

**Design: give\_me\_stimuli.s**

TS1\_analy.1D, TS2\_analy .1D,  
TS3\_analy.1D, TS4\_analy.1D

TSall\_analy.1D,  
concat.1D

E-Prime Programming

**Collect Data**

analyze\_ts\_fwhm4.s

register.s: register many image files to AFNI format.

3dTshift: slice timing adjustment

3dvolreg: motion correction

3dTcat: concatenate all fMRI data together

3dmerge: spatial blurring

mask generation: identify brain region with 3dcalc or 3dAutomask

Link design matrix (TSall\_analy.1D, concat.1D)

**3dDeconvolve (deconv.s)**: Compare fMRI image data with design matrix

decon\_TS (the "bucket")

PerSigCh.s: calculate % signal change

FLY\_group\_ANOVA2\_analyze.s: group analysis from all subjects

MonteCarlo\_moreIterfwhm4.s  
(AlphaSim)

gen\_clusterP5103.s (3dclust and 3dmerge): cluster analysis

**Final results**

```
#!/bin/tcsh

# To design the experiment
# David C. Zhu, Spring, 2007

# 4 runs

# experiment parameters
set ts          = 164          # length of timeseries
set stim        = 3           # number of input stimuli
set num_on      = 32          # time points ON per stimulus type (between 0 and $ts/3)

# execution parameters
set iterations  = 200         # number of iterations
set seed_1      = 1628873     # initial random seed for 1st run
set seed_2      = 2734891     # initial random seed for 2nd run
set seed_3      = 5734792     # initial random seed for 3rd run
set seed_4      = 6724890     # initial random seed for 4nd run
set outdir      = stim_TSall_results
set LCfile      = $outdir/LC_sums

rm -f -R $outdir

# -----
# make sure $outdir exists

if ( ! -d $outdir ) then
  mkdir $outdir
  if ( $status ) then
    echo "failure, cannot create output directory, $outdir"
    exit
  endif
endif

# create empty LC file
echo -n "" > $LCfile

echo -n "iteration: 000"

# -----
# run the test many times
```

```
foreach iter (`count -digits 3 1 $iterations`)

    # make some other random seed for Run 1

    @ seed_1 = $seed_1 + 1

    # create random order stim files

    RSFgen -nt ${ts} -num_stimts ${stim} -nreps 1 ${num_on} -nreps 2 ${num_on} -nreps 3 ${num_on} -seed
    ${seed_1} \
    -prefix RSF.stim.run1.${iter}. >& /dev/null

    # make some other random seed for Run 2

    @ seed_2 = $seed_2 + 1

    # create random order stim files

    RSFgen -nt ${ts} -num_stimts ${stim} -nreps 1 ${num_on} -nreps 2 ${num_on} -nreps 3 ${num_on} -seed
    ${seed_2} \
    -prefix RSF.stim.run2.${iter}. >& /dev/null

    # make some other random seed for Run 3

    @ seed_3 = $seed_3 + 1

    # create random order stim files

    RSFgen -nt ${ts} -num_stimts ${stim} -nreps 1 ${num_on} -nreps 2 ${num_on} -nreps 3 ${num_on} -seed
    ${seed_3} \
    -prefix RSF.stim.run3.${iter}. >& /dev/null

    # make some other random seed for Run 4

    @ seed_4 = $seed_4 + 1

    # create random order stim files
```

```
RSFgen -nt ${ts} -num_stimts ${stim} -nreps 1 ${num_on} -nreps 2 ${num_on} -nreps 3 ${num_on} -seed
${seed_4} \
-prefix RSF.stim.run4.${iter}. >& /dev/null
```

```
cat RSF.stim.run1.${iter}.1.1D RSF.stim.run2.${iter}.1.1D RSF.stim.run3.${iter}.1.1D
RSF.stim.run4.${iter}.1.1D > RSF.stim.all.${iter}.1.1D
cat RSF.stim.run1.${iter}.2.1D RSF.stim.run2.${iter}.2.1D RSF.stim.run3.${iter}.2.1D
RSF.stim.run4.${iter}.2.1D > RSF.stim.all.${iter}.2.1D
cat RSF.stim.run1.${iter}.3.1D RSF.stim.run2.${iter}.3.1D RSF.stim.run3.${iter}.3.1D
RSF.stim.run4.${iter}.3.1D > RSF.stim.all.${iter}.3.1D
```

```
# check deconvolution
```

```
3dDeconvolve \
-nodata -quiet -nfirst 0 -nlast 655 -polort 2 \
-num_stimts 3 -TR_times 2.5 -concat concat.1D \
-stim_file 1 "RSF.stim.all.${iter}.1.1D" -stim_label 1 'Cong' -stim_minlag 1 0 -stim_maxlag 1 6 \
-stim_file 2 "RSF.stim.all.${iter}.2.1D" -stim_label 2 'ICong' -stim_minlag 2 0 -stim_maxlag 2 6 \
-stim_file 3 "RSF.stim.all.${iter}.3.1D" -stim_label 3 'Neutral' -stim_minlag 3 0 -stim_maxlag 3 6 \
-num_glt 3 \
-gltsym 'SYM: +Cong[0..6] -Neutral[0..6]' -glt_label 1 'Cong-Neutral' \
-gltsym 'SYM: +ICong[0..6] -Neutral[0..6]' -glt_label 2 'ICong-Neutral' \
-gltsym 'SYM: +ICong[0..6] -Cong[0..6]' -glt_label 3 'ICong-Cong' \
>& 3dD.nodata.${iter}
```

```
# save the sum of the 3 LC values
```

```
set nums = ( `awk -F= '/LC/ {print $2 * 10000}' 3dD.nodata.${iter}` )
@ num_sum = $nums[1] + $nums[2] + $nums[3]
```

```
echo -n "$num_sum = $nums[1] + $nums[2] + $nums[3] : " >> $LCfile
echo "iteration $iter, seed $seed_1 $seed_2 $seed_3 $seed_4" >> $LCfile
```

```
mv RSF* $outdir
mv 3dD.nodata* $outdir
```

```
echo -n "\b\b\b$iter"
```

```
end
```

```
echo ""
echo "done, results are in '$outdir', LC sums are in '$LCfile'"
echo consider the command: "sort $LCfile | head -1"
```

```
#!/bin/csh -f

/# The overall script to analyze each subject data

# David C. Zhu, Spring 2007

ln -s ../AnalyScripts/register.s .

register.s

cd Afni_analy

###

3dTshift -prefix ts_TS1+orig TS_1+orig
3dTshift -prefix ts_TS2+orig TS_2+orig
3dTshift -prefix ts_TS3+orig TS_3+orig
3dTshift -prefix ts_TS4+orig TS_4+orig

# Register Time Series TS1
3dvolreg -dfile mot_tempTS1 -base 'ts_TS1+orig[1]' -prefix reg_TS1 ts_TS1+orig

# Register Time Series TS2
3dvolreg -dfile mot_tempTS2 -base 'ts_TS1+orig[1]' -prefix reg_TS2 ts_TS2+orig

# Register Time Series TS3
3dvolreg -dfile mot_tempTS3 -base 'ts_TS1+orig[1]' -prefix reg_TS3 ts_TS3+orig

# Register Time Series TS4
3dvolreg -dfile mot_tempTS4 -base 'ts_TS1+orig[1]' -prefix reg_TS4 ts_TS4+orig

#Concatenate each group of Registered Time Series

3dTcat -session . -prefix reg_cat_TSall_before_blur reg_TS1+orig reg_TS2+orig reg_TS3+orig reg_TS4+orig

#=====
# Try to blurring it a bit first (Gaussian filter with a FWHM of 4 mm)
#=====

3dmerge -1blur_fwhm 4 -doall -prefix reg_cat_TSall reg_cat_TSall_before_blur+orig
```

```
#Put the motion plots in one file

cat mot_tempTS* > motion_plotTSall

#Plot Rotations and Translations
#1dplot -volreg -xlabel 'Time Point' 'motion_plotTS1[1-6]'
#1dplot -volreg -xlabel 'Time Point' 'motion_plotTS2[1-6]'
#1dplot -volreg -xlabel 'Time Point' 'motion_plotTS3[1-6]'
#1dplot -volreg -xlabel 'Time Point' 'motion_plotTS4[1-6]'

#Create mask to only use Brain Voxels for Deconvolution
3dcalc -prefix mask800 -a 'ts_TS1+orig[1]' -expr 'astep(a,800)'
3dAutomask -prefix mask_auto reg_cat_TSall+orig

ln -s ../../ParadigmDesignII/TS*_analy.1D .
ln -s ../../ParadigmDesignII/concat.1D .

# 3dDeconvolve:

ln -s ../../AnalyScripts/deconv.s .

deconv.s

tNoiseSNRMapGen.s reg_TS1+orig
tNoiseSNRMapGen.s reg_TS2+orig
tNoiseSNRMapGen.s reg_TS3+orig
tNoiseSNRMapGen.s reg_TS4+orig

tNoiseSNRMapGen.s reg_cat_TSall+orig
```

```
#!/bin/csh -f
```

```
# David C. Zhu
```

```
# Register the raw image data to AFNI
```

```
rename_file.s
```

```
to3d -session Afni_analy -prefix Inplane './Inplane/*'
```

```
to3d -session Afni_analy -prefix TS_1 -epan -time:zt 34 164 2500 alt+z './TS_1/*'
```

```
to3d -session Afni_analy -prefix TS_2 -epan -time:zt 34 164 2500 alt+z './TS_2/*'
```

```
to3d -session Afni_analy -prefix TS_3 -epan -time:zt 34 164 2500 alt+z './TS_3/*'
```

```
to3d -session Afni_analy -prefix TS_4 -epan -time:zt 34 164 2500 alt+z './TS_4/*'
```

```
to3d -session Afni_analy -prefix T1Volume './T1Volume/*'
```

```
# The Deconvolve script
# David C. Zhu
```

```
3dDeconvolve -input reg_cat_TSall+orig -polort 2 -num_stimts 9 -concat concat.1D \  
  -stim_file 1 'TSall_analy.1D[0]' -stim_label 1 'Cong' -stim_minlag 1 0 -stim_maxlag 1 6 \  
  -stim_file 2 'TSall_analy.1D[1]' -stim_label 2 'ICong' -stim_minlag 2 0 -stim_maxlag 2 6 \  
  -stim_file 3 'TSall_analy.1D[2]' -stim_label 3 'Neutral' -stim_minlag 3 0 -stim_maxlag 3 6 \  
-stim_file 4 'motion_plotTSall[1]' -stim_label 4 roll \  
-stim_file 5 'motion_plotTSall[2]' -stim_label 5 pitch \  
-stim_file 6 'motion_plotTSall[3]' -stim_label 6 yaw \  
-stim_file 7 'motion_plotTSall[4]' -stim_label 7 IS \  
-stim_file 8 'motion_plotTSall[5]' -stim_label 8 RL \  
-stim_file 9 'motion_plotTSall[6]' -stim_label 9 AP \  
-mask mask800+orig \  
-fout -tout -vout \  
-num_glt 3 \  
-gltsym 'SYM: +Cong[0..6] -Neutral[0..6]' -glt_label 1 'Cong-Neutral' \  
  -gltsym 'SYM: +ICong[0..6] -Neutral[0..6]' -glt_label 2 'ICong-Neutral' \  
  -gltsym 'SYM: +ICong[0..6] -Cong[0..6]' -glt_label 3 'ICong-Cong' \  
-iresp 1 IRF_Cong \  
-iresp 2 IRF_ICong \  
-iresp 3 IRF_Neutral \  
-bucket deconv_TS
```



```

#!/bin/csh -f

# Calculate the % signal change
# David C. Zhu

rm -f mean_baseline*
rm -f *per_sig_ch*

3dcalc -a deconv_TS+orig'[0]' -b deconv_TS+orig'[6]' -c deconv_TS+orig'[12]' \
      -d deconv_TS+orig'[18]' -expr 'mean(a,b,c,d)' -prefix mean_baseline

3dcalc -a mean_baseline+orig -b deconv_TS+orig'[24]' -c deconv_TS+orig'[26]' -d deconv_TS+orig'[28]' \
      -e deconv_TS+orig'[30]' -f deconv_TS+orig'[32]' -g deconv_TS+orig'[34]' -h deconv_TS+orig'[36]' \
      -expr '((b+c+d+e+f+g+h)/a)*100' -prefix Cong_per_sig_ch+orig

3dcalc -a mean_baseline+orig -b deconv_TS+orig'[39]' -c deconv_TS+orig'[41]' -d deconv_TS+orig'[43]' \
      -e deconv_TS+orig'[45]' -f deconv_TS+orig'[47]' -g deconv_TS+orig'[49]' -h deconv_TS+orig'[51]' \
      -expr '((b+c+d+e+f+g+h)/a)*100' -prefix ICong_per_sig_ch+orig

3dcalc -a mean_baseline+orig -b deconv_TS+orig'[54]' -c deconv_TS+orig'[56]' -d deconv_TS+orig'[58]' \
      -e deconv_TS+orig'[60]' -f deconv_TS+orig'[62]' -g deconv_TS+orig'[64]' -h deconv_TS+orig'[66]' \
      -expr '((b+c+d+e+f+g+h)/a)*100' -prefix Neutral_per_sig_ch+orig

foreach cond (Cong ICong Neutral)
  adwarp -apar T1Volume+tlrc -dpar {$cond}_per_sig_ch+orig
end

```

```

#!/bin/csh -f

# David C. Zhu, Spring 2007
# ANOVA2 analysis

#=====
# Begin by making a new directory called 'FLY_group_ANOVA2'
#=====
    rm -f -R FLY_group_ANOVA2

    mkdir FLY_group_ANOVA2

#=====
# cd into subject FLY_001 directory
#=====

    cp ./FLY_003/Afni_analy/T1* ./FLY_group_ANOVA2/

#=====
# For each subject, copy their mean irf Talairached datasets into 'FLY_group_ANOVA2'
# No need to use '3dcopy' b/c these are being moved to 'FLY_group_ANOVA2' and
# just being used for ANOVA.
#=====

foreach subj (003 004 007 009 012 013 015 017)
    cd ./FLY_{$subj}/Afni_analy/
    ln -s -f ../../AnalyScripts/PerSigChan.s .
    PerSigChan.s
    cp Cong_per_sig_ch+tlrc.BRIK ../../FLY_group_ANOVA2/FLY_{$subj}_Cong_per_sig_ch+tlrc.BRIK
    cp Cong_per_sig_ch+tlrc.HEAD ../../FLY_group_ANOVA2/FLY_{$subj}_Cong_per_sig_ch+tlrc.HEAD

    cp ICong_per_sig_ch+tlrc.BRIK ../../FLY_group_ANOVA2/FLY_{$subj}_ICong_per_sig_ch+tlrc.BRIK
    cp ICong_per_sig_ch+tlrc.HEAD ../../FLY_group_ANOVA2/FLY_{$subj}_ICong_per_sig_ch+tlrc.HEAD

    cp Neutral_per_sig_ch+tlrc.BRIK ../../FLY_group_ANOVA2/FLY_{$subj}_Neutral_per_sig_ch+tlrc.BRIK
    cp Neutral_per_sig_ch+tlrc.HEAD ../../FLY_group_ANOVA2/FLY_{$subj}_Neutral_per_sig_ch+tlrc.HEAD

    cd ../../
end

```

```

#=====
# cd into 'FLY_group_ANOVA2' directory and run ANOVA from there...
#=====

cd FLY_group_ANOVA2

# run ANOVA now

3dANOVA2 -type 3 -alevels 3 -blevels 8 \
  -dset 1 1 FLY_003_Cong_per_sig_ch+tlrc \
  -dset 2 1 FLY_003_ICong_per_sig_ch+tlrc \
  -dset 3 1 FLY_003_Neutral_per_sig_ch+tlrc \
  -dset 1 2 FLY_004_Cong_per_sig_ch+tlrc \
  -dset 2 2 FLY_004_ICong_per_sig_ch+tlrc \
  -dset 3 2 FLY_004_Neutral_per_sig_ch+tlrc \
  -dset 1 3 FLY_007_Cong_per_sig_ch+tlrc \
  -dset 2 3 FLY_007_ICong_per_sig_ch+tlrc \
  -dset 3 3 FLY_007_Neutral_per_sig_ch+tlrc \
  -dset 1 4 FLY_009_Cong_per_sig_ch+tlrc \
  -dset 2 4 FLY_009_ICong_per_sig_ch+tlrc \
  -dset 3 4 FLY_009_Neutral_per_sig_ch+tlrc \
  -dset 1 5 FLY_012_Cong_per_sig_ch+tlrc \
  -dset 2 5 FLY_012_ICong_per_sig_ch+tlrc \
  -dset 3 5 FLY_012_Neutral_per_sig_ch+tlrc \
  -dset 1 6 FLY_013_Cong_per_sig_ch+tlrc \
  -dset 2 6 FLY_013_ICong_per_sig_ch+tlrc \
  -dset 3 6 FLY_013_Neutral_per_sig_ch+tlrc \
  -dset 1 7 FLY_015_Cong_per_sig_ch+tlrc \
  -dset 2 7 FLY_015_ICong_per_sig_ch+tlrc \
  -dset 3 7 FLY_015_Neutral_per_sig_ch+tlrc \
  -dset 1 8 FLY_017_Cong_per_sig_ch+tlrc \
  -dset 2 8 FLY_017_ICong_per_sig_ch+tlrc \
  -dset 3 8 FLY_017_Neutral_per_sig_ch+tlrc \
  -amean 1 Cong -amean 2 ICong -amean 3 Neutral \
  -adiff 2 1 ICong_Cong -adiff 2 3 ICong_Neutral -adiff 3 1 Neutral_Cong \
  -fa StimFtest \
  -bucket 3dANOVA2.result

```

```
#=====
# end of 3dANOVA
#=====

# generate some cluster report

# ln -s -f ../AnalyScripts/gen_cluster.s .

# gen_cluster.s
```