Source Localization with MEG

Dr. Tom Holroyd
NIMH MEG Core Facility

MEG Short Course

August 7, 2018

Overview – SAM Workflow

- Things you need:
  - Structural MRI including the nose and ears
  - MEG data
  - Research hypotheses
- Process the MRI – mark the fiducials, create head conductivity model
- Process the MEG – add event markers, filter, add MRI-based head model
- `sam_cov` – create covariance matrices
- `sam_wts` – create beamformer weights
- `sam_3d`, etc. – create source images
- Group stats – AFNI

MRI Processing

```
#! /bin/sh

if [ "$#" -ne 1 ]; then
        echo usage: $0 hashcode
        exit
fi

hashcode="$1"

cd mri/${hashcode}

orthohull -c  # clean up old files
orthohull -t  # ortho + tlrc
```

Once the fiducial markers have been placed, the MRI is rotated into *ortho* space, and the skull removed.

From that, we compute a conductivity model.

This is accomplished by the `orthohull` script. It also creates Talairach (or MNI) versions, for group averaging.

Usually a script like `doOrthohull.sh` is used.

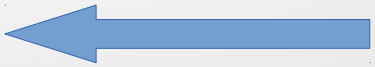
MRI Processing

MEG Event Markers – doThresh.sh

```
#! /bin/sh

if [ "$#" -ne 1 ]; then
        echo usage: $0 hashcode
        exit
fi

hashcode="$1"
ds=`grep ${hashcode} dslist`
echo $ds
dirname=`dirname $ds`
setname=`basename $ds .ds`

# threshold the ADCs for the ASVEF task

thresholdDetect2 -np -i -m leftpuff -a .5 -d .5 -dt 1 -c UADC001 $ds
thresholdDetect2 -np -i -m rightpuff -a .5 -d .5 -dt 1 -c UADC002 $ds
thresholdDetect2 -np -i -m beep -a .25 -d .25 -dt 1 -c UADC003 $ds
thresholdDetect2 -np -i -m check -a .5 -d .5 -dt 1 -c UADC016 $ds

newDs -f -filter processing.cfg $ds ${dirname}/${setname}-f.ds
```

Along with the MEG data, various information is collected in separate digital and analog channels. Analog channels store things like button presses or stimulus triggers. Here, ADCs 1 & 2 store airpuff stimulator triggers. These can be given names using the `thresholdDetect` program, which is a CTF utility.

Next we apply processing filters.



processing.cfg

```
processing
{
        // balance: order, adapted
        // (adapted=0 -> not adapted)
        // (adapted=1 -> adapted)
        balance:        3,0
        // lowpass: enable, filterOrder, fc
        lowpass:        0,4,0.00000000000000000
        // highpass: enable, filterOrder, fc
        highpass:       1,4,0.50000000000000000
        // bandreject: enable, filterOrder, fc1, fc2
        bandreject:     0,4,0.00000000000000000,0.00000000000000000
        // bandpass: enable, filterOrder, fc1, fc2
        bandpass:       0,4,0.00000000000000000,0.00000000000000000
        // bandreject: enable, filterOrder, fc1, fc2
        bandreject:     1,2,59.500000000000000000,60.500000000000000000
        // bandreject: enable, filterOrder, fc1, fc2
        bandreject:     1,2,119.500000000000000000,120.500000000000000000
        // bandreject: enable, filterOrder, fc1, fc2
        bandreject:     1,2,179.500000000000000000,180.500000000000000000
        // bandreject: enable, filterOrder, fc1, fc2
        bandreject:     1,2,239.500000000000000000,240.500000000000000000
        // offset: enable, baselineSelection, startPt, endPt
        // (baseline=0 --> use pretrigger data)
        // (baseline=1 --> use from startPt to endPt)
        // (baseline=2 --> use whole trial)
        // (baseline+=10 --> do trend removal)
        offset: 1,2,1,1
}
```

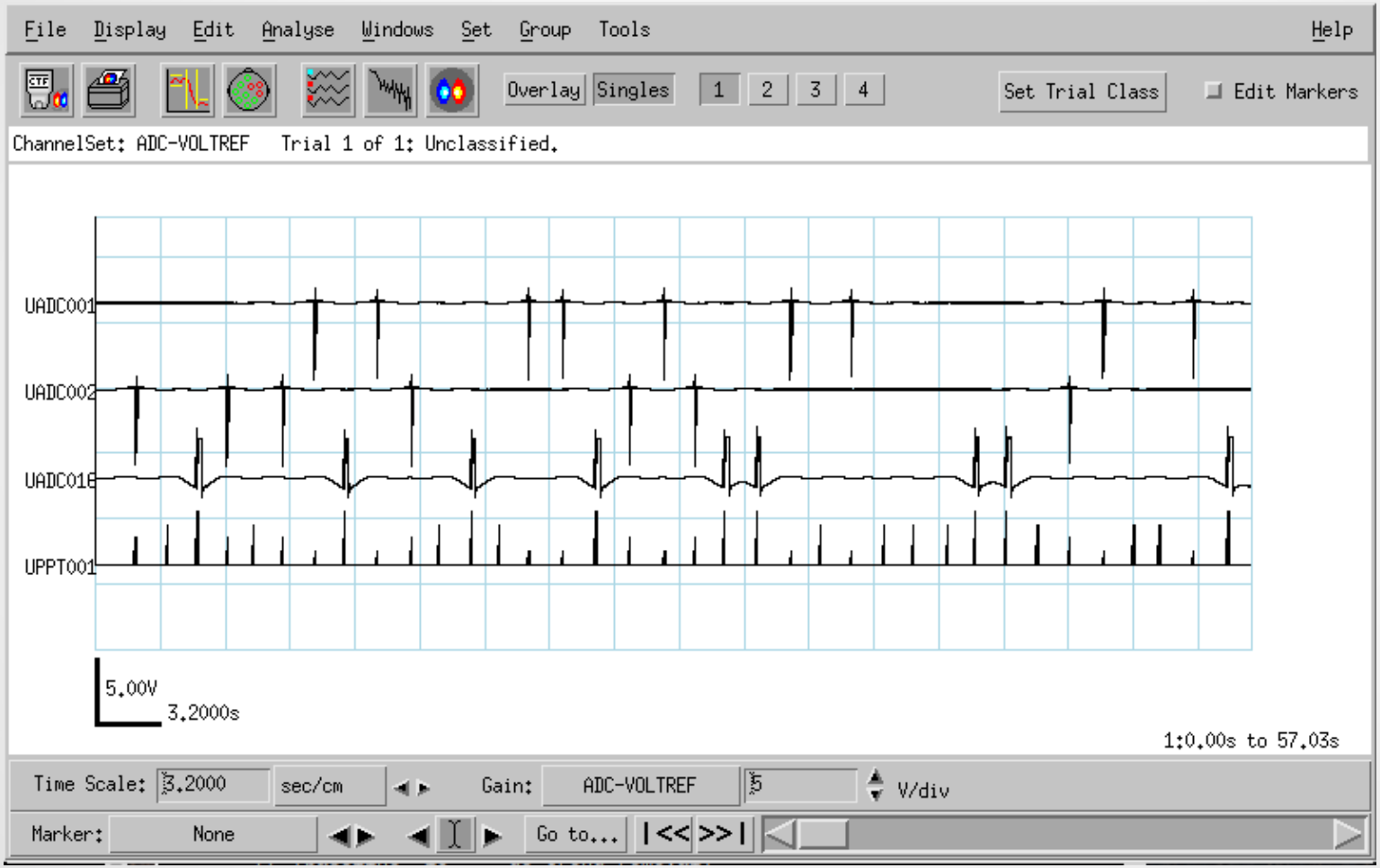
3rd gradient (default)

High pass filter (optional)

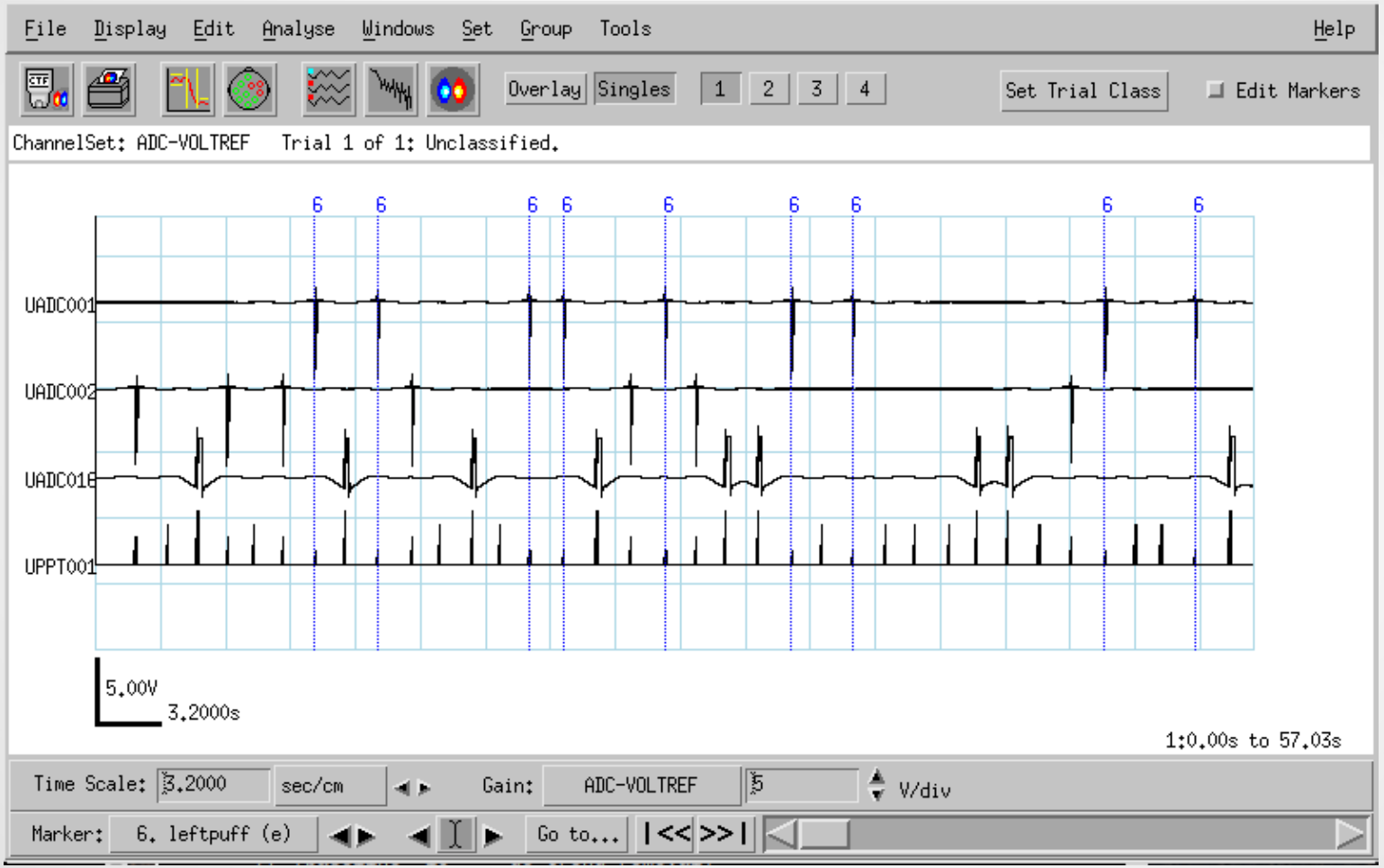
60 Hz line-noise notch filters including harmonics

Baseline removal

Visually Check the Markers



Visually Check the Markers



doLocalSpheres.sh

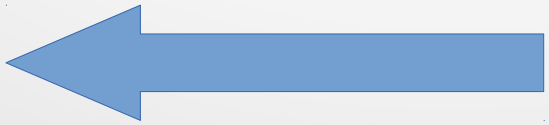
```
#! /bin/sh


if [ "$#" -ne 1 ]; then
        echo usage: $0 dataset
        exit
fi


ds="$1"
hashcode=`basename $ds | cut -d_ -f1`


localSpheres -s mri/${hashcode}/multisphere.shape -d $ds -r 9
```

The `multisphere.shape` file is the output of `orthohull`. `localSpheres` uses that to create the final MultiSphere model used for magnetic field calculations. It is stored inside the MEG dataset folder.

doSam.sh

```
#! /bin/sh

dosam() {
    ds="$1"
    m="$2"

    sam_cov -r $ds -m $m
    sam_wts -r $ds -m $m
    sam_4d -r $ds -m $m
}

for dsname in `cat dslist`; do
    dirname=`dirname $dsname`          # directory
    setname=`basename $dsname .ds`     # setname without .ds
    ds=${dirname}/${setname}-f.ds      # full pathname

    dosam $ds sef
done
```

Usually we use a script like this to run SAM. First `sam_cov` creates the covariance matrices (which are stored in the dataset's SAM subdirectory). Then, `sam_wts` creates the beamformers for the ROI. Finally, other programs such as `sam_3d` or `sam_4d` use the weights to create 3d (or 3d+time) volumes.



All SAM analysis is controlled by a *parameter* file (specified with `-m`).

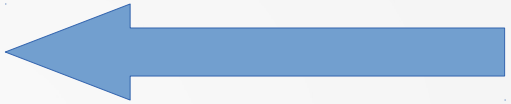
SAM parameter files

```
%include ROI            # standard ROI

NumMarkers 2            # dataset markers
Marker1 rightpuff -.2 .5 true
Marker2 leftpuff -.2 .5 true

OrientBand 15 30        # beta is recommended because of its stability
CovBand 15 30           # the covariance band can be larger
ImageBand 15 30         #   than the image band, for stability
SmoothBand 0 20         # for sam_4d
TimeStep .01

Model MultiSphere       # forward model
Mu 2                    # optional regularization (stability)

MRIDirectory mri        # input and
ImageDir image          #   output dirs

ImageMetric Power       # what sort of image to compute
CovType SUM             # which covariance to use for the image
```

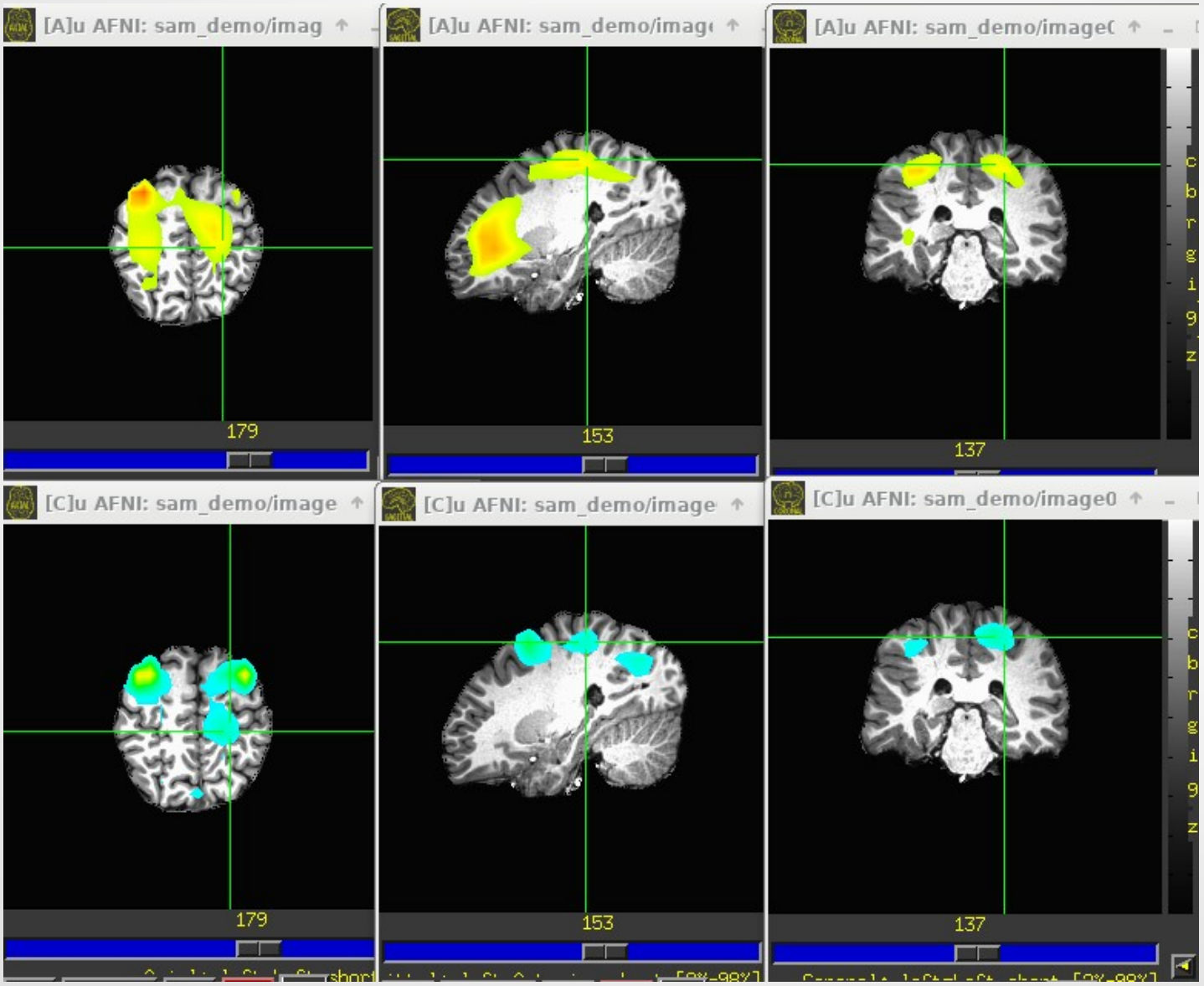
Running AFNI – `.afnirc`

```
***ENVIRONMENT
 AFNI_LEFT_IS_LEFT = YES
 AFNI_ALWAYS_LOCK = YES
 AFNI_ORIENT = PRI
 AFNI_DATASET_BROWSE = YES
 AFNI_DECONFLICT = OVERWRITE
 AFNI_NIFTI_VIEW = orig
 AFNI_SLAVE_THROLAY = YES
```

CTF ordering is PRI



sam_3d – Mean and Variance



sam_4d – 3d+time Images and Graph