
Table of Contents

.....	1
Create Analysis Report - HPF	1
Initialization	1
Filter Table	1
Analysis: All trials	1
Analysis: S, D1, D2 trials	2
Analysis: T M N trials	2
Analysis: T0 M0 N0 trials	2
===== Local Functions =====	6
===== FilterTableScores =====	7
Build filter rm index	7
===== SortTableScores =====	8

```
function CreateReport_HPComp
```

Create Analysis Report - HPF

```
% Overview
% -----
%
% CalcMeans. Calculate mean betas and ResMS.nii.
% ChooseModels. Decide which models to compare
% PlotResults. Filter date by Choose models and plot results
% GetModelID
```

Initialization

```
close all;
C = Study_greco2;
T = load(fullfile(C.dir.tables, 'greco_tables.mat'));
C.subjects.subj2run= C.subjects.subj2inc;

filter.Names = {'spec_name', 'struct_reg'};
filter.Vals = { 'STris_20', 0};
sorter = {'hpf'};
```

Filter Table

```
newT = T;
newT.rs = FilterTableScores(newT, filter);
newT.rs = SortTableScores(newT, sorter);
```

Analysis: All trials

```
AnalysisAll(C, newT, 'uni', filter) %Uni / All
AnalysisAll(C, newT, 'rs', filter) %RS / All
```

```

Error using load
Unable to read file '/Users/jdstokes/Studies/greco/analysis/scores/r000000

Error in GetImgScores (line 21)
    S = load(fullfile(inDir,fName));

Error in CreateReport_HPFcomp>AnalysisAll (line 78)
    newScores = GetImgScores(C,CI_comp,scoreRow.rs_ID);

Error in CreateReport_HPFcomp (line 30)
AnalysisAll(C,newT,'uni',filter) %Uni / All

```

Analysis: S, D1, D2 trials

```

AnalysisSD1D2(C,newT,'uni',filter) %Uni / S D1 D2
AnalysisSD1D2(C,newT,'rs',filter) %RS / S D1 D2

```

Analysis: T M N trials

```

AnalysisTMN(C,newT,'uni',filter) %Uni / T M N
AnalysisTMN(C,newT,'rs',filter) %RS / T M N

```

Analysis: T0 M0 N0 trials

```

AnalysisT0M0N0(C,newT,'uni',filter) %Uni / T M N
AnalysisT0M0N0(C,newT,'rs',filter) %RS / T M N

```

```
end
```

```

function AnalysisAll(C,T,mode,filter)
rois ={'ash_right_CA1.nii';'ash_right_DG.nii'};
C = SetupROIs(rois,C);

```

```

switch mode
    case 'uni'
        C = Configure4uni(C);
    case 'rs'
        C = Configure4rs(C);
    otherwise
        error('incorrect mode argument');
end

```

```

CI_comp{1}.input{1}.var = {'tt_code'};
CI_comp{1}.input{1}.val = [0 1 2];

```

```

subjects = C.subjects.subj2run;
numScores = height(T.rs);

```

```

data = [];
legend = {};
hpfList = [];
modelList = {};
for cS = 1:numScores
    scoreRow = T.rs(cS,:);
    modelRow = GetModelRow(scoreRow,T);

    newScores = GetImgScores(C,CI_comp,scoreRow.rs_ID);
    data = [data, newScores];
    legend = [legend;num2str(modelRow.hpf)];
    hpfList = [hpfList,modelRow.hpf];
    modelList = [modelList;scoreRow.rs_ID];

end

labels = FixStrings(rois,{' .nii', 'ash_', '_'},{' ', ' ', ' '});
figure
StatBarPlot(data, 'within', labels, mode, 0, legend);
title([scoreRow.rs_ID, ' AnalysisAll, HPF ', mode]);
end

function AnalysisSD1D2(C,T,mode,filter)
    rois = {'ash_right_CA1.nii'; 'ash_right_DG.nii'};
    C = SetupROIs(rois,C);
    switch mode
        case 'uni'
            C = Configure4uni(C);
        case 'rs'
            C = Configure4rs(C);
        otherwise
            error('incorrect mode argument');
    end

    CI_comp{1}.input{1}.var = {'tt_code'};
    CI_comp{1}.input{1}.val = [0];
    CI_comp{2}.input{1}.var = {'tt_code'};
    CI_comp{2}.input{1}.val = [1];
    CI_comp{3}.input{1}.var = {'tt_code'};
    CI_comp{3}.input{1}.val = [2];

    legend = Input2Str(CI_comp);

    spec_name = 'STris_20';
    subjects = C.subjects.subj2run;
    numRows = height(T.rs);

    labels = FixStrings(rois,{' .nii', 'ash_', '_'},{' ', ' ', ' '});

    data = [];
    for cS = 1:numRows

```

```

        scoreRow = T.rs(cS,:);
        modelRow = GetModelRow(scoreRow,T);

        try
            newScores = GetImgScores(C,CI_comp,scoreRow.rs_ID);
            data = newScores;
            figure
            StatBarPlot(data,'within',labels,mode,0,legend);
            title([scoreRow.rs_ID, ' AnalysisSD1D2, HPF:', num2str(modelRow.hpf
end
end

end

function AnalysisTMN(C,T,mode,filter)
    rois ={'ash_right_CA1.nii','ash_right_DG.nii'};
    C = SetupROIs(rois,C);
    switch mode
        case 'uni'
            C = Configure4uni(C);
        case 'rs'
            C = Configure4rs(C);
        otherwise
            error('incorrect mode argument');
    end

    CI_comp{1}.input{1}.var = {'cityTargC'};
    CI_comp{1}.input{1}.val = {'T'};

    CI_comp{2}.input{1}.var = {'cityTargC'};
    CI_comp{2}.input{1}.val = {'M'};

    CI_comp{3}.input{1}.var = {'cityTargC'};
    CI_comp{3}.input{1}.val = {'N'};
    legend = Input2Str(CI_comp);

    spec_name = 'STris_20';
    subjects = C.subjects.subj2run;
    numScores = height(T.rs);

    labels = FixStrings(rois,{' .nii', 'ash_', '_'},{' ', ' ', ' '});

    data =[];
    for cS = 1:numScores
        scoreRow = T.rs(cS,:);
        modelRow = GetModelRow(scoreRow,T);

```

```

        try
            newScores = GetImgScores(C,CI_comp,scoreRow.rs_ID);
            data = newScores;
            figure
            StatBarPlot(data,'within',labels,mode,0,legend);
            title([scoreRow.rs_ID, ' Mean AnalysisTMN, HPF:', num2str(modelRow.hpf)]);
        end

    end

end
end

function AnalysisTOMON0(C,T,mode,filter)
    rois ={'ash_right_CA1.nii','ash_right_DG.nii'};
    C = SetupROIs(rois,C);
    switch mode
        case 'uni'
            C = Configure4uni(C);
        case 'rs'
            C = Configure4rs(C);
        otherwise
            error('incorrect mode argument');
    end

    CI_comp{1}.input{1}.var = {'cityTargC'};
    CI_comp{1}.input{1}.val = {'T'};
    CI_comp{1}.input{2}.var = {'tt_code'};
    CI_comp{1}.input{2}.val = [0];

    CI_comp{2}.input{1}.var = {'cityTargC'};
    CI_comp{2}.input{1}.val = {'M'};
    CI_comp{2}.input{2}.var = {'tt_code'};
    CI_comp{2}.input{2}.val = [0];

    CI_comp{3}.input{1}.var = {'cityTargC'};
    CI_comp{3}.input{1}.val = {'N'};
    CI_comp{3}.input{2}.var = {'tt_code'};
    CI_comp{3}.input{2}.val = [0];

    legend = Input2Str(CI_comp);

    spec_name = 'STris_20';
    subjects = C.subjects.subj2run;
    numScores = height(T.rs);

    labels = FixStrings(rois,{' .nii', 'ash_', '_'},{' ', ' ', ' '});

    data =[];
    for cS = 1:numScores
        scoreRow = T.rs(cS,:);

```

```

modelRow = GetModelRow(scoreRow,T);

try
    newScores = GetImgScores(C,CI_comp,scoreRow.rs_ID);
    data = newScores;
    figure
    StatBarPlot(data,'within',labels,mode,0,legend);
    title([scoreRow.rs_ID, ' Mean AnalysisT0M0N0, HPF:', num2str(modelRow.hpf)]
end

end
end

```

===== Local Functions

```

function tts = Input2Str(comp)

for i = 1:length(comp)
    s = '';
    input=comp{i}.input;
    for j = 1: length(input)
        if ~isempty(input{j})
            s = [s,input{j}.var{1}];
            if isnumeric(input{j}.val) || islogical(input{j}.val)
                s = [s,'_',num2str(input{j}.val),'_'];
            elseif iscell(input{j}.val)
                s = [s,'_', input{j}.val{:},'_'];
            end
        end
    end
    tts{i} = s(1:end-1);
end
end

```

```

function C = SetupROIs(rois,C)
C.masks.mask2inc = zeros(1,length(C.masks.maskAll));
for j = 1: length(rois)
    C.masks.mask2inc(strcmp(rois{j},C.masks.maskAll)) = 1;
end
C.masks.mask2inc = logical(C.masks.mask2inc);
end

```

```

function modelRow = GetModelRow(score_row,T)
betaIDCell = cellstr(T.betas.beta_ID);
betaRow = T.betas(strcmp(score_row.beta_ID,betaIDCell),:);
modelIDCell = cellstr(T.model.model_ID);
modelRow = T.model(strcmp(betaRow.model_ID,modelIDCell),:);

```

```
end
```

```
function C = Configure4uni(C)
C.scores.mode = 'uni';
C.scores.score_type = 'uni_m';
C.scores.OL_type = 'global_IQR';
```

```
end
function C = Configure4rs(C)
C.scores.mode = 'rs_pair';
C.scores.score_type = 'rsz';
C.scores.OL_type = 'global_IQR';
```

```
end
```

===== FilterTableScores =====

```
function newTable = FilterTableScores(T,F)

% Check inputs
if (length(F.Names) ~= length(F.Vals)) || ...
    (~iscell(F.Names) && ~iscell(F.Vals))
    error('colNames and colVals must be equal length cell arrays');
end

numScores = height(T.rs);
rmIndex = [];
for cS = 1:numScores

    scoreRow = T.rs(cS,:);
    modelRow = GetModelRow(scoreRow,T);
```

Build filter rm index

```
for curN = 1:length(F.Names)
    switch class(F.Vals{curN})
        case 'char'
            if ~any(strfind(modelRow.(F.Names{curN}),F.Vals{curN}))
                rmIndex = [rmIndex; cS];
            end
        case 'double'
            if modelRow.(F.Names{curN}) ~= F.Vals{curN}
                rmIndex = [rmIndex; cS];
            end
    end
end
```

```
end
```

```
newTable = T.rs;  
newTable(rmIndex,:) = [];
```

```
end
```

===== SortTableScores =====

```
function newTable = SortTableScores(T,S)  
  
numScores = height(T.rs);  
newTable = T.rs;  
  
for curS = 1:length(S)  
for cS = 1:numScores  
scoreRow = newTable(cS,:);  
modelRow = GetModelRow(scoreRow,T);  
sorterValues(cS,1) = modelRow.(S{curS});  
end  
  
[Y,I]= sort(sorterValues);  
  
newTable = newTable(I,:);  
clear sorterValues  
end  
end
```

Published with MATLAB® R2014a