**EEG preprocessing for 128Ch EGI (Under construction)**

brxEGIpreprocessTool 폴더 안에 있는 파일들로 프리프로세싱

addpath(eeglab2020.0 directory without using genpath) = eeglab2020.0만 경로에 올려놓는다.

그리고 brx\_preprocessScript.m 파일을 돌린다. 이렇게 되면 아티팩트가 제거된 clean raw data가 각 서브젝트 폴더 안 preproc 폴더 안에 subject\_220302\_postADJUST.set 파일로 저장된다.

brx\_openNpreprocess.m 마지막에 ADJUST 알고리즘이 돌아가는데 오리지널은 수동이어서 약간의 수정을 한 pop\_ADJUST\_interface\_brx.m 파일을 불러서 자동화한다. 하지만 마지막 각 서브젝트 ica reject된 component 정보가 report.txt 파일로 저장이 되기 때문에 각 서브젝트 분석이 끝날때마다 subject\_220302\_report.txt로 이름을 바꿔주고 각 서브젝트의 preproc 폴더 안에 넣어준다.

이 과정이 끝나면 subject\_220302\_postADJUST.set 파일이 preproc 폴더에 저장된다.

clean\_artifacts 라는 함수를 쓰는게 프리프로세싱 과정 중에 들어있는데 이것은 artifact subspace reconstruction이라는 방법인데 이 경우 채널 반응이 이상한 채널과 시간을 짤라내기 때문에 오리지널 merged data시간이랑 달라질 수 있다. 오리지널 각 블록 데이타마다 끝나는 시간은 EEG.endIdx 에 들어있다. 그리고 EEG.etc.clean\_sample\_mask 는 0과 1로 이루어진 숫자들로서 0으로 표시된 시간은 리젝트된 시간이다. 그래서 어떤 경우에는 총 블록 데이터길이보다 짧아져 있을 수 있는데 이것을 원래 길이로 바꾸는 작업을 nan을 채워넣는 방식으로 만든다.

이 과정이 끝나면 subject\_220302\_postFilt\_NaN.set 파일을 eeglab에서 열고 epoch을 한다. 이 경우 조건에 맞는 이벤트 코드에서 가장 가까운 DIN1 시간에 맞춰 파일을 에포크를 해준다. 이 DIN1 시간이 실제 화면에 자극이 나타난 시간이다. 이 때 nan이 epoch에 포함되어 있으면 그 시행은 버린다.

=======================================================

뇌파 데이터를 Netstation으로 레코딩할 때 (즉, Psychtoolbox에서 Netstation 함수를 불러 이벤트 트리거할 때), 반드시 실험 조건들 (나중에 epoch될 실험에 쓰인 모든 조건들)마다 4개의 문자로 된 이벤트 마커, 즉 c001, c002, c003, c004, ……., c0xx 이렇게 보내도록 한다. 나머지 행동 반응이나 실제로 보여준 값 등등의 변수들은 option variable로 c001과 같은 이벤트 마커를 보낼 때 추가하여 보내거나, 아예 다른 4개의 문자열로 된 이벤트 마커를 (e.g. RESP ) 보낼 수 있다. 실험 레코딩할 때 모두 이와 같은 방식으로 보내야만 각 개인의 실험에 상관없이 동일한 preprocessing 코드 이용이 가능하다.

When matlab starts, addpath of all toolboxes in the below. (OR copy and paste into startup.m file if you don’t wanna do the following overtime you run).

Use the eeglab2020\_0\_2 version or after this version.

Put all the following plug-ins under plugins subfolders of EEGLAB. All of them can be added using EEGLAB gui except FastICA. Basically you can use the most recent versions of toolboxes.

1. PREP toolbox (<https://github.com/VisLab/EEG-Clean-Tools>) for line noise
2. Clean\_rawdata (<https://github.com/sccn/clean_rawdata>) for ASR
3. FastICA\_25 for ICA
4. ICLabel1.3
5. TESA1.0.1
6. ERPLAB8.10 for extracting frequency band activities (delta, theta, alpha, etc)
7. Fieldtrip-lite20201214 or after this version

If all the above plug-ins are under plugins subfolder EEGLAB, you just need to run addpath(EEGLAB path)

Then run addpath(ssProcessToolfaster) 🡪 This one’s will be used for parsing events based on evt files. 각각 c001, c002 등등 이벤트 마커 시간과 끝나는 시간을 LED 시각에 맞춰 시행들을 epoch 하기 위해 필요한 파일이 들어 있음.

1. Merge all the raw data (nnn\_20230214\_1~6.mff) across all blocks using EEGLAB. If there are 6 blocks of recording, \_1\_, \_2\_, \_3\_, …, \_6\_ will be your file’s postfix with recording time.
2. Highpass filtering of raw data. Van Driel et al. (Journal of Neuroscience Method, 2021, 352, 109080) and de Cheveigne & Arzounian (Neuroimage, 2018, 172, 903-912, <http://audition.ens.fr/adc/NoiseTools/>) demonstrate that highpass filtering generates artifacts in multivariate decoding. So use robust detrend (ORDER = 10) for highpass filtering, which will make clean\_rawdata work less efficient to clean noise. But it’s the sacrifice that we make in order not to distort the decoding result. So when you decode raw data, please play around with downsampling and sliding window size to get the decoding result. This robust detrend is the recommendation that we make. But in most case, pop\_basicfilter function from ERPLab is gonna be fine with butterworth filter order 2 for highpass filtering. You can set RobDetr = ‘off’ for pop\_basicfilter. RobDeter = ‘on’ is to use Robust Detrend method.
3. Clean\_rawdata cleans out most channel noise using artifact subspace reconstruction.

For SSVEP (or EEG) experiment preprocess analysis pipeline

When matlab starts, addpath of all toolboxes in the below. (OR copy and paste into startup.m file if you don’t wanna do the following overtime you run). Fieldtrip should be added first before EEGLAB.

Right now, I’m using old EEGLAB and Fieldtrip on matlab 2011b version.

addpath( '/Volumes/MRI/toolbox/ssPreprocessTool/fieldtrip' ); % for SSVEP analysis

fieldtripdefs; % for SSVEP analysis

addpath(genpath( '/Volumes/MRI/toolbox/ssPreprocessTool/eeglab' ));

addpath(genpath( '/Volumes/MRI/toolbox/ssPreprocessToolfaster' ));

addpath( '/Volumes/MRI/toolbox/ssPreprocessTool/ss\_faster\_toolbox' );

In matlab command line, go to, for example (OriAvgNoFlicker project), /Volumes/MRI/Projects/eky/OriAvgNoFlicker/raw/large\_var/khs\_20160609.

>> cd /Volumes/MRI/toolbox/ssPreprocessTool/raw/large\_var/khs\_20160609

OR just drag that folder into matlab command line.

This folder should have mrC and preproc folders. Just create two empty folders and name them as those two names.

>>mkdir preproc

>>mkdir mrC

Preprocessing data will be saved into those two folders.

\*\*\* Collapse evt files with Excel \*\*\*

Since our EGI Netstation (GES 400) spits out event file (evt file) in a format slightly different from old GES 300, we need to work on evt file using Excel program before running matlab analysis.

Depending on how many blocks we run for each variance condition (large\_var and small\_var), there will be 2 or 4 vet files.

(1) Duplicate the 1st one. And change its name name same as its folder name (usually subjectName\_2016xxxx.evt). Open it with Excel. Make the 2nd column (Label) empty just below Label until the end row. Also make the 3rd column (Description) full with something like “Stimulus Event” just below Description. Leave this evt file open. You will repeat this procedure for the 2nd and 3rd column for the rest of evt files.

(2) Open another evt file (2nd one for that variance condition). Run “Find and Replace” function in excel. Find “ \_[1] “ and replace it with “ \_[2] “. Change the 2nd (empty) and 3rd (Stimulus Event) columns as in (1). Copy all contents from the 4th rows and paste them into just below the end of the 1st evt file that was left open. Basically we are appending 2nd evt file into the 1st one.

(3) Open the rest of evt file for that condition (large\_var or small\_var), then do the same thing as (2).

(4) Once you’re done, click the save button of the 1st evt file (the one you duplicated and named “subjectName\_2016xxx.evt”). Close this file. It will ask if you want to save it. Since we already saved, you don’t have to save it again. But don’t save other evt files that we corrected when you closed them all so that we can keep the original.

Related to this procedure, we also rename our raw date file like the following. subjectName\_2016xxxx001.raw, subjectName\_2016xxxx002.raw, subjectName\_2016xxxx003.raw, subjectName\_2016xxxx004.raw and so on. If we recorded raw data with pause command, these files would’ve been created automatically. event file too. But since we alternate two variance conditions. We have to do this by hand.

\*\*\* Now in matlab \*\*\*

Again you have to be in /Volumes/MRI/Projects/eky/OriAvgNoFlicker/raw/large\_var/khs\_20160609/

1. First check trial lookup table.

>> [ tLut, tLutI ] = ssLongTrialLut\_SerialOriAvg( ‘khs\_20160609.evt', 500 )

See if number of trials that this m file spits out matches your original number of trials that you are supposed to see.

2. Check bad channels

>> ssImpedanceNet400( 'khs\_2016060900X\_IMP.txt, 50)

Here X means 1,2,3, 4 etc… See which channels have impedance > 50 kilo-ohm. You will put them in the number 3 procedure below.

3. Run ssProcessForMrCurrent.m file, Before running that, Make sure if ssLongTrialLut\_SerialOriAvg is called in ssProcessSession.m file. If it is, then type next command.

>> ssProcessForMrCurrent( '/Volumes/MRI/Projects/eky/OriAvgNoFlicker/raw/large\_var/khs\_20160609', 'khs\_20160609', 'preproc', 4, 'mrC', 1000/600, 0, 'badChan', {[10 54 17], [44], [], [100 57 78 91]})

For ‘barchan’ argument inside {}, you will put channel numbers that was spitted out in number 2 procedure for each block of 4.

so ssProcessForMrCurrent('folder of your current subject for your project', 'preproc: name of files inside that folder (it has to be consistent. in this example, ll\_20140122)', 'preprocessing ICA data will be saved in this empty folder that you just created above', number of blocks (ssExportData from Netstaion will create 'consistent\_name'001~004.raw depending on how many blocks you run, 'mrC: Axx\_c001~c008.mat files will be saved here', F1, F2, 'badChan', { [10 20 …]:1st block bad channels, [21 44 ..]:2nd block bad channels)

help ssProcessForMrCurrent and ssSessionParams for input arguments.

Once this preprocessing is done, copy Axx\_c001~c008.mat files from 'mrC' folder into /Volumes/MRI/eky/OriAvgNoFlicker/ibs00xx/Exp\_MATL\_HCN\_128\_Avg/ (These are not created yet cause we don’t have head models yet).

Also copy digitization files (elp and/or hsp) in /Volumes/MRI/eky/OriAvgNoFlicker/ibs00xx/Polhemus/

Read wiki page (not created yet) MNEsuite\_manual\_for\_IBS for mrCurrent analysis

Also for more about mrCurrent (how to install etc), look at this wiki page (Chapter\_9:\_mrCurrent\_v.9 not created).