

# Network Flexibility and Reinforcement Learning

Raphael Gerraty

2015-2016

Example scripts for running network preprocessing and analysis functions contained here. See paper for details when it comes out.

## Preprocessing

Need to add

## Running nonlinear registration with FNIRT

Run this after confound regression. Make sure `fsl_anat` has been run on each structural image first. Bash script.

```
for i in /data/engine/rgerraty/learn_dyncon/4*/Learn*; do
    flirt -ref ${i}/../structural/mprage.anat/T1_biascorr_brain.nii.gz\
    -in ${i}/reg/example_func.nii.gz -omat ${i}/reg/example_func2highres.mat;
    echo warping ${i};
    applywarp --ref=$FSLDIR/data/standard/MNI152_T1_2mm.nii.gz\
    --in=${i}/36par+spikes.feats/stats/res4d.nii.gz\
    --out=${i}/36par+spikes.feats/stats/res4d_std.nii.gz\
    --warp=${i}/../structural/mprage.anat/T1_to_MNI_nonlin_field.nii.gz\
    --premat=${i}/reg/example_func2highres.mat;
done
```

## Extracting time courses

Input a folder of ROIs and preprocessed 4D data. Bash script.

```
for i in /data/engine/rgerraty/learn_dyncon/4*/Learn?_PEprior.feats/36par+spikes.feats/;
do
    bash ~/GitHub/rl_flexibility/extract_ROIs.sh ${i}/stats/res4d_std.nii.gz ~/Harvard-Oxford;
done
```

## Calculate coherence matrices for each time window

In MATLAB. Input ROI timecourses for each block, number of windows per block in TRs, and minimum/maximum frequency in Hz.

```

addpath ~/GitHub/rl_flexibility
[a,b]=system('ls -d /data/engine/rgerraty/learn_dyncon/4*/Learn?_PEprior.feat/36par+spikes.
c=strread(b,'%s');

for i=1:size(c,1)
    filename=char(strcat(c(i),'/all_rois.txt'))
    conn_cell=coherence_by_block(filename,25,.5,.06,.12);
    save(char(strcat(c(i),'/conn_cells')), 'conn_cell')
end

```

### Run multi-slice community detection and flexibility statistics

Input coherence matrix for each block. Also need number of blocks, resolution and coupling parameters. In Matlab

```

%%%%%%%%pretty hacky, remember to fix
addpath ~/GitHub/rl_flexibility
addpath ~/scripts/MATLAB/GenLouvain_for_Raphael/
addpath ~/scripts/MATLAB/Bassett_Code/
[a,b]=system('ls -d /data/engine/rgerraty/learn_dyncon/4*/Learn?_PEprior.feat/36par+spikes.
c=strread(b,'%s');
numruns=4
k=1;
for j=1:size(c,1)/numruns
    c(k)
    conn_cell_cat=[];
    for i=1:numruns
        load(strcat(char(c(k-1+i)), '/conn_cells'))
        conn_cell_cat=cat(3,conn_cell_cat,conn_cell)
    end
    [a_mat,flex]=network_diags(conn_cell_cat,4,100,1,1.1813)
    save(char(strcat(c(k), '/../../../../../a_mat')), 'a_mat')
    save(char(strcat(c(k), '/../../../../../flex')), 'flex')
    k=k+numruns;
end

```

### Pull flexibility statistics

For plotting and analysis. Matlab.

```

[a,b]=system('ls -d /data/engine/rgerraty/learn_dyncon/4*/flex.mat');
c=strread(b,'%s');
flex_cat=[];
for j=1:size(c,1)
    load(char(c(j)))

```

```
flex_cat=cat(3,flex_cat,flex)
end
plot(squeeze(mean(flex_cat)))
```