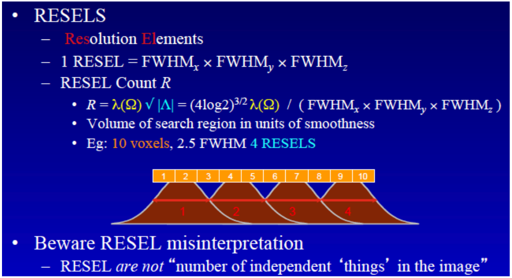
- $E(\chi_u)$ depends on $|\Lambda|^{1/2}$
  - Λ roughness matrix:

$$\Lambda = \mathbf{Var}\left(\frac{\partial G}{\partial(x,y,z)}\right) = \begin{pmatrix} \mathbf{Var}\left(\frac{\partial G}{\partial x}\right) & \mathbf{Cov}\left(\frac{\partial G}{\partial x},\frac{\partial G}{\partial y}\right) & \mathbf{Cov}\left(\frac{\partial G}{\partial x},\frac{\partial G}{\partial z}\right) \\ \mathbf{Cov}\left(\frac{\partial G}{\partial y},\frac{\partial G}{\partial x}\right) & \mathbf{Var}\left(\frac{\partial G}{\partial y}\right) & \mathbf{Cov}\left(\frac{\partial G}{\partial y},\frac{\partial G}{\partial z}\right) \\ \mathbf{Cov}\left(\frac{\partial G}{\partial z},\frac{\partial G}{\partial x}\right) & \mathbf{Cov}\left(\frac{\partial G}{\partial z},\frac{\partial G}{\partial y}\right) & \mathbf{Var}\left(\frac{\partial G}{\partial z}\right) \end{pmatrix} = \begin{pmatrix} \lambda_{xx} & \lambda_{xy} & \lambda_{xz} \\ \lambda_{yx} & \lambda_{yy} & \lambda_{yz} \\ \lambda_{zx} & \lambda_{zy} & \lambda_{zz} \end{pmatrix}$$

- Smoothness parameterized as Full Width at Half Maximum
  - FWHM of Gaussian kernel needed to smooth a white noise random field to roughness Λ

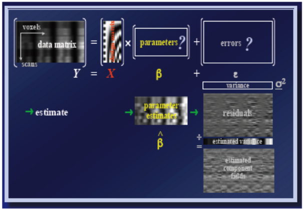
$$|\Lambda|^{1/2} = \frac{(4\log 2)^{3/2}}{\mathrm{FWHM}_x \mathrm{FWHM}_y \mathrm{FWHM}_z}.$$

RESEL: RESolution ELements



- RESELS
  - Resolution Elements
  - 1 RESEL = $FWHM_x \times FWHM_y \times FWHM_z$
  - RESEL Count R
    - $R = \lambda(\Omega) \sqrt{|\Lambda|} = (4\log 2)^{3/2} \lambda(\Omega) \;/\; (FWHM_x \times FWHM_y \times FWHM_z)$
    - Volume of search region in units of smoothness
    - Eg: 10 voxels, 2.5 FWHM 4 RESELS

- Beware RESEL misinterpretation
  - RESEL *are not* "number of independent 'things' in the image"

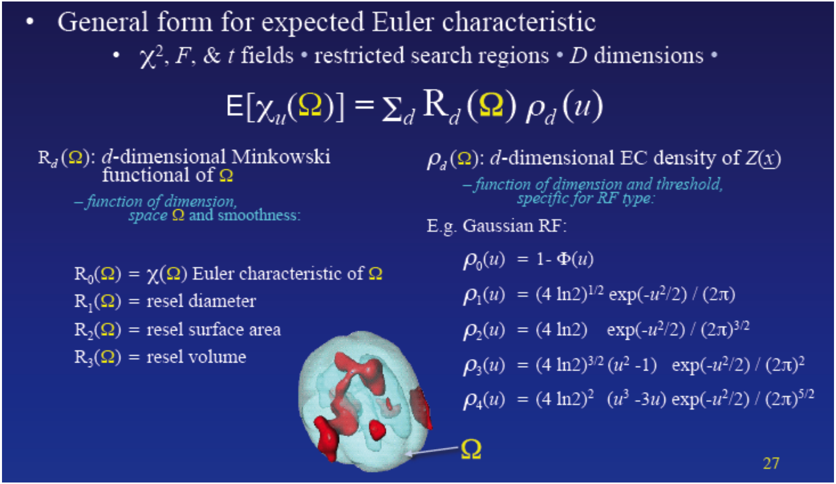


Variance of gradients calculated from the residuals yielding resels per voxel (RPV).

$$FWHM = RPV^{-1/D}$$

Corrected P value $P^c = P(maxT > t) \approx E(\chi_t))$

RFT



- General form for expected Euler characteristic
  - χ^2, F, & t fields • restricted search regions • D dimensions •

$$E[\chi_u(\Omega)] = \sum_d R_d(\Omega)\, \rho_d(u)$$

$R_d(\Omega)$: d-dimensional Minkowski functional of Ω

– *function of dimension, space* Ω and smoothness:

$R_0(\Omega) = \chi(\Omega)$ Euler characteristic of Ω

$R_1(\Omega) =$ resel diameter

$R_2(\Omega) =$ resel surface area

$R_3(\Omega) =$ resel volume

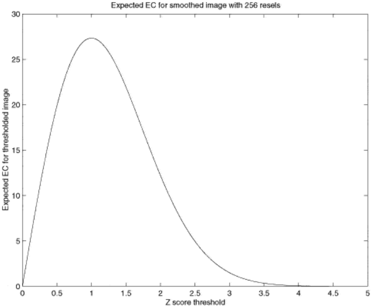
$\rho_d(\Omega)$: d-dimensional EC density of $Z(x)$

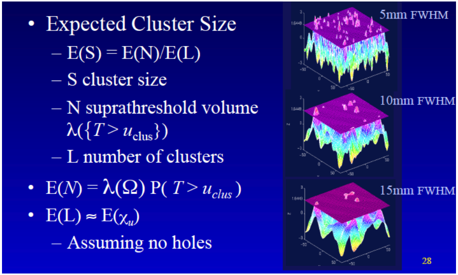
– *function of dimension and threshold, specific for RF type:*

E.g. Gaussian RF:

$$\rho_0(u) = 1 - \Phi(u)$$
$$\rho_1(u) = (4 \ln 2)^{1/2} \exp(-u^2/2) / (2\pi)$$
$$\rho_2(u) = (4 \ln 2) \quad \exp(-u^2/2) / (2\pi)^{3/2}$$
$$\rho_3(u) = (4 \ln 2)^{3/2} (u^2 - 1) \quad \exp(-u^2/2) / (2\pi)^2$$
$$\rho_4(u) = (4 \ln 2)^2 \quad (u^3 - 3u) \exp(-u^2/2) / (2\pi)^{5/2}$$

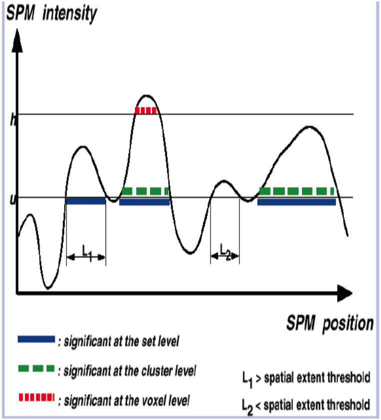
Euler Characteristics: $E(\chi_t)$





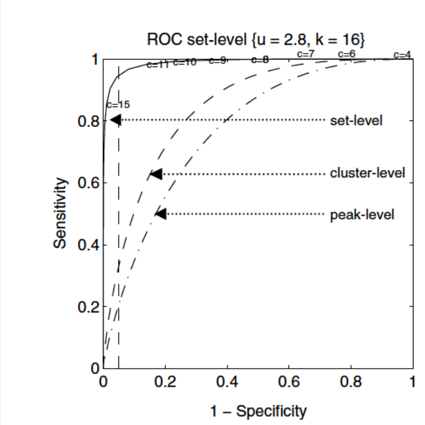
- Expected Cluster Size
  - $E(S) = E(N)/E(L)$
  - S cluster size
  - N suprathreshold volume $\lambda(\{T > u_{\text{clus}}\})$
  - L number of clusters
- $E(N) = \lambda(\Omega)\, P(\, T > u_{clus}\,)$
- $E(L) \approx E(\chi_u)$
  - Assuming no holes

Inference Levels in SPM



- **Voxel level Inference** :
  Best spatial specificity
- **Cluster level Inference** :
  Better sensitivity
  Worse spatial specificity
  The null hyp. of entire cluster is rejected
- **Set level Inference** :
  Worst spatial specificity
  Only can reject global null hypothesis

RFT: Set Level, Cluster Level and Voxel Level Threshold



Limitations of RFT

- Sufficient smoothness
  FWHM smoothness $3-4\times$ voxel size
  More like $\sim 10\times$ for low-dof T images
- Smoothness estimation
  Estimate is biased when images not sufficiently smooth
- Multivariate normality
  Virtually impossible to check
- Several layers of approximations
- Stationary required for cluster size results

2D error field with RFT and BONFERRONI correction

```
V=spm_vol();
N=64; % Set the df
DIM=[128 128 1];
sigma_2 = 0.8; % variance of white noise
PARENTDIR = '/Users/ahmetademoglu/Documents/MATLAB/MY_SPM_TOOLS/';
NEWDIR = 'RFTDATA/'
mkdir(PARENTDIR,NEWDIR) ;
for i=1:N,
P = [PARENTDIR NEWDIR 'test_' num2str(i,'%.3i') '.img' ];
V(i).fname= P;
V(i).dim=DIM;
V(i).pinfo=[1 ;0;0];
V(i).dt=[spm_type('float32') 1];
V(i).mat=eye(4,4);
% create gaussian unit variance field
x=randn(DIM(1),DIM(2),1);
x=sqrt(sigma_2)*(x-mean(x(:))/std(x(:)));   % generate white noise field with variance sigma_2
% write the data to the file
spm_write_vol(V(i),x);
end;
% read data for checking
for i=1:N,
P(i,:)=  [ PARENTDIR NEWDIR   'test_' num2str(i,'%.3i') '.hdr' ];
end;
V=spm_vol(P);
X=spm_read_vols(V);
```

2D error field with RFT and BONFERRONI correction

```
FWHM_t=3;
sx=FWHM_t;
sx    = sx/sqrt(8*log(2)); % std of gaussian filter computed from FWHM
lx=N;
Ex    = min([round(6*sx) lx]);
x     = [-Ex:Ex];
kx    = exp(-x.^2/(2*sx^2));
kx    = kx/sum(kx);
K=spm_convmtx(kx',lx,'circular');
% K=eye(size(K));  % by pass temp correlation
% % set the whitening filter
S=inv(K);
% Sigma=K*K'; % Covariance of error field SIGMA
% create  temporally correlated 2D fields  and store them in Y
for i=1:DIM(1),
for j=1:DIM(2),
T=squeeze(X(i,j,1,:));
Y(i,j,1,:)=K*T;
end;
end;
%create a box car signal and add to error field
Scale=1;
T1=zeros(DIM(1),DIM(2),1);
for i=1:10, for j=1:10,
T1(DIM(1)/4+[1:10],DIM(2)/4+[1:10])=1*Scale;
T1(2*DIM(1)/4+[1:10],2*DIM(2)/4+[1:10])=0.5*Scale;
end;end;
```

2D error field with RFT and BONFERRONI correction

```
T1=cat(4,repmat(T1,[1 1 1 N/4])*0, repmat(T1,[1 1 1 N/2])*1,  repmat(T1,[1 1 1 N/4])*0 ) ;
stem(squeeze(T1(40, 40,1,:)))
imagesc(squeeze(T1(:,:,1,32)));

% add the box car signal onto the temporally correlated 2D field
Z=Y+T1;
plot([ squeeze(T1(40,40,1,:))  squeeze(Z(40,40,1,:)) ]);
imagesc(squeeze(Z(:,:,1,32)));

% create  2D gaussian smoothed images with FWHM
FWHM=[  4 6 ];
for i=1:N,
T = squeeze(Z(:,:,1,i));
T = spm_conv(T,FWHM(1),FWHM(2));
Z1(:,:,1,i) = T/ std(T(:));
end;
imagesc(squeeze(Z1(:,:,1,32)));

Xd=[ squeeze(T1(DIM(1)/4+5,DIM(1)/4+5,1,:))  ]; % construct design matrix by using box car regressor
stem([ Xd(:,1)  ]);

Xd1=S*Xd; % prewhiten the design matrix with the inverse of the covariance matrix of temporal error
% find the residual forming matrix R
R=eye(N,N) - Xd1*pinv(Xd1) ;

V_e = eye(N) ; % error is assumed to be unit variance normal after prefiltering
v=(trace(R*V_e))^2/trace(R*V_e*R*V_e);
RV=trace(R*V_e);
```

2D error field with RFT and BONFERRONI correction

```
for i=1:DIM(1), for j=1:DIM(2),
y = squeeze(Z1(i,j,1,:)); % get the  observed time series
y = S*y; % prewhiten the observed time series
beta = pinv(Xd1)*y ; % solve the prewhitened GLM for beta
c = zeros(size(Xd1,2),1);
c(1) =1; % choose the boxcar regressor  for contrast
% compute the standart deviation of the contrast
r  = R*y; % residual error
s2=r'*r/RV; % residual error variance
s2 = s2* (c'*inv(Xd1'*Xd1)*c);
t_v(i,j,1) = c'*beta/sqrt(s2);
% residual 2D field
Z2(i,j,1,:) = (r-mean(r)) / std(r(:));
B(i,j) = c' * beta;
end;end;
% write the error field
for i=1:N,
P=[PARENTDIR NEWDIR  'test_smooth_' num2str(i,'%.3i') '.img' ];
V(i).fname= P;
V(i).dim=DIM;
V(i).pinfo=[1 ;0;0];
V(i).dt=[spm_type('float32') 1];
V(i).mat=eye(4,4);
spm_write_vol(V(i),Z2(:,:,1,i));
end;
```

2D error field with RFT and BONFERRONI correction

```
% find the RFT corrected height threshold
a= 0.05; % significance level
df= [0 v];
STAT = 'T';
[FWHM2,VRpv,R1] = spm_est_smoothness(V,V(1),[N,v]);
FWHM2
% RFT corrected threshold u
u_RFT = spm_uc_RF(a,df,STAT,R1,1)
% Bonferroni corrected threshold
u_BONF = spm_u(a/(DIM(1)*DIM(2)),df,STAT)
subplot(1,2,1);imagesc((t_v>u_RFT).*t_v);title('RFT');colormap gray;
subplot(1,2,2);imagesc((t_v>u_BONF).*t_v);title('BONFERRONI');colormap gray;
```

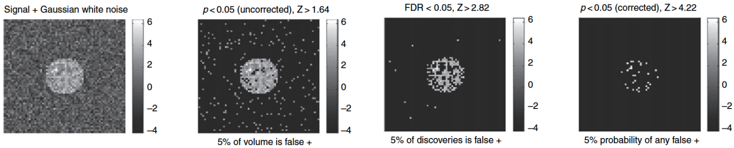
False Discovery Rate (FDR)

$$FDR = \frac{\#\ of\ False\ Positives}{All\ Positives} = \frac{V}{R}$$

R voxels declared active, V falsely

FDR Procedure

- Calculate the $p-$value of all voxels
- Sort them as $P_1 \leq P_2 \leq \ldots \leq P_N$
- Find the largest value of k for which $P_k < \alpha k/N$ to control the FDR at α.



Signal + Gaussian white noise

$p < 0.05$ (uncorrected), $Z > 1.64$

5% of volume is false +

FDR < 0.05, $Z > 2.82$

5% of discoveries is false +

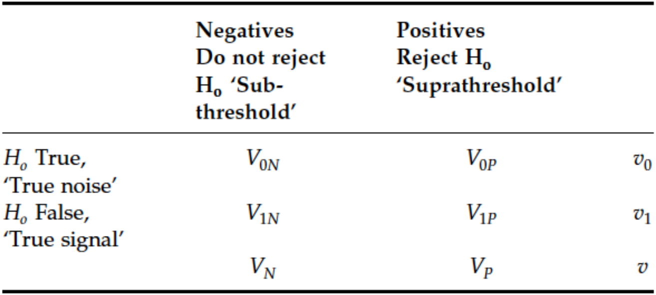
$p < 0.05$ (corrected), $Z > 4.22$

5% probability of any false +

- In FDR: False positives will be detected; we are simply controlling their rate so that they do make up no more than α of our discoveries.
- Bonferroni and RFT control the probability of ever reporting a false discovery
- FDR controls expected false discovery rate of voxels
- RFT controls the false positive rate of regions or maxima

FDR

A 0.05 FWE method has 95% confidence of producing results totally free of type I errors.
Remarkable control of false positives but with risk of false negatives (poor sensitivity)



	Negatives Do not reject $\mathrm{H_o}$ 'Sub-threshold'	**Positives Reject $\mathrm{H_o}$ 'Suprathreshold'**	
H_o True, 'True noise'	V_{0N}	V_{0P}	v_0
H_o False, 'True signal'	V_{1N}	V_{1P}	v_1
	V_N	V_P	v

FDR

Control of FWE

- Weak : A procedure that controls the chance of a FWE when there is no signal ($v = v_0$)
- Strong : A procedure that controls the chance of a FWE when there is signal ($v = v_1$)

A test with strong FWE can localize an activation, asserting that any group of voxels are falsely detected with probability at most α. A test with weak FWE is an omnibus test, and can only assert that there is some signal somewhere.

FDR

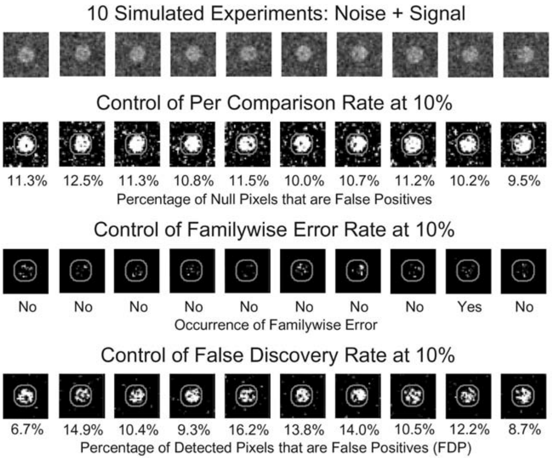
False discovery proportion FDP is a random variable varying with each statistical realization of the image;

$$FDP = \frac{V_{oP}}{V_P}$$

and $FDR = E[FDP]$
Per comparison error rate $PCE = P\{T_i > u | H_0\}$, α level of the test

- PCE $\longrightarrow$ "uncorrected"
- FWE and FDR $\longrightarrow$ "corrected"

Simulations of control of false positives with different metrics



Sensitivity ↑
False Positives ↑

Sensitivity ↓
False Positives ↓

Sensitivity ↑
False Positives ↑

Non Parametric Procedure

The lady and the tea from Fisher's *The design of experiments*, first published in 1935

Once upon a time, there was a British lady who claimed that she could tell from the taste which had been poured into the cup first, the tea or the milk. So Fisher designed an experiment to test it.

- Eight cups of tea were prepared.
- In four, the tea was poured first.
- In the other four, the milk was poured first.
- Other features of the cups of tea (size, temperature, etc.) were held constant.
- Cups were presented in a random order (critical).
- The lady tasted them, and judged.
- She knew there were four of each type.

The null hypothesis

- The null hypothesis is that the lady has no ability to taste the difference.
- If so, all possible ways of lining up the lady's judgements and the truth about the tea cups should be equally likely.
- Equally likely *because of the random order of presentation*.
- The test statistic is the number of correct judgements.
- What is the distribution of the test statistic under the null hypothesis?

Data file

```
Truth Judgement
1   tea      milk
2  milk       tea
3  milk      milk
4  milk      milk
5   tea       tea
6   tea       tea
7   tea      milk
8  milk       tea
```

- Under H_0, the reasons for the lady's judgements are unknown, except that they have nothing to do with the truth.
- Because of randomization, all $8! = 40,320$ permutations of the cups are equally likely, and each one has its own number of correct judgements.
- But there are lots of repeats.

Counting argument

- How many ways are there to choose 4 cups to put the tea in first? $\binom{8}{4} = 70$
- All are equally likely.
- Only one lines up perfectly with the lady's judgements.
- The chances of this under H_0 are $\frac{1}{70} = 0.0143 < 0.05$.
- So H_0 would be rejected at $\alpha = 0.05$ if she guessed perfectly.

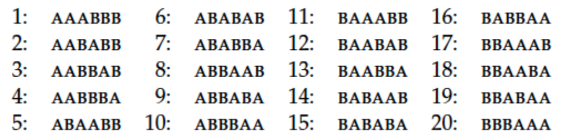
The permutation distribution

- Decide on a test statistic T.
- List the possible values of T.
- Under H_0, all ways of re-arranging the data are equally likely.
- $P(T = t)$ is proportional to the number of ways of getting the value t.

The permutation p-value is the probability of getting a value of T as extreme or more extreme as the value we observed, "extreme" meaning in a direction inconsistent with H_0.

Non Parametric Procedures

Randomization by Permutations

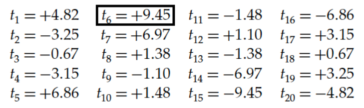


1: AAABBB
2: AABABB
3: AABBAB
4: AABBBA
5: ABAABB
6: ABABAB
7: ABABBA
8: ABBAAB
9: ABBABA
10: ABBBAA
11: BAAABB
12: BAABAB
13: BAABBA
14: BABAAB
15: BABABA
16: BABBAA
17: BBAAAB
18: BBAABA
19: BBABAA
20: BBBAAA

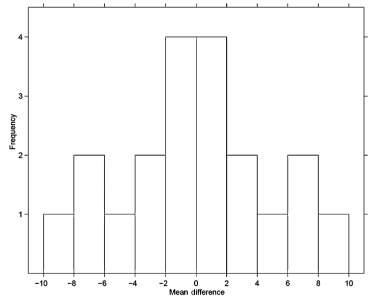
A: baseline, B: Condition $t-$statistics $= \frac{1}{3} \sum_{i=1}^{3} (A_i - B_i)$ A_i and B_i : values of i^{th} scan at a voxel. The Null Hypothesis $\mathcal{H}_0$: There is no change in scan values even the conditions are changed.

Non Parametric Procedures

Randomization by Permutations



$t_1 = +4.82$	$\boxed{t_6 = +9.45}$	$t_{11} = -1.48$	$t_{16} = -6.86$
$t_2 = -3.25$	$t_7 = +6.97$	$t_{12} = +1.10$	$t_{17} = +3.15$
$t_3 = -0.67$	$t_8 = +1.38$	$t_{13} = -1.38$	$t_{18} = +0.67$
$t_4 = -3.15$	$t_9 = -1.10$	$t_{14} = -6.97$	$t_{19} = +3.25$
$t_5 = +6.86$	$t_{10} = +1.48$	$t_{15} = -9.45$	$t_{20} = -4.82$



$t_6 = +9.45$ gives the largest mean difference of all the possible labellings with a p value is $1/20 = 0.05$.

There is significant evidence against the null hypothesis of no activation at this voxel at level $\alpha = 0.05$.

Multiple Testing Problem (FWE)

Single threshold test

- for each possible relabelling $i = 1, \ldots, N$, note the maximal statistic t_i^{max}, the maximum of the voxel statistics for relabelling $i : t_i^{max} = max\{t_i\}_{i=1}^{N}$
- This gives the permutation distribution for T^{max}, the maximal statistic. The critical threshold is the $c + 1$ largest member of the permutation distribution for T^{max} , where $c = \alpha N$ rounded down.
- Voxels with statistics exceeding this threshold exhibit evidence against the corresponding voxel hypotheses at level α.

The corresponding corrected p-value for each voxel is the proportion of the permutation distribution for the maximal statistic that is greater than or equal to the voxel statistic.

Multiple Testing Problem (FWE)

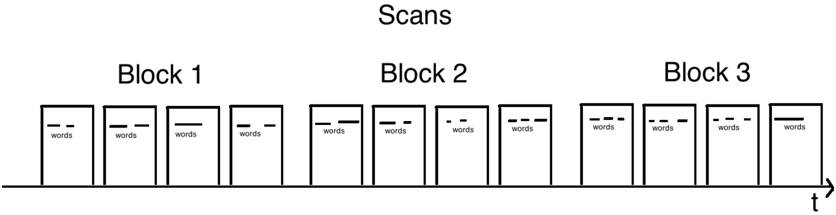
Suprathreshold cluster tests

- For the statistic image corresponding to each possible relabelling, threshold the image at a *primary threshold*, and note the size of the largest suprathreshold cluster above it.
- The critical suprathreshold cluster size for this primary threshold is the $[\alpha N + 1]^{th}$ largest member of this permutation distribution.
- Corrected p values for each suprathreshold cluster in the image are obtained by comparing their size to the permutation distribution.
- With a low threshold, large suprathreshold clusters are to be expected, so intense focal ‘signals’ will be missed.
- At higher thresholds these focal activations will be detected, but lower intensity diffuse ‘signals’ may go undetected below the primary threshold (more power at the expense of reduced localization).

Permutation Analysis

1. **Null hypothesis** Specify the null hypothesis.
2. **Exchangeability** Specify exchangeability of observations under the null hypothesis.
3. **Statistic** Specify the statistic of interest, usually broken down into specifying a voxel-level statistic and a summary statistic.
4. **Relabellings** Determine all possible relabellings given the exchangeability scheme under the null hypothesis.
5. **Permutation distribution** Calculate the value of the statistic for each relabelling, building the permutation distribution.
6. **Significance** Use the permutation distribution to determine significance of correct labelling and threshold for statistic image.

Example: Single-subject PET

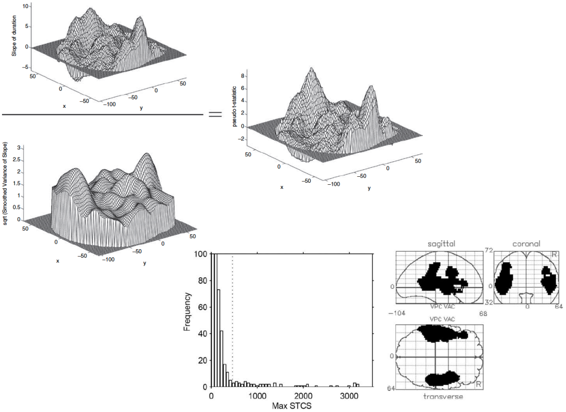


For each scan, the amount of radioactivity entering the brain during the events is calculated to derive a score reflecting the contribution of the events to the image.
The design matrix : [Grand mean ; slope of score ; gCBF]

Voxel Level Statistics

$12 - 3 = 9$ degrees of freedom for the $t-$field.
Permutations : $4! \times 4! \times 4! = 13824$ *too big a number*
Monte Carlo test choosing a random 999 relabelings + the actual labeling.
Primary threshold = 3
1000 relabellings, $1000 \times 0.95 = 950$, so the 950^{th} largest max cluster size will be our significance threshold.

Results



Only five relabellings yield max STCS equal to or larger than 3101, so the p-value for the experiment is $5/1000 = 0.005$. Two clusters have 3101 and 1716 voxels in size, with p values of 0.005 and 0.015, respectively.

RANDOMIZATION APPROACHES

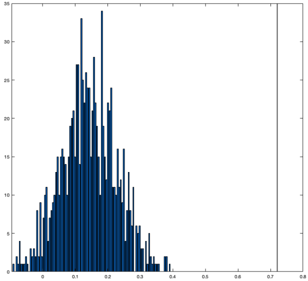
- PERMUTATION
- BOOTSTRAPPING
- JACKNIFING
- N-FOLD CROSS VALIDATION

Permutation: Statistical significance of a parameter is assessed by means of sampling without replacement to reassign the order of conditions for each observation. It is used to show if an effect represented by the parameter is sufficiently strong to be different from random noise, in a statistical sense.
Bootstrapping: The standard error estimates of the saliences from the bootstrap tests are used to assess the reliability of the nonzero saliences on significant LVs.

AR(1) Model : $x(n) = 0.8x(n) + w(n)$ where $w(n) \sim \mathcal{N}(0, 1)$

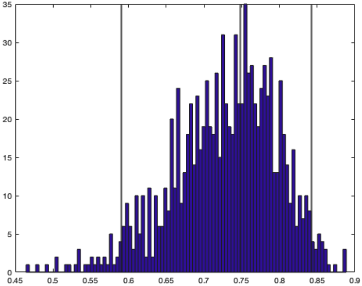
```
clear all
beta = 0.8; N=100; e=randn(N,1);
e= (e-mean(e))/std(e);
x=filter(1,[1 -beta],e);
beta_h = x(2:end)\x(1:end-1);
% PERMUTATION : REORDERING THE SAMPLES
for i=1:1000,
Ind= randperm(N,N);
x_r=x(Ind);
beta_r(i)= x_r(2:end)\x_r(1:end-1);
end;
[c,d]=hist(beta_r,50);
subplot(1,2,1)
bar(d,c) ; xline( beta_h,LineWidth=2 );
% BOOTSTRAPPING : RANDOM SAMPLING WITH REPLACEMENT
for i=1:1000,
Ind= randi([1 N],N,1);
x_r=filter(1,[1 -beta_h],e(Ind));
beta_r(i)= x_r(2:end)\x_r(1:end-1);
end;
[p1]=sort(beta_r);
a=cumsum(p1);a=a/max(a);
subplot(1,2,2)
p1([min(find(a>0.025)) min(find(a>(1-0.025))) ])
hist(beta_r,50);xline([ p1([min(find(a>0.025)) min(find(a>(1-0.025)))]) beta_h],LineWidth=2 );
```

PERMUTATION



Permutation test asseses if the
the estimated parameter
reflects an effect different from the one
generated by random noise.

BOOTSTRAPPING

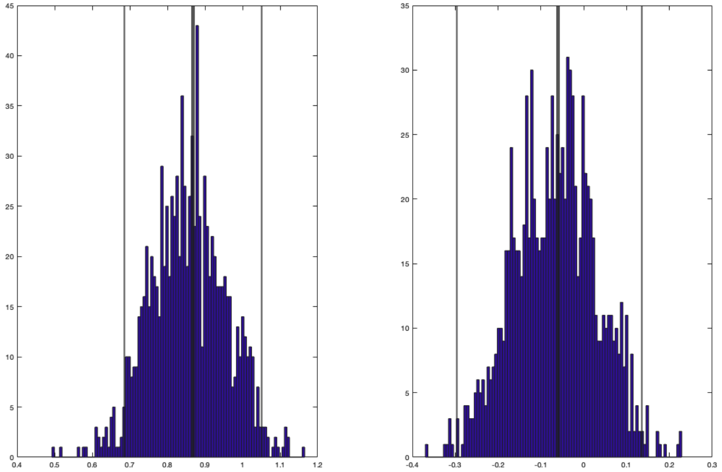


The bootstrap test is used to assess the reliability of a nonzero parameter.

Fitting an AR(2) Model to $x(n) = 0.8x(n) + w(n)$ where $w(n) \sim \mathcal{N}(0,1)$

```
beta = 0.8;
N=100;
e=randn(N,1);
e= (e-mean(e))/std(e);
x=filter(1,[1 -beta],e);

X =[ [1; x(1:end-1)] [1 ; 0 ; x(1:end-2)] ] ;
a2 = X\x
for i=1:1000,
Ind= randi([1 N],N,1);
x_r=filter(1,[1 -a2'],e(Ind));
X =[ [1; x_r(1:end-1)] [1 ; 0 ; x_r(1:end-2)] ] ;
beta_2(:,i)=X\x_r;
end;
hist(beta_2(1,:),100);xline(a2(1),LineWidth=2 );
pause
hist(beta_2(2,:),100);xline(a2(2),LineWidth=2 );
std(beta_2')/sqrt(1000)
```



$a(1)$ significantly different from 0 $a(2)$ not so different from 0

No strong evidence for that AR(1) model does not give a resonable representation of the data.

Jacknifing is a bootstrap technique which picks up all the samples but one to estimate a parameter. It is simpler that bootstrapping and approximates it linearly, though it does not use the data in full sense.
For instance, if we have a sample data $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and an estimator $\hat{\theta} = s(\mathbf{x})$, *we leave out one observation at a time* to generate *jacknife samples*,

$$\mathbf{x}_{(i)} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

The estimated $(i)^{th}$ jacknife sample is

$$\hat{\theta}_{(i)} = s(\mathbf{x}_{(i)})$$

Cross validation is splitting the data in two parts with a certain ratio and use one part for fitting the other for estimation.