This Procedure is for those who wants to use eeglab to process the EGI data.

Tested software version: Matlab 2015b, Nestation 5.3.0.1, EEGLAB V13.6.5b

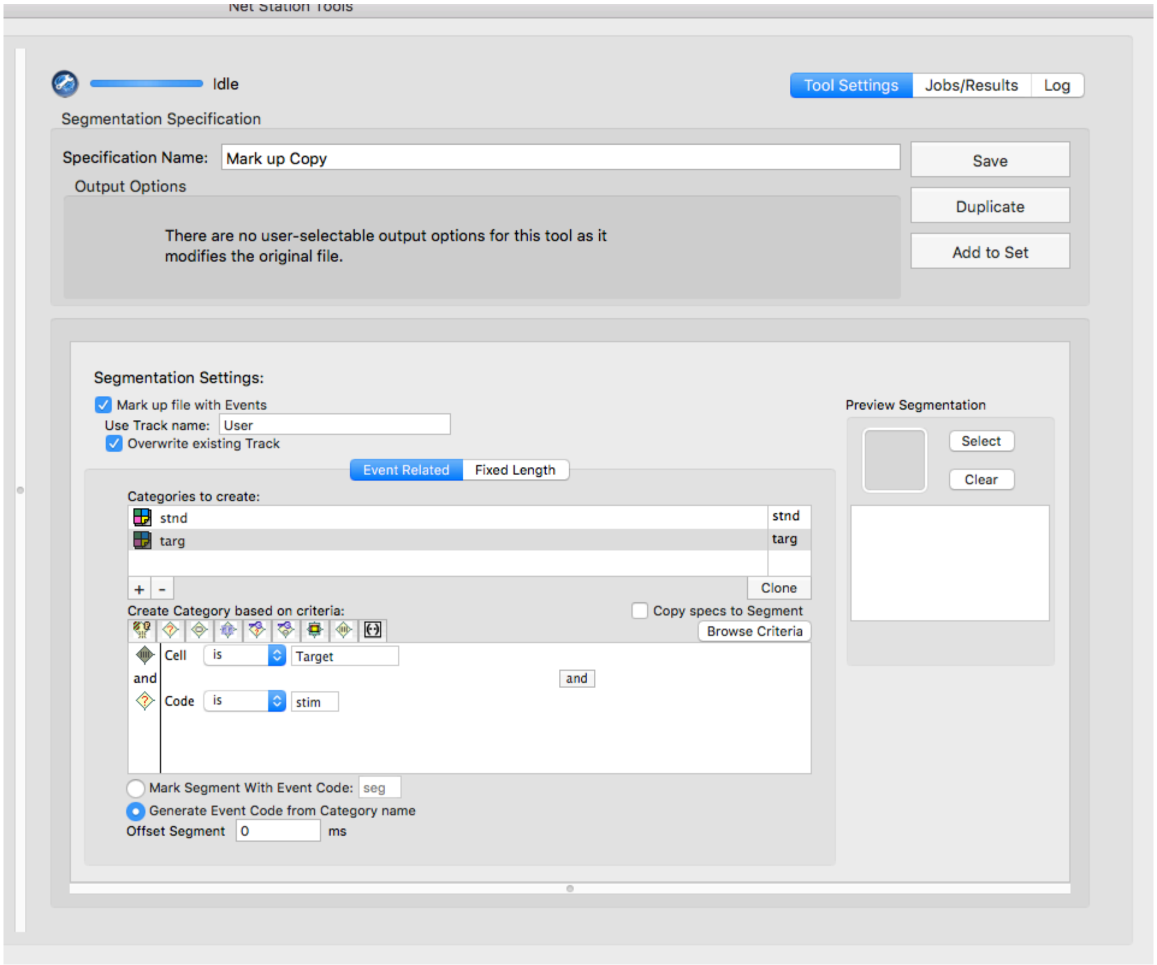
Before we export data to EEGLAB, we need to know some difference to treat Events name between Netstation and EEGLAB.

Topic 1 Markup function

Net Station events contain key lists, in other words, mini databases. Using a simple standard/target experiment as an example, in Net Station all stimuli events might be named “stim.” The distinction between standards and targets can’t be determined from the name of the event. It can only be determined from the key list in the events.

In most systems, including FieldTrip, events only have names. They don’t have key lists. To make the jump from key-list events to non-keylist events, you must use Net Station’s segmentation markup. Segmentation markup adds events to the recording that you can use to distinguish different conditions. To use segmentation markup, create a segmentation specification, and check the “Mark Up File” checkbox in the segmentation specification editor. When you run segmentation using this specification, instead of segmenting the file, new events will be added. For example, if all your standard and target stimuli are named “stim”, you would create a segmentation specification just as you would for segmenting this file into standard and target categories. Then, if you check the “Mark Up File” checkbox, the specification will cause new events to be added to the file instead of segmentation. Then, you can add events called “stnd” for standard, and “targ” for target.

Note: Use 4 ASCII characters to name the events. Do not include Space.



Segmentation Specification

Specification Name: Mark up Copy

Output Options

There are no user-selectable output options for this tool as it modifies the original file.

Segmentation Settings:

☑ Mark up file with Events

Use Track name: User

☑ Overwrite existing Track

Event Related | Fixed Length

Categories to create:

stnd	stnd
targ	targ

Create Category based on criteria:

□ Copy specs to Segment

Cell is Target

and

Code is stim

○ Mark Segment With Event Code: seg

● Generate Event Code from Category name

Offset Segment 0 ms

Preview Segmentation

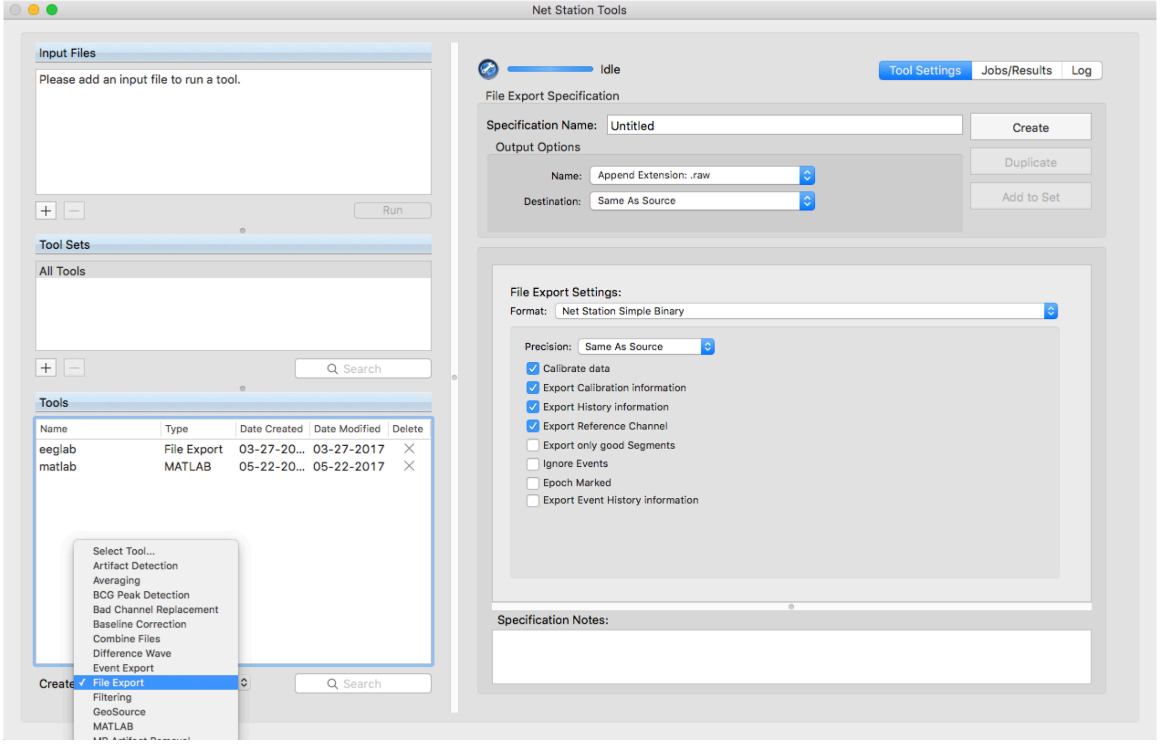
Topic 2: Load location file and Average reference in EEGLAB

Then you need to consider if you want to make use of the "bad channel replacement" tool in NetStation. Sometimes it's very useful, especially when there are some bad channels while you do want to retain them.

If you do not want it and just need to finish all the work in EEGlab, please go to Topic 2 . If you think the toos is necessary for you, then go to Topic 3.

Step 1: export your file to Simple binary file. Launch Netstation tools, and create a file export tool.

You can check the options what you want to keep. For example, if you want to see Ref channels in EEGLAB, then check option"Export Reference Channel". Then name the tool, and click "Create" button.



Net Station Tools

Input Files

Please add an input file to run a tool.

Tool Sets

All Tools

Tools

Name	Type	Date Created	Date Modified	Delete
eeglab	File Export	03-27-20...	03-27-2017	×
matlab	MATLAB	05-22-20...	05-22-2017	×

Create ✓ File Export

Idle

File Export Specification

Specification Name: Untitled

Output Options

Name: Append Extension: .raw

Destination: Same As Source

File Export Settings:

Format: Net Station Simple Binary

Precision: Same As Source

☑ Calibrate data
☑ Export Calibration information
☑ Export History information
☑ Export Reference Channel
☐ Export only good Segments
☐ Ignore Events
☐ Epoch Marked
☐ Export Event History information

Specification Notes:

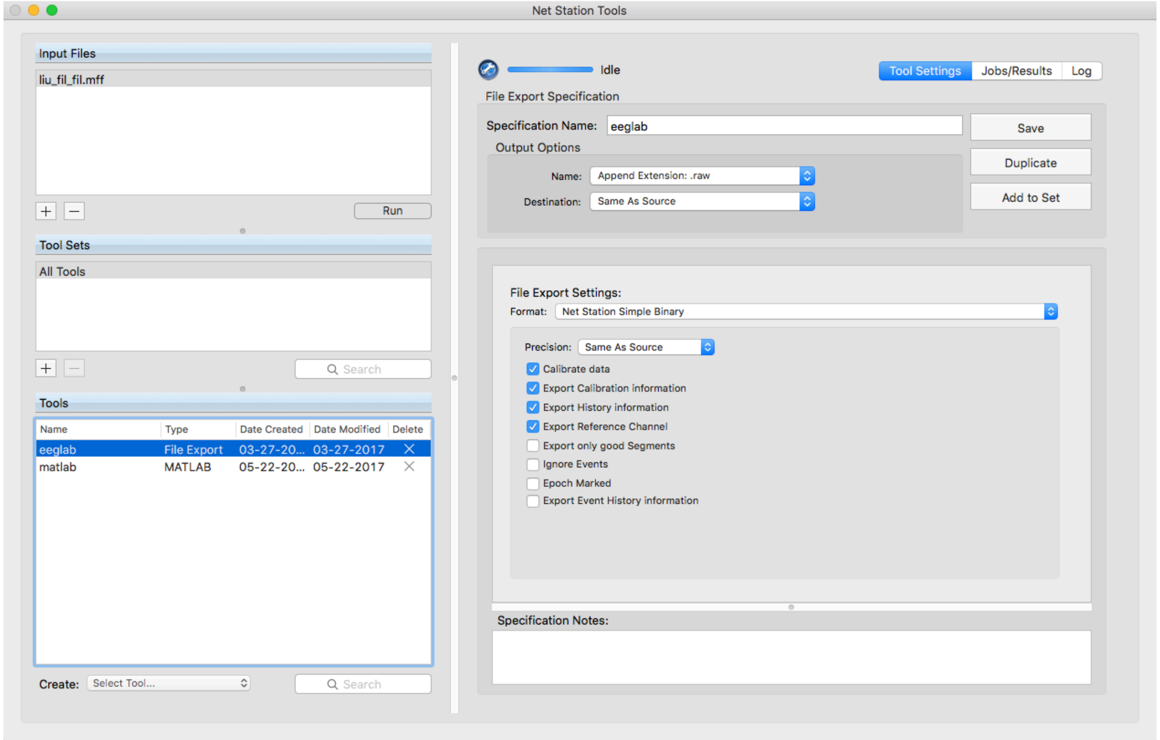
Step 2: Run the tool on the file you want to convert. After that, you will see a file whose format is raw.

Note:

- EEGLab does not recognize breaks in recording (Epoch breaks) when reading .raw files.
- To avoid recording breaks with EPrime for experiment control, add "false" to the end of parameter line in NSInit object, such as following:

c, "on", CellList, "socket", "10.10.10.42", false

And do not put NSStopRecording object except at the end of experiment.



Input Files: liu_fil_fil.mff

Name	Type	Date Created	Date Modified	Delete
eeglab	File Export	03-27-20...	03-27-2017	✕
matlab	MATLAB	05-22-20...	05-22-2017	✕

File Export Specification

Specification Name: eeglab

Output Options

Name: Append Extension: .raw

Destination: Same As Source

File Export Settings:

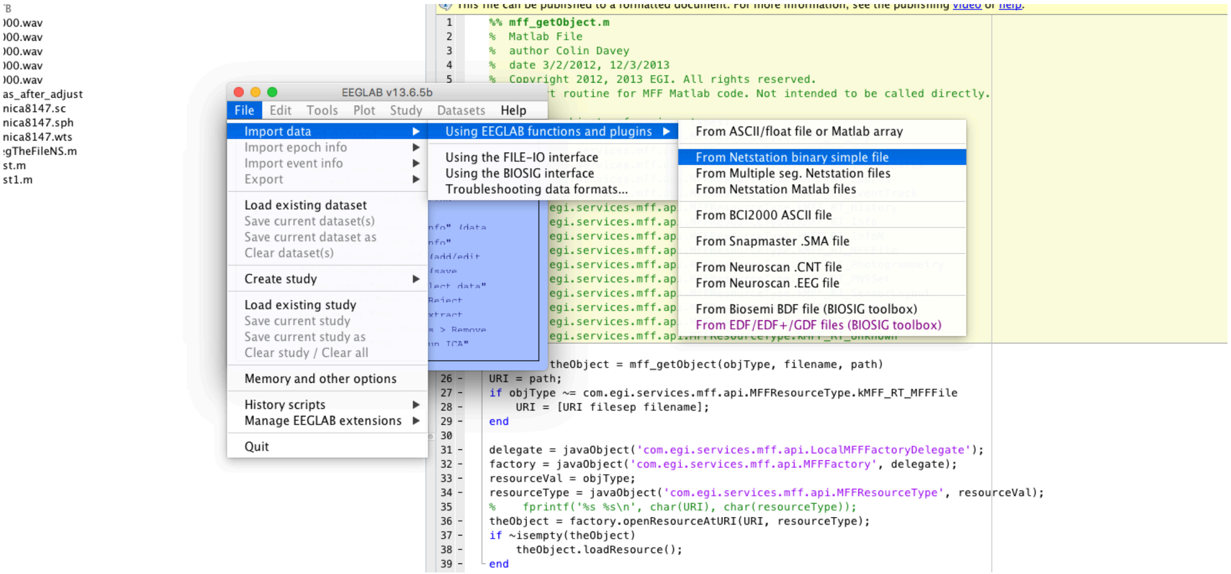
Format: Net Station Simple Binary

Precision: Same As Source

☑ Calibrate data
☑ Export Calibration information
☑ Export History information
☑ Export Reference Channel
☐ Export only good Segments
☐ Ignore Events
☐ Epoch Marked
☐ Export Event History information

Step 3: Lauch eeglab and import the data.

Go to File/import data/Using EEGLAB functions and plugins/From nestation binary simple file:



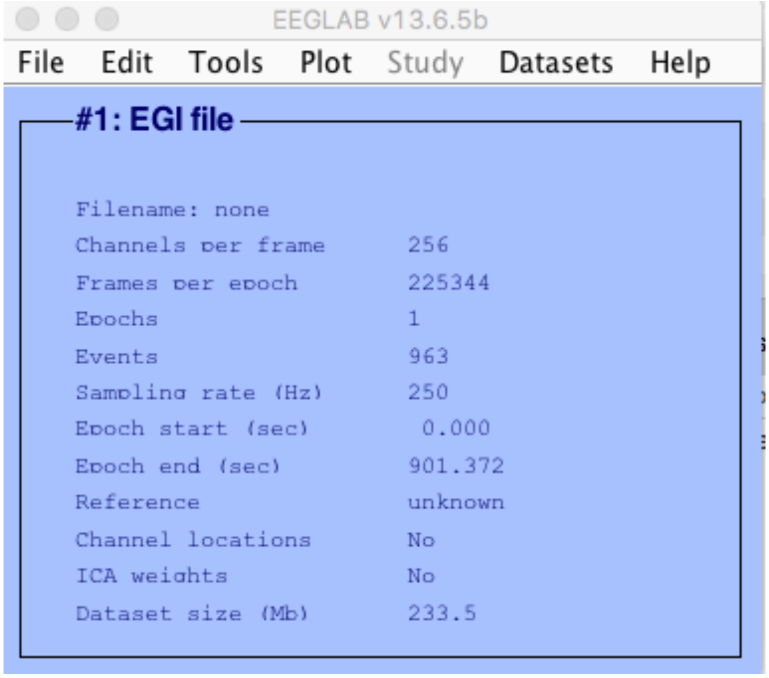
```
%% mff_getObject.m
%  Matlab File
%  author Colin Davey
%  date 3/2/2012, 12/3/2013
%  Copyright 2012, 2013 EGI. All rights reserved.
  t routine for MFF Matlab code. Not intended to be called directly.

egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.ap
egi.services.mff.api.MFFResourceType.kMFF_RT_Unknown

  theObject = mff_getObject(objType, filename, path)
URI = path;
if objType ~= com.egi.services.mff.api.MFFResourceType.kMFF_RT_MFFFile
    URI = [URI filesep filename];
end

delegate = javaObject('com.egi.services.mff.api.LocalMFFFactoryDelegate');
factory = javaObject('com.egi.services.mff.api.MFFFactory', delegate);
resourceVal = objType;
resourceType = javaObject('com.egi.services.mff.api.MFFResourceType', resourceVal);
%    fprintf('%s %s\n', char(URI), char(resourceType));
theObject = factory.openResourceAtURI(URI, resourceType);
if ~isempty(theObject)
    theObject.loadResource();
end
```

The following windows will tell you EEGLAB will use the default location file, just click Ok. Then name your file:

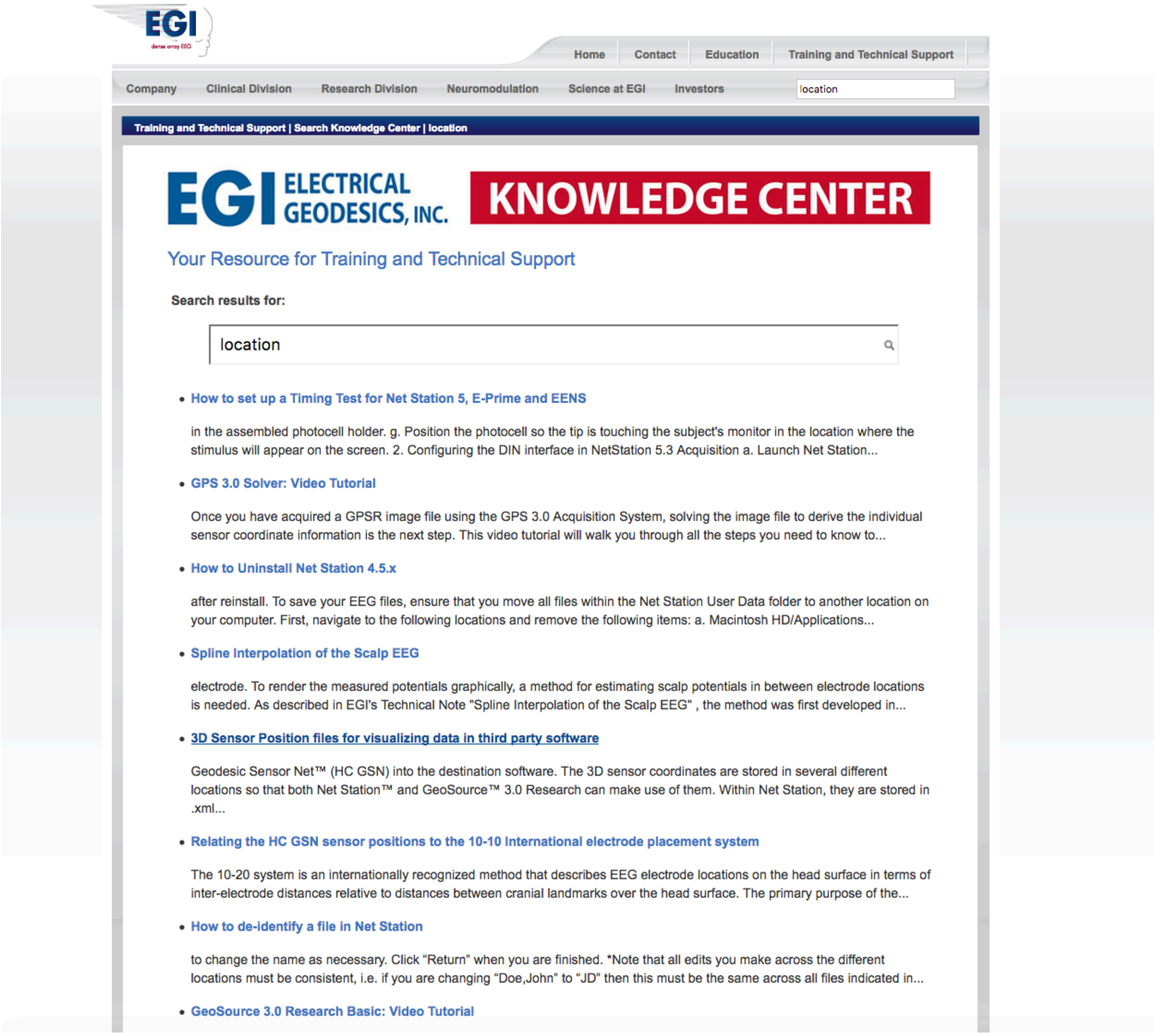


#1: EGI file

```
Filename: none
Channels per frame       256
Frames per epoch         225344
Epochs                   1
Events                   963
Sampling rate (Hz)       250
Epoch start (sec)         0.000
Epoch end (sec)          901.372
Reference                unknown
Channel locations        No
ICA weights              No
Dataset size (Mb)        233.5
```

Step 4: Load the location file.

Plase visit www.egi.com and enter knowledge center by click “Trianing and Technical support” tab. Then search the keyword “location” . You will see a list of topics related to keyword “location”. Click the link named “**3D Sensor Position files for visualizing data in third party software**”



Your Resource for Training and Technical Support

Search results for:

location

- How to set up a Timing Test for Net Station 5, E-Prime and EENS

  in the assembled photocell holder. g. Position the photocell so the tip is touching the subject's monitor in the location where the stimulus will appear on the screen. 2. Configuring the DIN interface in NetStation 5.3 Acquisition a. Launch Net Station...

- GPS 3.0 Solver: Video Tutorial

  Once you have acquired a GPSR image file using the GPS 3.0 Acquisition System, solving the image file to derive the individual sensor coordinate information is the next step. This video tutorial will walk you through all the steps you need to know to...

- How to Uninstall Net Station 4.5.x

  after reinstall. To save your EEG files, ensure that you move all files within the Net Station User Data folder to another location on your computer. First, navigate to the following locations and remove the following items: a. Macintosh HD/Applications...

- Spline Interpolation of the Scalp EEG

  electrode. To render the measured potentials graphically, a method for estimating scalp potentials in between electrode locations is needed. As described in EGI's Technical Note "Spline Interpolation of the Scalp EEG" , the method was first developed in...

- 3D Sensor Position files for visualizing data in third party software

  Geodesic Sensor Net™ (HC GSN) into the destination software. The 3D sensor coordinates are stored in several different locations so that both Net Station™ and GeoSource™ 3.0 Research can make use of them. Within Net Station, they are stored in .xml...

- Relating the HC GSN sensor positions to the 10-10 International electrode placement system

  The 10-20 system is an internationally recognized method that describes EEG electrode locations on the head surface in terms of inter-electrode distances relative to distances between cranial landmarks over the head surface. The primary purpose of the...

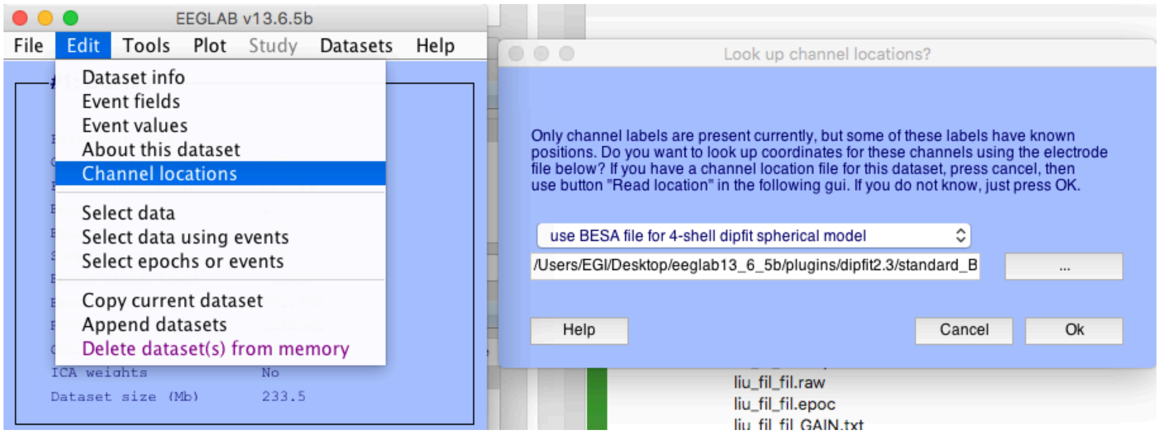
- How to de-identify a file in Net Station

  to change the name as necessary. Click "Return" when you are finished. *Note that all edits you make across the different locations must be consistent, i.e. if you are changing "Doe,John" to "JD" then this must be the same across all files indicated in...

- GeoSource 3.0 Research Basic: Video Tutorial

Then choose the file you need to use. Download it and put it on your desktop. Then we need to load it into your EEGLAB data.

In eeglab, go to Edit and click “channel locations”, click cancel button in the following pop up window. “Edit channel info window” will appear then.



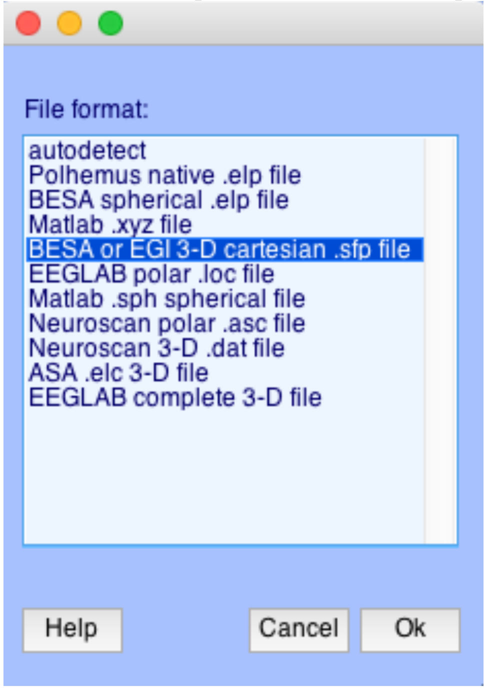
ICA weights No
Dataset size (Mb) 233.5

Only channel labels are present currently, but some of these labels have known positions. Do you want to look up coordinates for these channels using the electrode file below? If you have a channel location file for this dataset, press cancel, then use button "Read location" in the following gui. If you do not know, just press OK.

use BESA file for 4-shell dipfit spherical model

/Users/EGI/Desktop/eeglab13_6_5b/plugins/dipfit2.3/standard_B

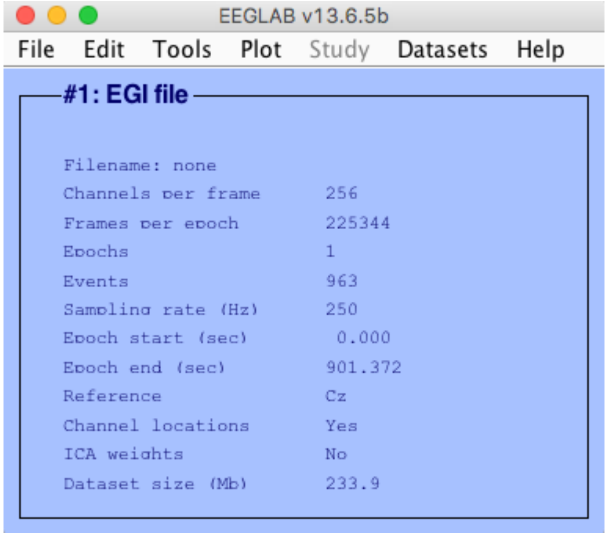
Then Click “Read Locatioins” button and choose the location file you just downloaded. Choose the 5th option, so that the sfp file can be recognized.



Go to the last channel. Then you will see it should be Cz. Uncheck the option”Channel in data array”, and click the “Set reference” button. In the following window: you need to enter “1:257” into Channel indices, and “Cz” into Reference.

Note, Here we use a 256ch data, if your data use different channels, it need to be different. For example, if your data is 128ch, enter “1:129”

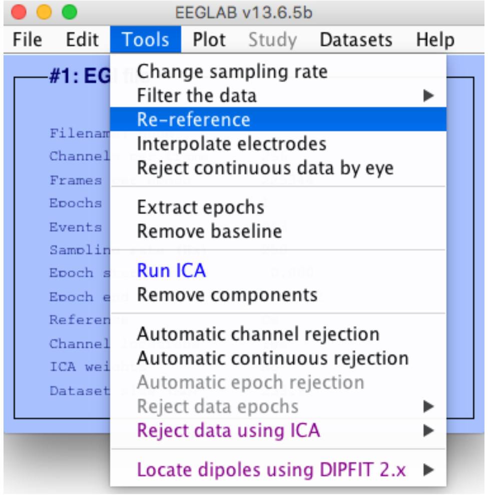
After that, you will see your data has be recognized to have ref channel: Cz



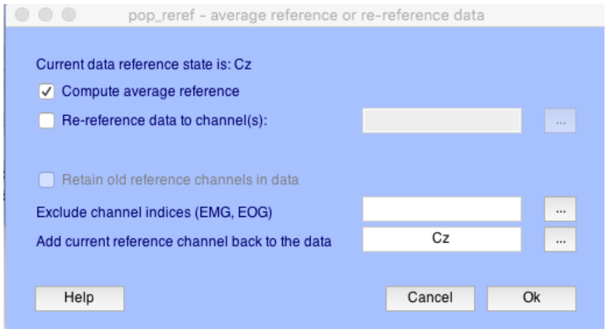
#1: EGI file

Filename: none	
Channels per frame	256
Frames per epoch	225344
Epochs	1
Events	963
Sampling rate (Hz)	250
Epoch start (sec)	0.000
Epoch end (sec)	901.372
Reference	Cz
Channel locations	Yes
ICA weights	No
Dataset size (Mb)	233.9

Step 5: Average reference(optional). Tools/Re-reference:



The important thing you need to keep in mind is that you need to add ref channel back to your data channel. Check "compute average reference" and add Cz back.



pop_reref - average reference or re-reference data

Current data reference state is: Cz

☑ Compute average reference

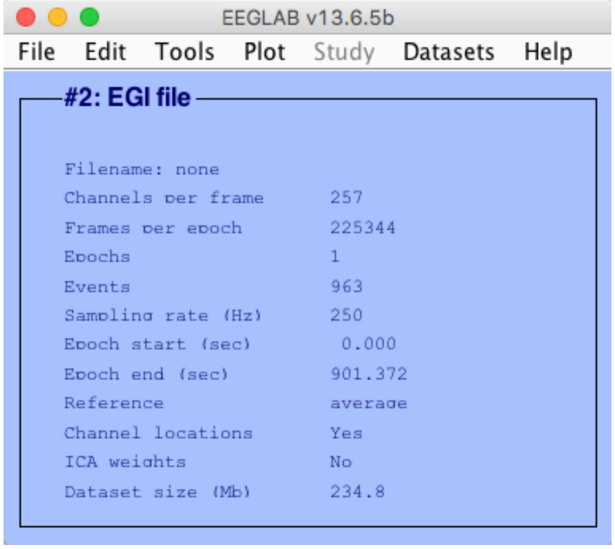
☐ Re-reference data to channel(s):

☐ Retain old reference channels in data

Exclude channel indices (EMG, EOG)

Add current reference channel back to the data: Cz

Then you will see that you already change your reference to Average reference. If you want to use another reference, just repeat this step and choose another reference channel whatever you want.



```
Filename: none
Channels per frame        257
Frames per epoch          225344
Epochs                    1
Events                    963
Sampling rate (Hz)        250
Epoch start (sec)          0.000
Epoch end (sec)           901.372
Reference                 average
Channel locations         Yes
ICA weights               No
Dataset size (Mb)         234.8
```

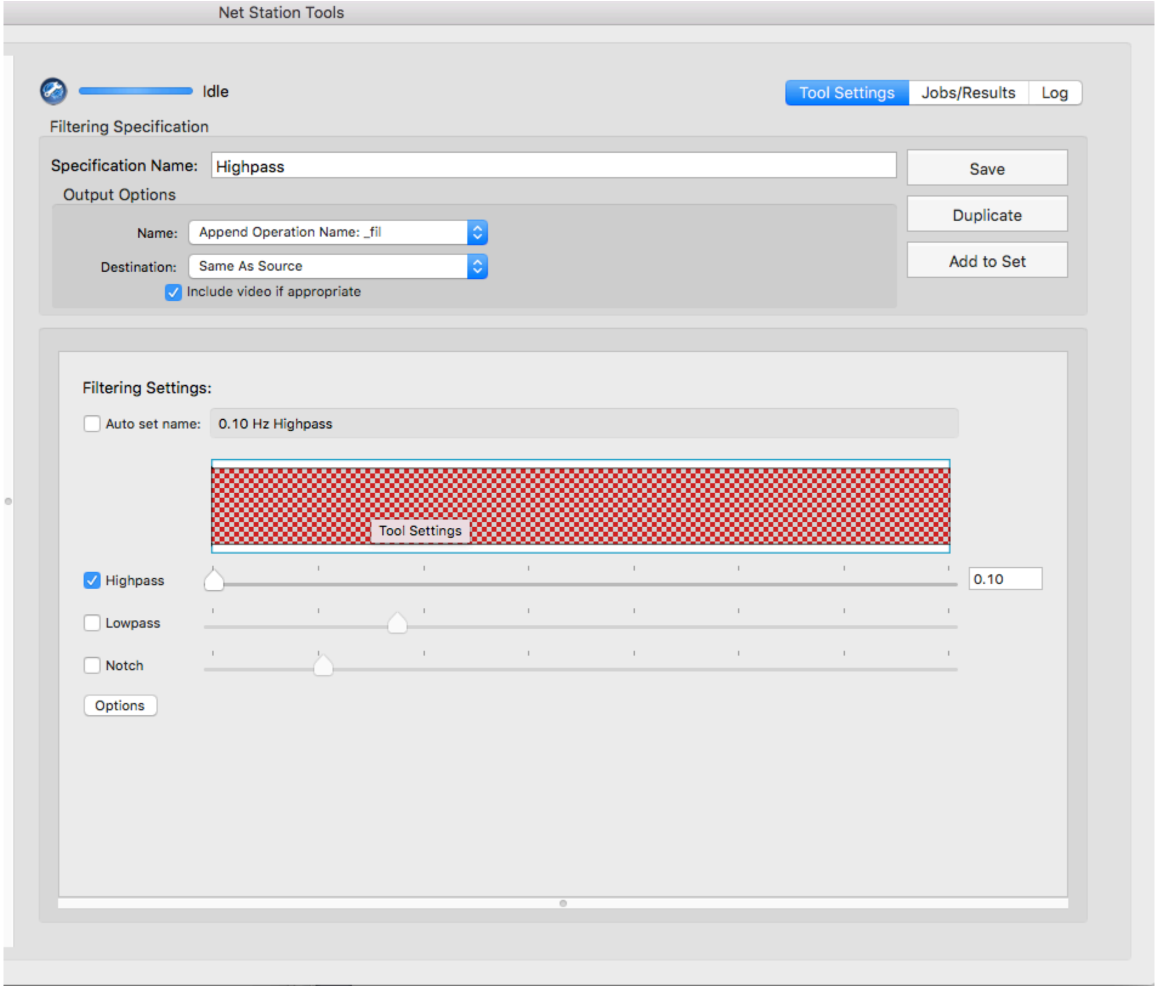
Topic 3 Bad channel replacement

Sometimes we want to make use of ICA function to remove artifacts such eyeblink, line nosie, EKG etc, while we found some bad channels and want to make use of the “bad channel replacement tool” in Netstation to maintain flexibility. Then we do the following steps:

Step 1: filter your data.

This step is necessary since we will use “artifact detection” tool to detect the bad channels in the segments. If you do not remove your DC signals, then this tool cannot work well.

Note: High pass needed. Low pass is optional, you can make use of the filter in EEGLAB or do ICA analysis to remove the line noise.



Specification Name: Highpass

Name: Append Operation Name: _fil

Destination: Same As Source

☑ Include video if appropriate

Filtering Settings:

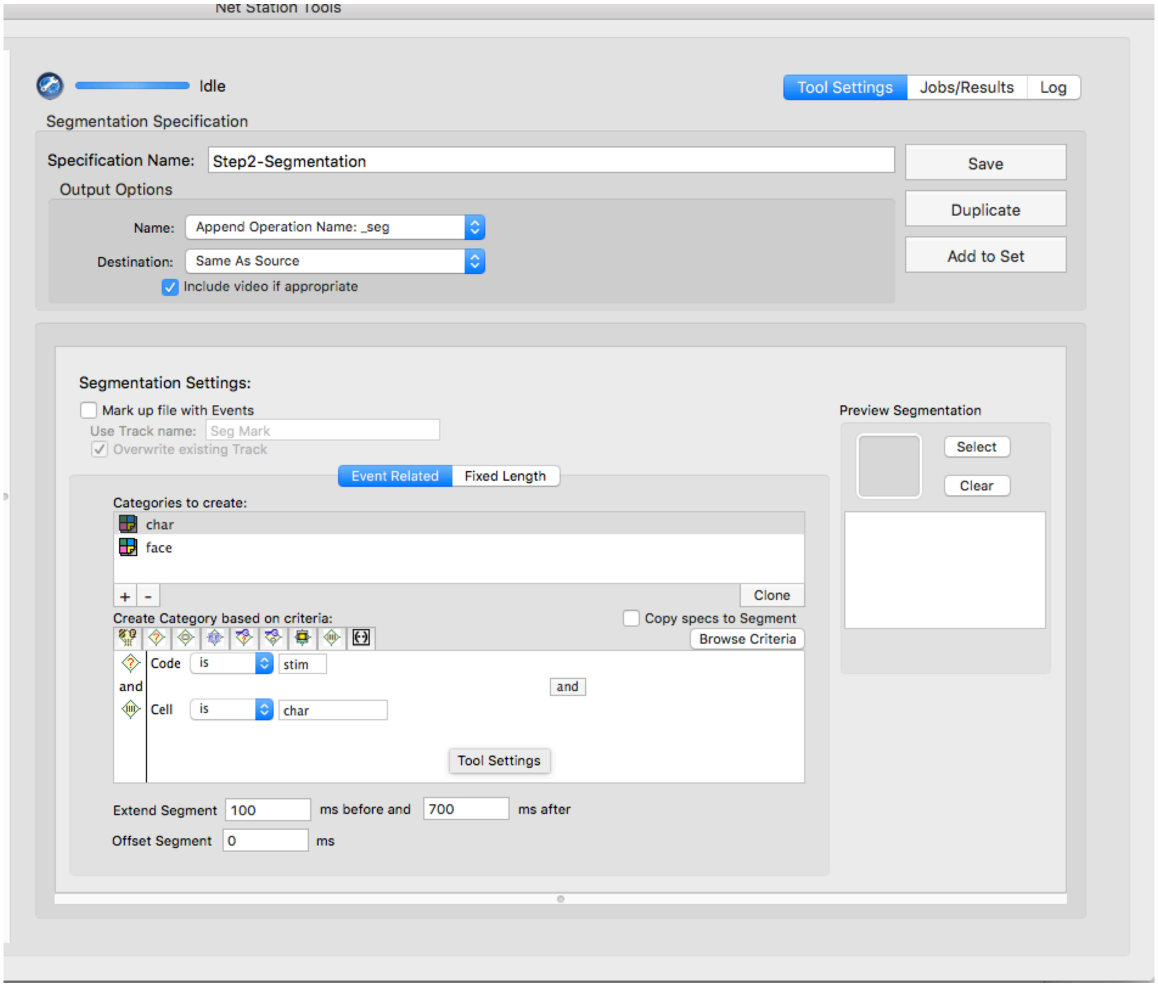
□ Auto set name: 0.10 Hz Highpass

☑ Highpass 0.10

□ Lowpass

□ Notch

Step 2: Segment. We do segmentation here:



Segmentation Specification

Specification Name: Step2-Segmentation

Output Options

Name: Append Operation Name: _seg

Destination: Same As Source

☑ Include video if appropriate

Segmentation Settings:

☐ Mark up file with Events

Use Track name: Seg Mark

☑ Overwrite existing Track

Categories to create:

- char
- face

Create Category based on criteria:

☐ Copy specs to Segment

Code is stim

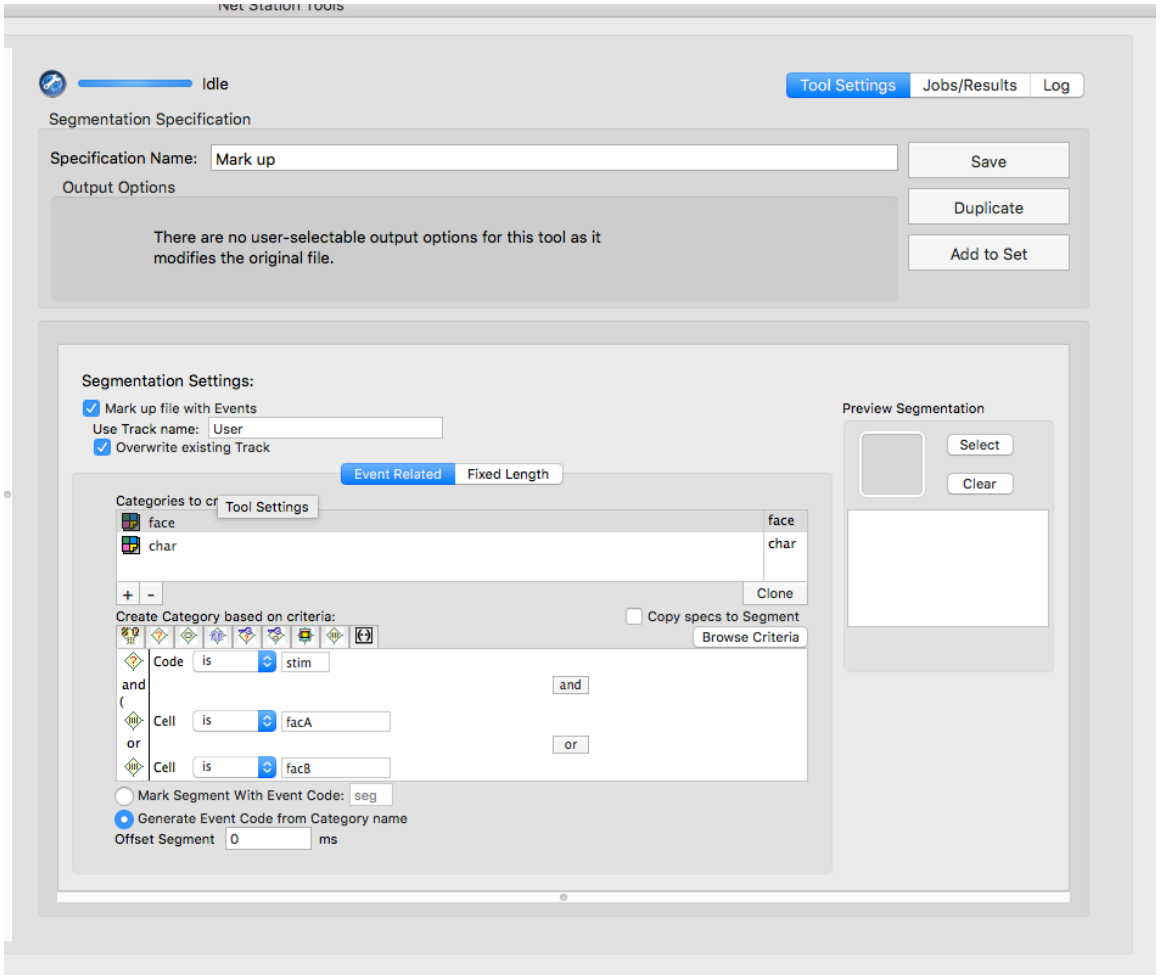
and

Cell is char

Extend Segment 100 ms before and 700 ms after

Offset Segment 0 ms

Step 3: Mark up events. As we described in the topic 1, we change the names of the stimulus into different names so that we can analyze them better in EEGLAB



Net Station Tools

Idle

Tool Settings | Jobs/Results | Log

Segmentation Specification

Specification Name: Mark up

Save

Duplicate

Add to Set

Output Options

There are no user-selectable output options for this tool as it modifies the original file.

Segmentation Settings:

☑ Mark up file with Events

Use Track name: User

☑ Overwrite existing Track

Event Related | Fixed Length

Categories to cr Tool Settings

face	face
char	char

+ - Clone

Create Category based on criteria:

□ Copy specs to Segment

Browse Criteria

Code is stim

and

(

Cell is facA

or

Cell is facB

and

or

○ Mark Segment With Event Code: seg

● Generate Event Code from Category name

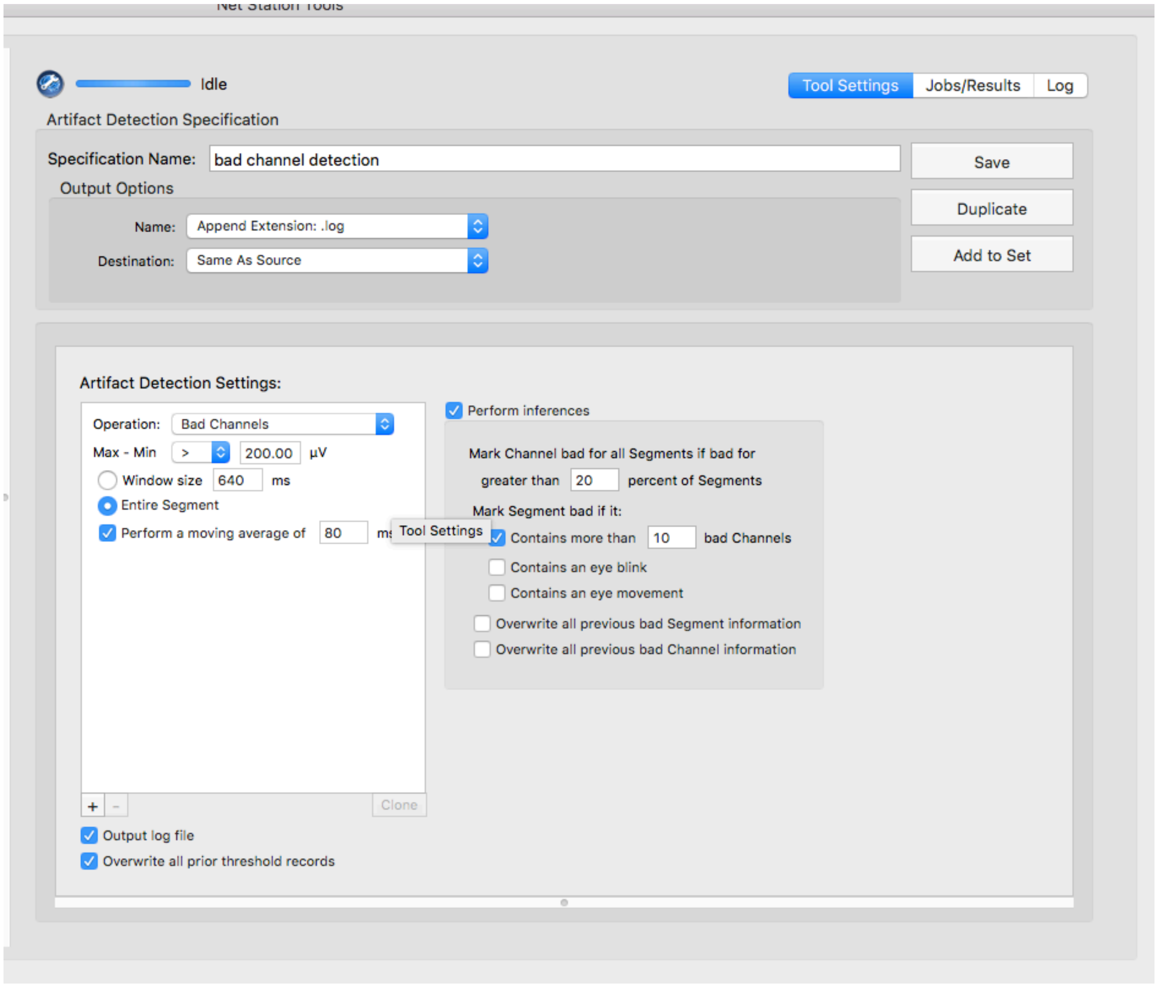
Offset Segment 0 ms

Preview Segmentation

Select

Clear

Step 4-artifact detection. We only do bad channel detection here.



Artifact Detection Specification

Specification Name: bad channel detection

Output Options

Name: Append Extension: .log

Destination: Same As Source

Artifact Detection Settings:

Operation: Bad Channels

Max - Min > 200.00 µV

○ Window size 640 ms

● Entire Segment

☑ Perform a moving average of 80 ms

☑ Perform inferences

Mark Channel bad for all Segments if bad for greater than 20 percent of Segments

Mark Segment bad if it:

☑ Contains more than 10 bad Channels

□ Contains an eye blink

□ Contains an eye movement

□ Overwrite all previous bad Segment information

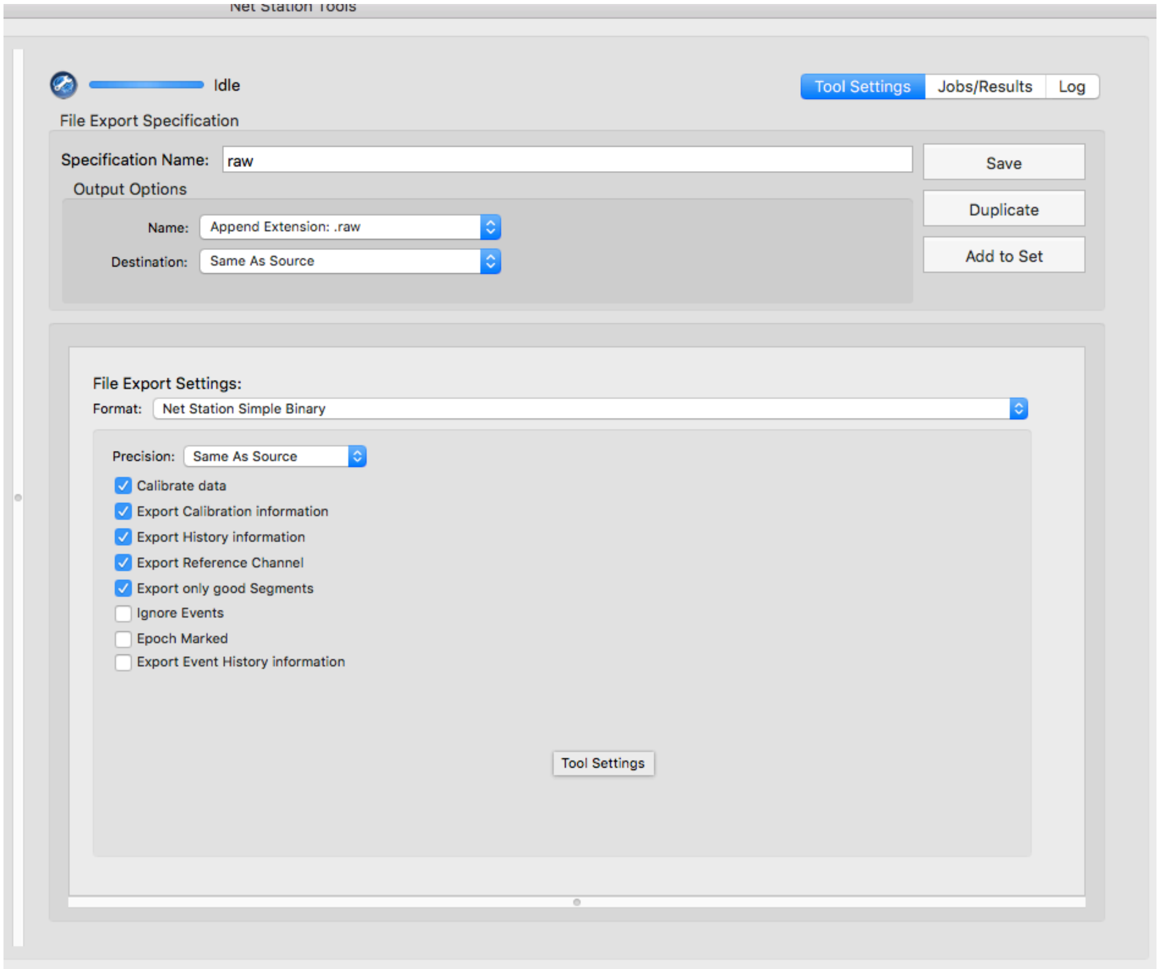
□ Overwrite all previous bad Channel information

☑ Output log file

☑ Overwrite all prior threshold records

Step5 bad channel replacement. Since you do not need to set any parameters to this tool, just create one and run it to the file produced by Step 4.

Step 6. Then repeat the steps in Topic 2. The only difference here is the settings of the option of “file export” tools. We do not need to export the bad segments which are recognized by Nestation: Check the option “export only good segments”



File Export Specification

Specification Name: raw

Output Options

Name: Append Extension: .raw

Destination: Same As Source

File Export Settings:

Format: Net Station Simple Binary

Precision: Same As Source

☑ Calibrate data
☑ Export Calibration information
☑ Export History information
☑ Export Reference Channel
☑ Export only good Segments
☐ Ignore Events
☐ Epoch Marked
☐ Export Event History information

Topic 4 ICA analysis in EEGLAB

Sometimes the customer want to make use of the ICA function in EEGLAB, for example, to remove the artifacts. Netstaion will give up a epoch when there is an Eye blink or Eye movement in it. However, in some experiments, the trail will be last very long(several seconds even longer). So they just want to retain the epoch.

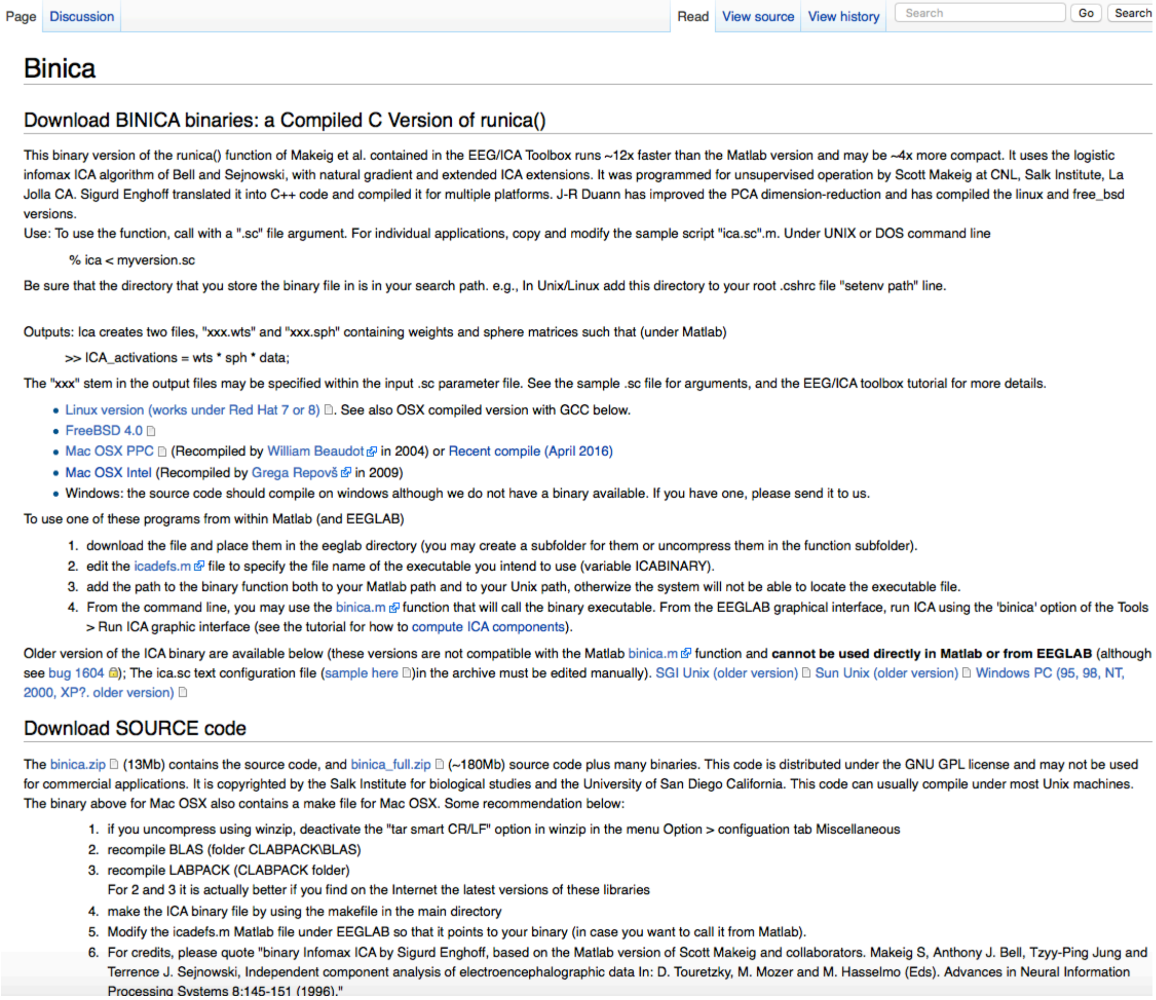
After we follow the step in Topic 2 or Topic 3, we can do ICA analysis in EEGLAB. Here we do not teach how to use ICA in EEGLAB, please refer to the EEGLAB manual which can be downloaded from the official website. We introduce how to run binica to EGI datas.

There are 4 ICA methods in EEGLAB. The default one is runica. Binica is runica's binary code, but can run much faster than Runica.

Step 1 download Binica from the website.

Go to https://sccn.ucsd.edu/wiki/Binica .

From there, you can download the most recent pacakages. Since EGI use Mac as its workstation, we only introduce how to run Binica on Mac. According to CPU type, download the correct one:



Binica

Download BINICA binaries: a Compiled C Version of runica()

This binary version of the runica() function of Makeig et al. contained in the EEG/ICA Toolbox runs ~12x faster than the Matlab version and may be ~4x more compact. It uses the logistic infomax ICA algorithm of Bell and Sejnowski, with natural gradient and extended ICA extensions. It was programmed for unsupervised operation by Scott Makeig at CNL, Salk Institute, La Jolla CA. Sigurd Enghoff translated it into C++ code and compiled it for multiple platforms. J-R Duann has improved the PCA dimension-reduction and has compiled the linux and free_bsd versions.
Use: To use the function, call with a ".sc" file argument. For individual applications, copy and modify the sample script "ica.sc".m. Under UNIX or DOS command line

```
% ica < myversion.sc
```

Be sure that the directory that you store the binary file in is in your search path. e.g., In Unix/Linux add this directory to your root .cshrc file "setenv path" line.

Outputs: Ica creates two files, "xxx.wts" and "xxx.sph" containing weights and sphere matrices such that (under Matlab)

```
>> ICA_activations = wts * sph * data;
```

The "xxx" stem in the output files may be specified within the input .sc parameter file. See the sample .sc file for arguments, and the EEG/ICA toolbox tutorial for more details.

- Linux version (works under Red Hat 7 or 8). See also OSX compiled version with GCC below.
- FreeBSD 4.0
- Mac OSX PPC (Recompiled by William Beaudot in 2004) or Recent compile (April 2016)
- Mac OSX Intel (Recompiled by Grega Repovš in 2009)
- Windows: the source code should compile on windows although we do not have a binary available. If you have one, please send it to us.

To use one of these programs from within Matlab (and EEGLAB)

1. download the file and place them in the eeglab directory (you may create a subfolder for them or uncompress them in the function subfolder).
2. edit the icadefs.m file to specify the file name of the executable you intend to use (variable ICABINARY).
3. add the path to the binary function both to your Matlab path and to your Unix path, otherwize the system will not be able to locate the executable file.
4. From the command line, you may use the binica.m function that will call the binary executable. From the EEGLAB graphical interface, run ICA using the 'binica' option of the Tools > Run ICA graphic interface (see the tutorial for how to compute ICA components).

Older version of the ICA binary are available below (these versions are not compatible with the Matlab binica.m function and **cannot be used directly in Matlab or from EEGLAB** (although see bug 1604); The ica.sc text configuration file (sample here)in the archive must be edited manually). SGI Unix (older version) Sun Unix (older version) Windows PC (95, 98, NT, 2000, XP?. older version)

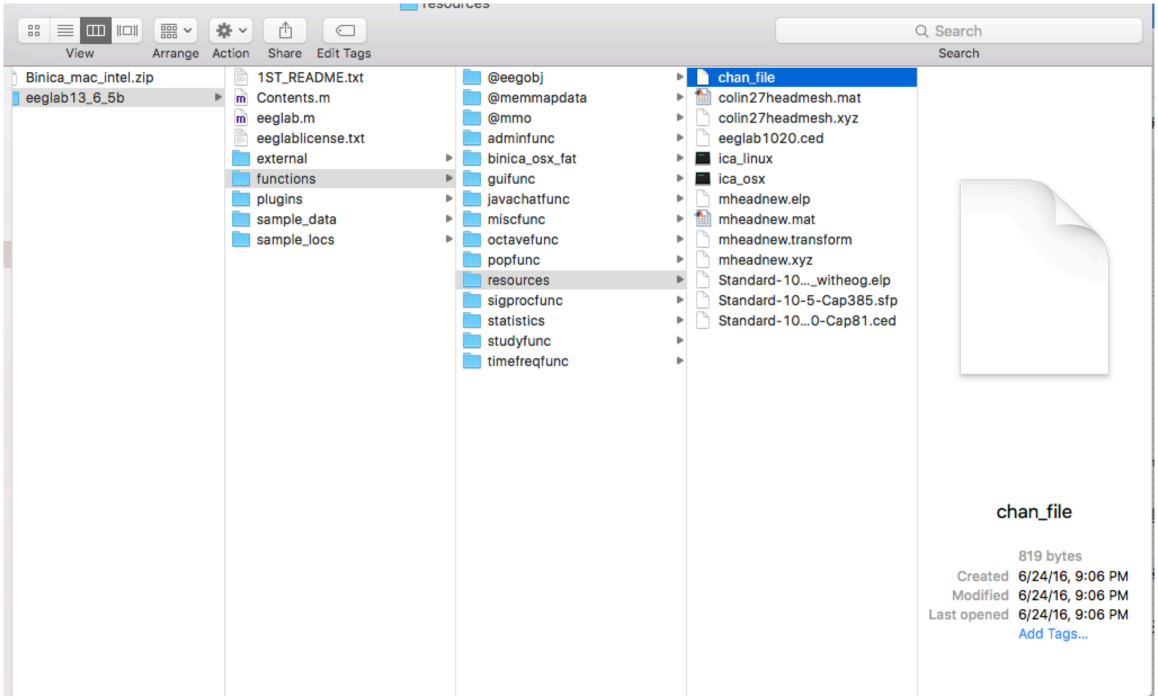
Download SOURCE code

The binica.zip (13Mb) contains the source code, and binica_full.zip (~180Mb) source code plus many binaries. This code is distributed under the GNU GPL license and may not be used for commercial applications. It is copyrighted by the Salk Institute for biological studies and the University of San Diego California. This code can usually compile under most Unix machines. The binary above for Mac OSX also contains a make file for Mac OSX. Some recommendation below:

1. if you uncompress using winzip, deactivate the "tar smart CR/LF" option in winzip in the menu Option > configuation tab Miscellaneous
2. recompile BLAS (folder CLABPACK\BLAS)
3. recompile LABPACK (CLABPACK folder)
   For 2 and 3 it is actually better if you find on the Internet the latest versions of these libraries
4. make the ICA binary file by using the makefile in the main directory
5. Modify the icadefs.m Matlab file under EEGLAB so that it points to your binary (in case you want to call it from Matlab).
6. For credits, please quote "binary Infomax ICA by Sigurd Enghoff, based on the Matlab version of Scott Makeig and collaborators. Makeig S, Anthony J. Bell, Tzyy-Ping Jung and Terrence J. Sejnowski, Independent component analysis of electroencephalographic data In: D. Touretzky, M. Mozer and M. Hasselmo (Eds). Advances in Neural Information Processing Systems 8:145-151 (1996)."

Step 2 install Binica to your EEGLAB

Uncompress the zip file to the functions folder under EEGLAB folders. Then copy the “ica_osx” file to resource folder.



Step 3 : go to sigprocfunc folder and fine icadefs.m,modify the following sentence

from

```
ICABINARY = fullfile(eeglab_p, 'functions', 'resources',  'ica_linux');
```

To

```
ICABINARY = fullfile(eeglab_p, 'functions', 'resources',  'ica_osx');

Save and close.
```