
applyContrast.m

```
function [con] = applyContrast(beta1,beta2,residuals)
    %Computes contrast between two betas.
    %First computes the difference, then does a significance test
    using the
    %sum of squared errors resulting t values for a voxels. Any
    %insignificant voxels are then set to 0 so they will not show up
    in any
    %visual representation of the data.

    %calc absolute difference between beta1 and beta2
    con = abs(beta1 - beta2);

    %for later use
    xm = size(residuals,1);
    ym = size(residuals,2);
    zm = size(residuals,3);
    nscans = size(residuals, 4);

    %reshaping for easier calculations.
    residuals = reshape(residuals, [xm * ym * zm, nscans]);
    squaredsd = zeros(xm * ym * zm, 1, 'single');

    %Calc sum of squared errors per voxel
    for i = 1:size(residuals,1)
        squaredsd(i) = sum(residuals(i,:).^2, 2)/(nscans-1);
    end
    squaredsd = reshape(squaredsd, [xm, ym, zm, 1]);

    %Calc T value per voxel
    Tk = con./sqrt(squaredsd./nscans);

    %Set voxels with to low T value to 0.
    threshold = 10;
    Tk(abs(Tk)<threshold) = 0;
    beta1(Tk==0) = 0;
    beta2(Tk==0) = 0;

    %Calc difference of significant voxels.
    con = abs(beta1-beta2);
end
```

Published with MATLAB® R2017b