Homework 3 - Model answer

Background:

Listen to the lecture on Data formats (Voxel and affine transformation matrices, nifti formats) and group analysis (Group Analysis, Anatomical Normalization, Multiple Testing problem, Randomization Tests).
Also, at least skim over the Eklund et al. (PNAS, 2016) paper to get a feel for the current debate in the field.

Data homework:

Your task is to conduct a group analysis from a subset of the task-based Human connectome project (https://www.humanconnectome.org/study/hcp-young-adult/project-protocol/task-fmri). For this study you are given the contrast between reward vs. punishment from the gabling task for N=25 subjects (see Barch et al. for meaning of this contrast). The zip-files contains the data from all these 25 subjects in NIFTI format + the anatomical normalization template in MNI space.

`GA_RewardvsPunish_xx.nii:`
Contrast from the gambling task reward-punishment for s01-s25

`MNI152_T1_2mm.nii:`
Anatomical template image in MNI space

`mask_GA.nii:`
Mask for the gambling task - marking all within-brain voxels

1. Load and plot horizontal slice z=0, z=-20, and z=40 of the MNI standard template. To translate the MNI coordinates to voxel coordinates in the image, you need to consider the affine transformation matrix in the header of the image (see lectures of data format).

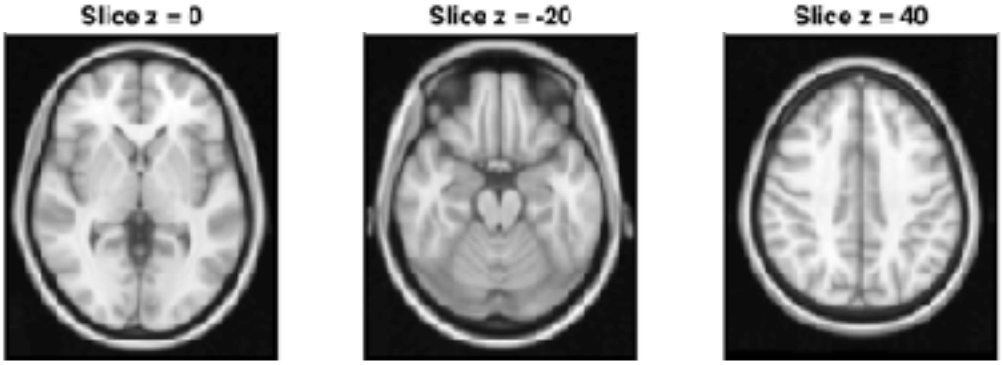

Figure 1. Average anatomical image shown in three axial slices.

We can see that some anatomical details are lost after aligning and averaging brains, especially on the cortical level.

2. Calculate a group t-test for the contrast Reward>Punishment and of Punishment> Reward. For each of the contrasts, what is the peak t-value and at what MNI coordinate

can they be found? What is the uncorrected p-value? Plot the slice with the maximal t-value for each contrast. Only consider voxels where binary_mask.nii is 1.

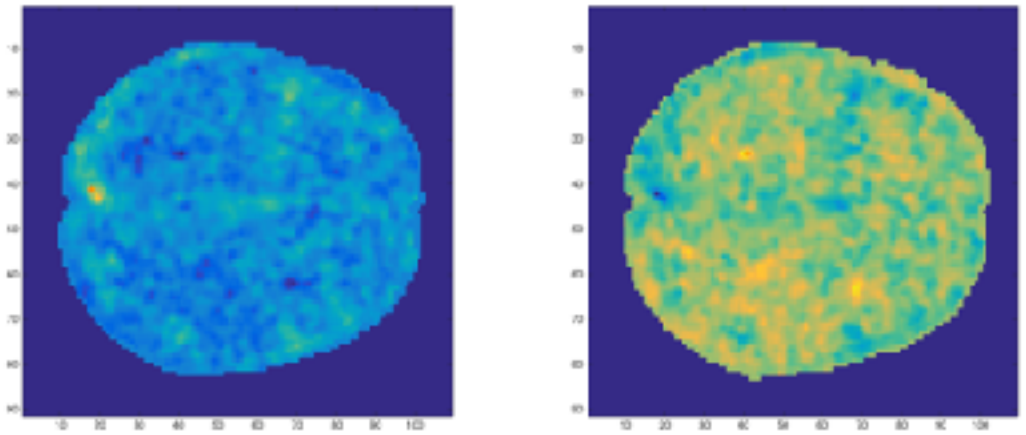

Fig.2. Slice with maximal t-value for Reward>Punishment (left), and Punishment>Reward (right). The peak t-values are indicated with red crosses.

Contrast 1: t(24)=10.221, p=1.599e-10 i:41,j:17,k:45 x:10,y:-92,z:16
Contrast 2: t(24)=4.299, p=1.23e-4 i:33,j:41,k:44 x:26,y:-46,z:14
The i,j,k coordinates are in voxel coordinates.
The x,y,z coordinates are in world (MNI coordinates).

Voxels with maximal t-values are indicated with a red cross.

3. Is the peak t-value for each contrast significant when correcting for multiple tests using Bonferroni correction over the number of voxels considered (within the mask)? For each hypothesis use a one-sided test.

Number of voxels: 265690
Corrected p: 0.0000001882
Crticial T-value: 6.915

Contrast 1 survives Bonferroni correction, but contrast 2 doesn't.

For details see attached code (case '3_bernulli');

4. Obtain a Null-distribution of the peak t-value in an image through a randomization test. For each contrast, flip the sign of the contrast image for each subject separately. Try at least 1000 ways of flipping the data randomly.
a) What is your Null-hypothesis for each test and why is the way you conduct the randomization test justified under this null-hypothesis (i.e. what is exchangeable?).
b) Plot a histogram of the Null-distribution and mark the value of the real maximal t-values in the histogram with a vertical line.
c) Calculate and report the corrected p-value by determining from the randomized t-distribution how likely it is that the peak t-value (or a larger value) has occurred under the null-hypothesis?

The null hypothesis is that there is no systematic activity across subjects, so the contrasts labels are exchangeable.

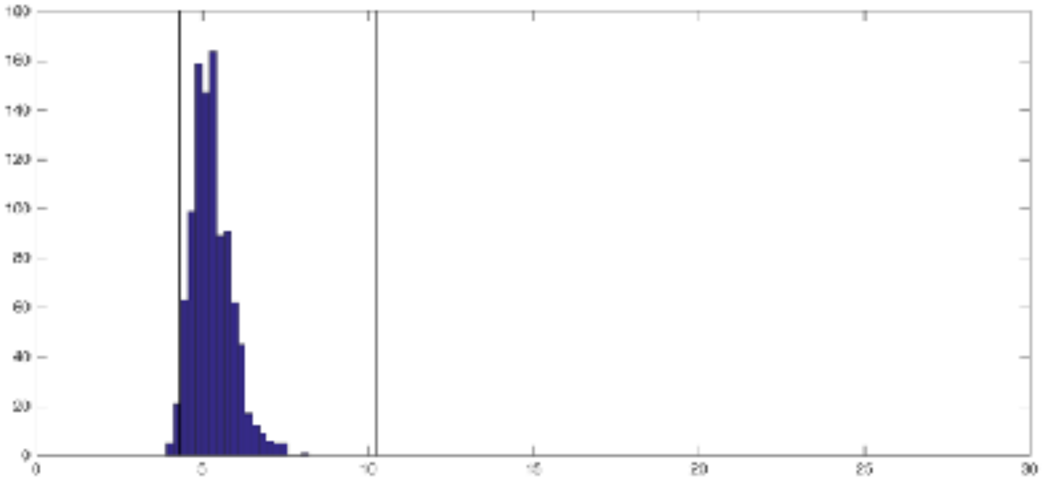

Fig. 3. Histogram with peak t-values for each of the 1000 permutations. The vertical lines indicates the measured peak t-value - on the left side for Punishment>Reward, and on the right side Reward>Punishment.

The corrected p-values are calculated as the chance of getting a peak T-value as observed, or higher:

t(24)=10.221	p_corr<0.001
t(24)=4.299	p_corr=0.98

Again, the first contrast is significant while the second is not.

Example code:

```
function varargout = homework3(what,varargin)
numSubj = 25;

switch (what)
    case '1_anatomical'
        % read in the image data
        V=spm_vol('data/MNI152_T1_2mm.nii');
        X=spm_read_vols(V);

        % Calculate the 3 slices to plot: MNI-coordinates
        z1=[0 0 0 1]';
```

```
   z2=[0 0 -20 1]';
   z3=[0 0 40 1]';
   % Transform into Voxel
   x1 = inv(V.mat)*z1;
   x2= inv(V.mat)*z2;
   x3 = inv(V.mat)*z3;


   % Present 3 axial slices
   figure(1);
   set(gcf,'PaperPosition',[2 2 4 10]);
   subplot(1,3,1);
   imagesc(X(:,:,x1(3))');
   title(sprintf('Slice z = %d',z1(3)));
   set(gca,'YDir','normal','XTick',[],'YTick',[]);
   axis equal;

   subplot(1,3,2);
   imagesc(X(:,:,x2(3))');
   title(sprintf('Slice z = %d',z2(3)));
   set(gca,'YDir','normal','XTick',[],'YTick',[]);
   axis equal;

   subplot(1,3,3);
   imagesc(X(:,:,x3(3))');
   set(gca,'YDir','normal','XTick',[],'YTick',[]);
   title(sprintf('Slice z = %d',z3(3)));
   axis equal;
   colormap(gray);


case '2_calcT'
   V=spm_vol('data/mask_GA.nii');
   MASK = spm_read_vols(V);
   for s=1:numSubj
        V=spm_vol(fullfile('data',sprintf('GA_REWARDvsPUNISH_%2.2d.nii',s)));
        X(:,:,:,s)=spm_read_vols(V);
   end;
   STDERR = std(X,0,4)*sqrt(1/numSubj); % Standard error of the mean

   for c=1:2
      switch (c)
         case 1
            T=mean(X,4)./STDERR;           % T - value
         case 2
            T=-mean(X,4)./STDERR;           % inverted contrast
      end;
      T=T.*(MASK>0);  % Mask

      [maxT(c),indx]=max(T(:));
      [i,j,k]=ind2sub(V.dim,indx);
      % Translate into world coordinates (see lecture on affine transformations)
```

```
        wcoord = V.mat*[i;j;k;1];
        fprintf('Contrast %d: T(24)=%3.3f p=%f  i:%d j:%d k:%d x:%2.2f   y:%2.2f   z:
       %2.2f\n',c,maxT(c),1-
       tcdf(maxT(c),numSubj-1),i,j,k,wcoord(1),wcoord(2),wcoord(3));
        subplot(1,2,c);
        imagesc(T(:,:,k));
    end;
    varargout={maxT};
case '3_bernulli'          % Now calculate the bernulli correction. (% Question 3)
    V=spm_vol('data/mask_GA.nii');
    MASK = spm_read_vols(V);
    inMask = find(MASK); % Find indices
    FWE = 0.05;     % Target Family-wise error
    numVoxel = length(inMask);
    correctedP = FWE / numVoxel;  % Corrected P-value
    Tcrit    = tinv(1-correctedP,numSubj-1); % Crtitical T-value for numSubj-1 df
    fprintf('Number of Voxels: %d\n',numVoxel);
    fprintf('Corrected p: %2.10f\n',correctedP);
    fprintf('Crticial T-value: %2.3f\n',Tcrit);
case '4_permute'
    RealmaxT=homework3('2_calcT');
    V=spm_vol('data/mask_GA.nii');
    MASK = spm_read_vols(V);
    numSubj = 25;
    numIter = 100;
    for s=1:numSubj
          V=spm_vol(fullfile('data',sprintf('GA_REWARDvsPUNISH_%2.2d.nii',s)));
          X(:,:,:,s)=spm_read_vols(V);
    end;

    inMask = find(MASK); % Find indices

    % Now build up a sample distribution, here I do all 128
    %ways of fliping the signs randomly.
    for s=1:numIter
        fprintf('.');
        flipsign=unidrnd(2,numSubj,1)*2-3;
        S = bsxfun(@times,X,permute(flipsign,[4 3 2 1]));  % Flip them by the sign
        STDERR = std(S,0,4)*sqrt(1/numSubj); % Standard error of the mean
        T=mean(S,4)./STDERR;           % T - value
        maxT(s)=max(T(inMask));  % find max
    end;
    fprintf('\n');

    % Now make histogram with real data
    hist(maxT,20);
    set(gca,'XLim',[0 30]);
    drawline(RealmaxT);
    for c=1:2
        p_corr(c) = sum(maxT>=RealmaxT(c))./numIter;   % Correct p-value is the
       porportion of samples higher than real one
        fprintf('T(24)=%3.3f  P_corr = %3.3f\n',RealmaxT(c),p_corr(c));
```

```
   end;
end;
```