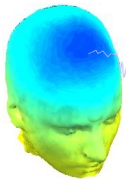
EEG Preprocessing in EEGLAB (cont.)

EEGLAB

Ramon Martinez-Cancino, PhD
Arnaud Delorme, PhD
Johanna Wagner, PhD

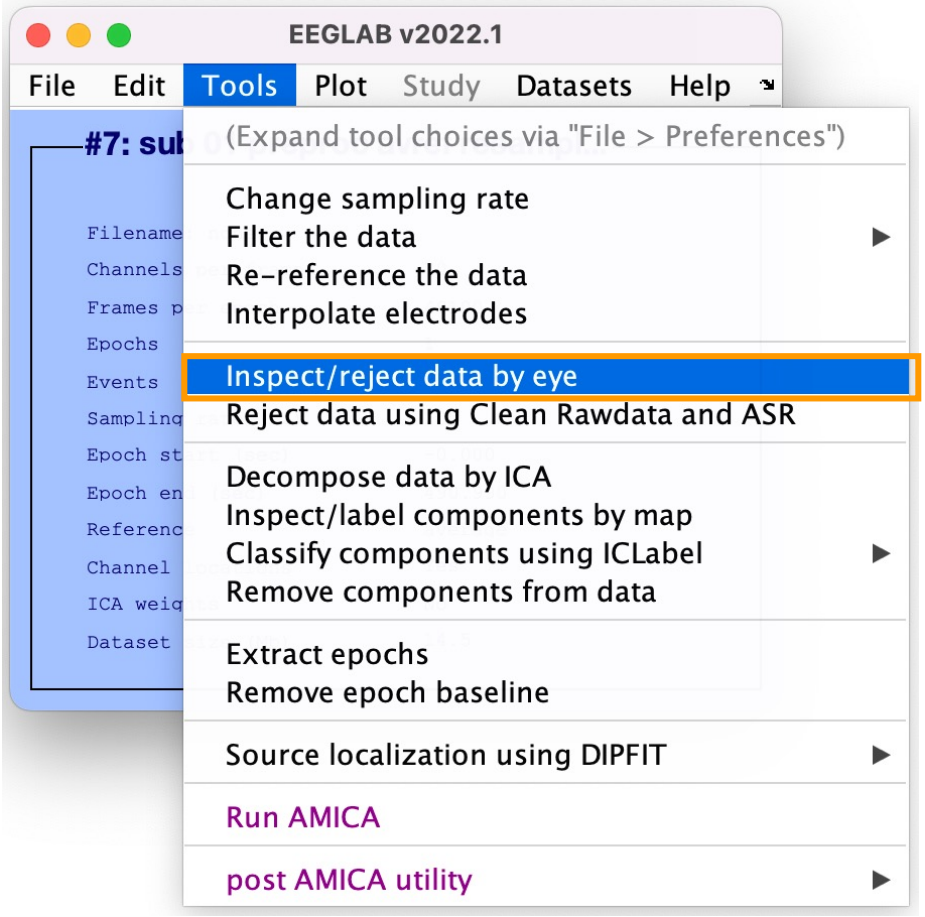


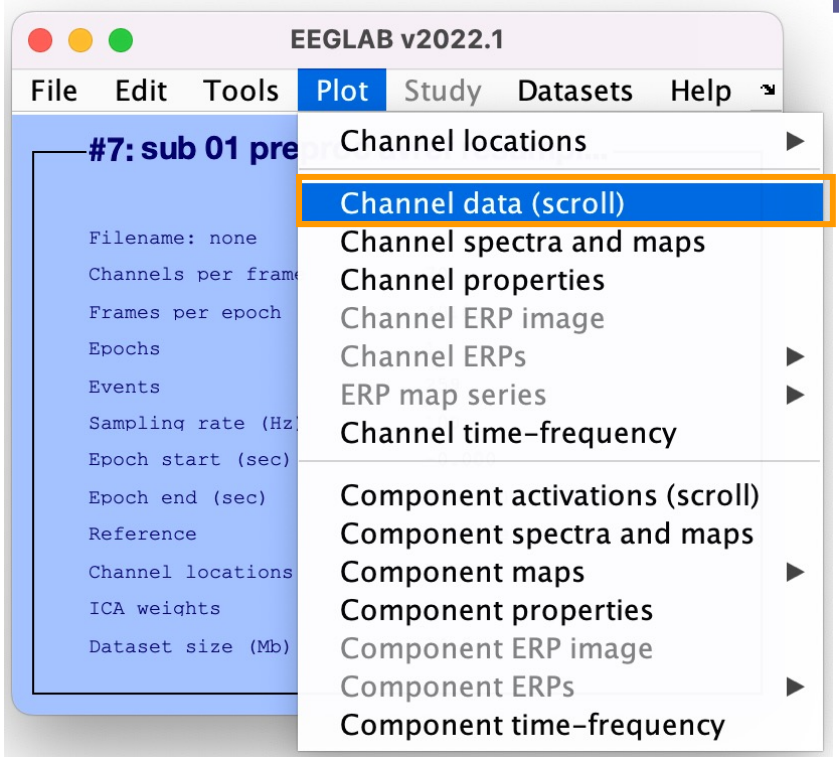
Data Cleaning for ICA

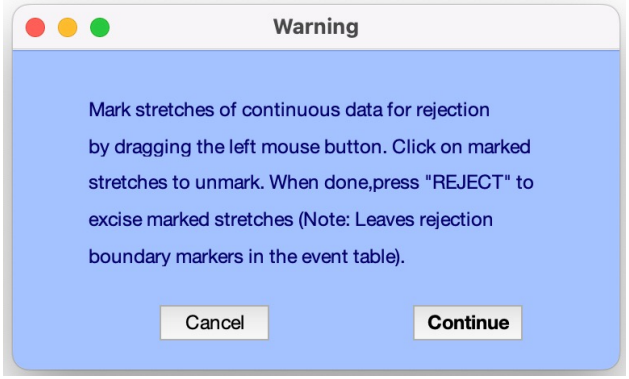
Variant 1: Continuous Data

Reject continuous data

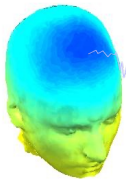
Equivalent

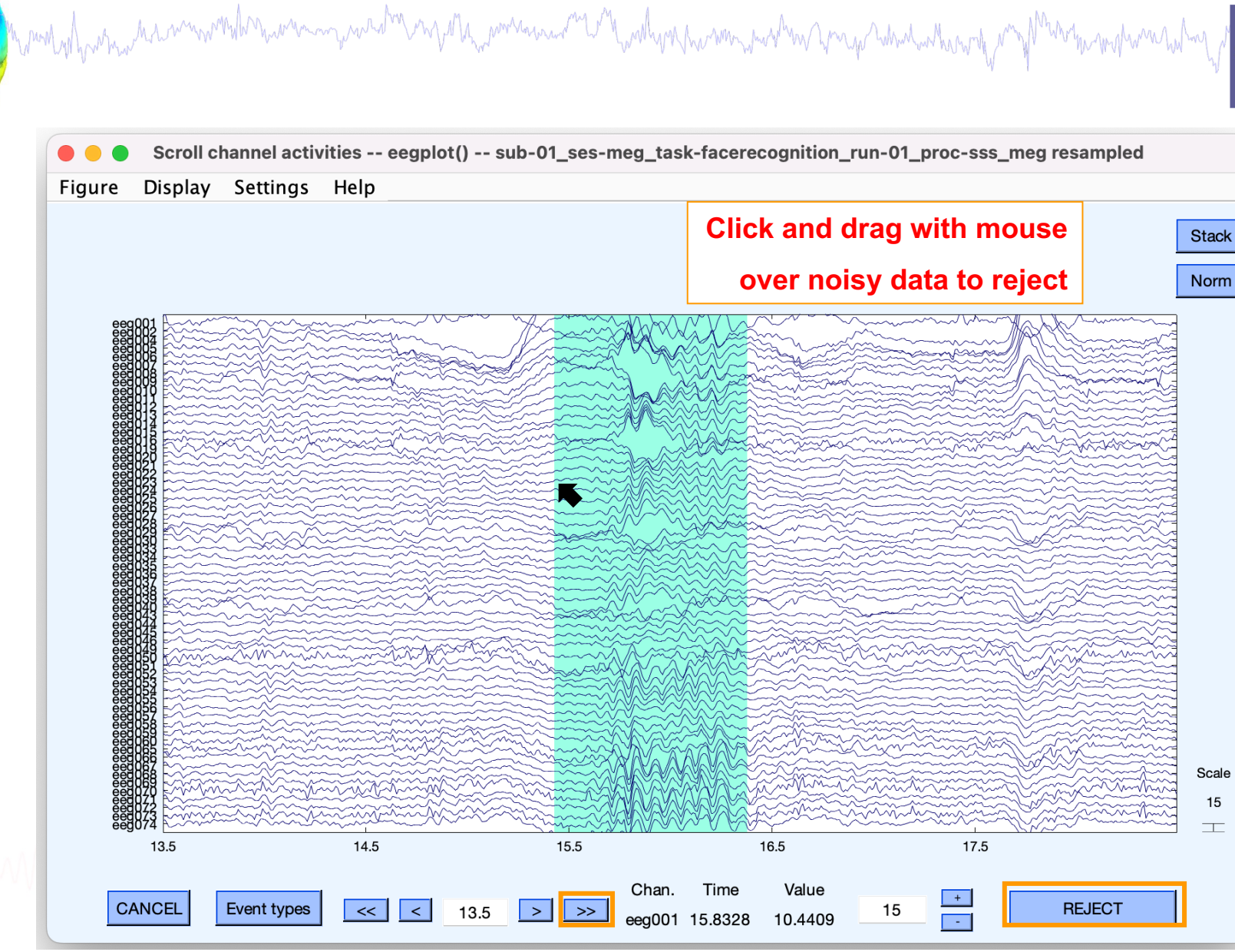






Reject continuous data

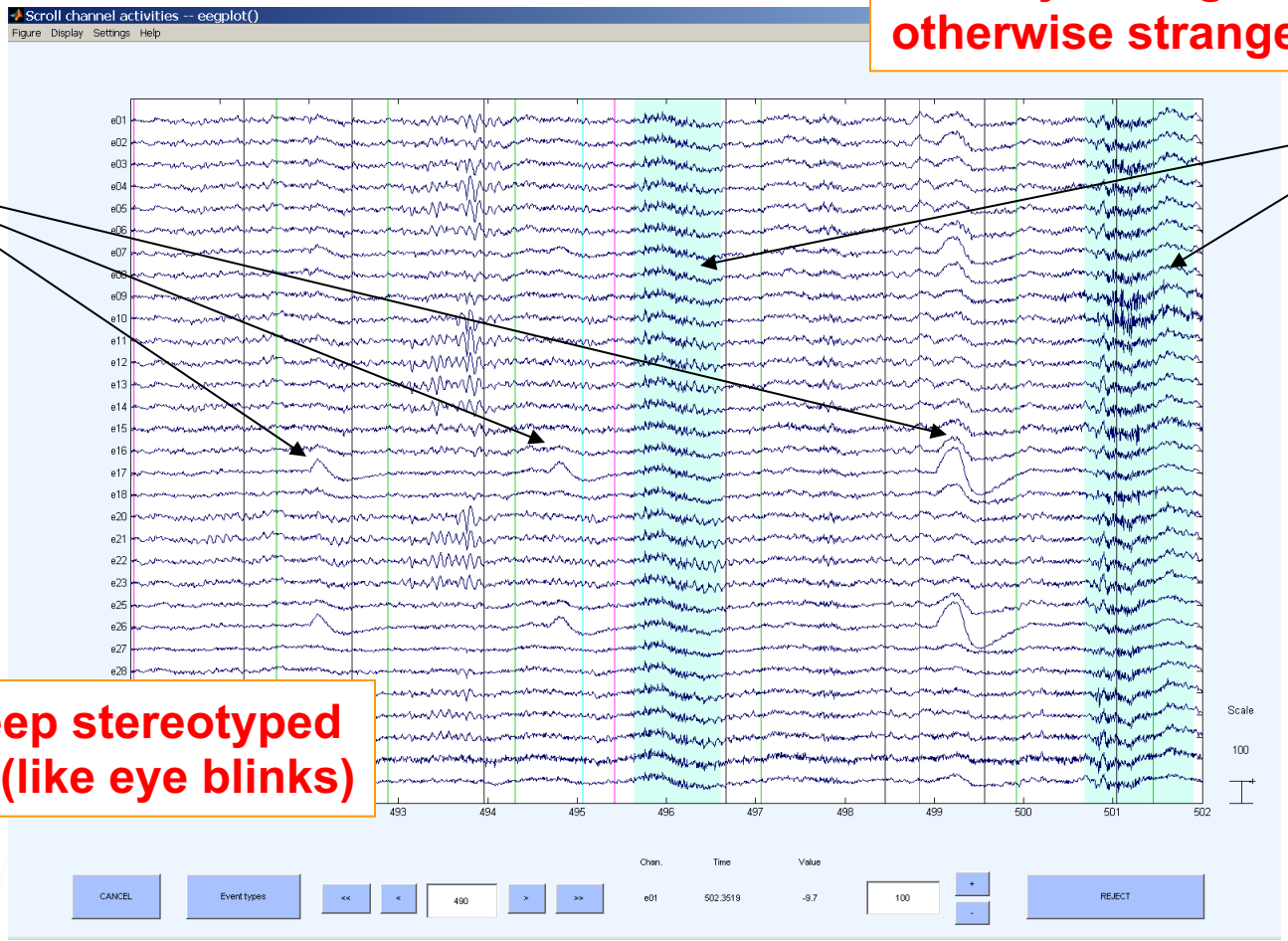




Rejecting data for ICA

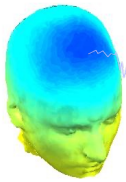
To prepare data for ICA:

Reject large muscle or otherwise strange events...

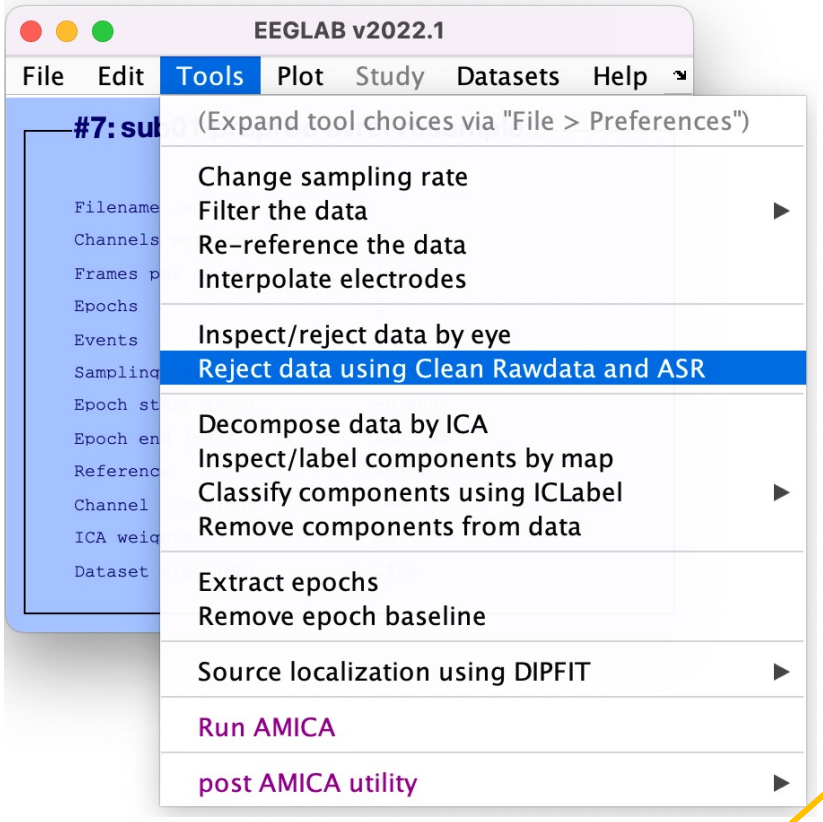


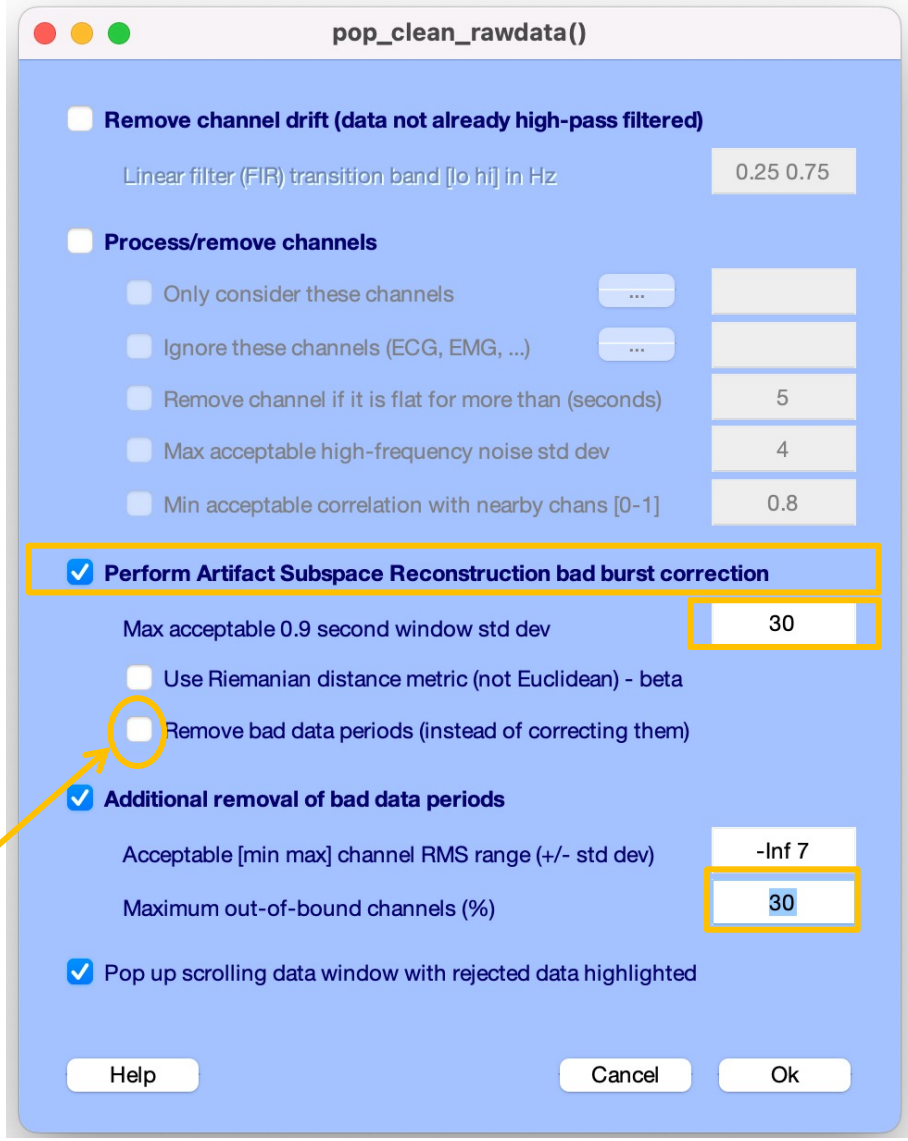
... but keep stereotyped artifacts (like eye blinks)

Automatic rejection of continuous data



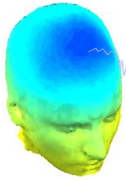
Clean_rawdata plugin of EEGLAB





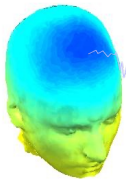
uncheck

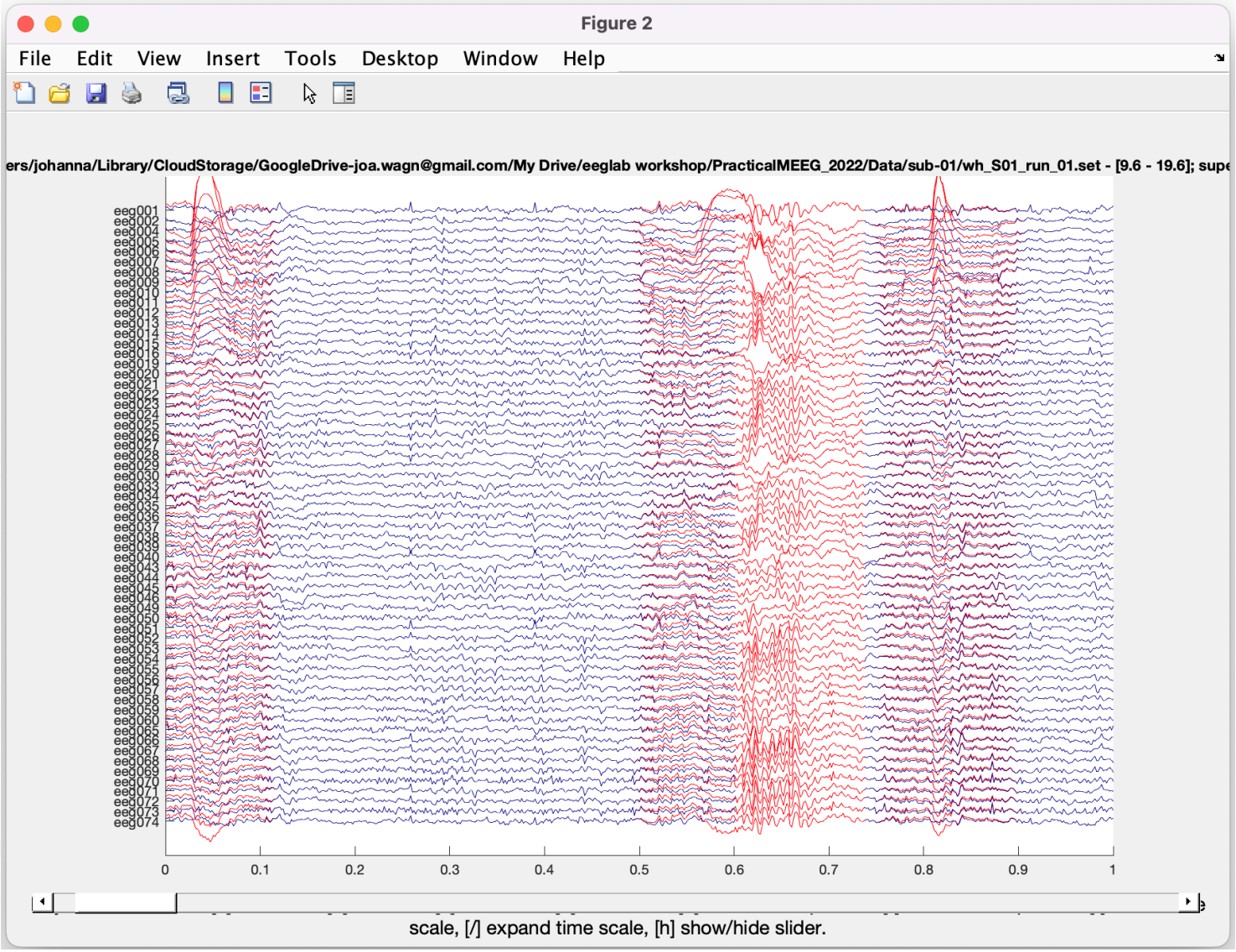
Automatic rejection of continuous data



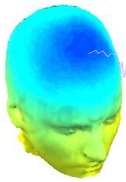
```
EEG = clean_artifacts( EEG, 'Highpass', 'off',...
                'ChannelCriterion', 'off',...
                'LineNoiseCriterion', 'off',...
                'BurstCriterion', 30,...
                'WindowCriterion',0.3);
```

Automatic rejection of continuous data





Pre-processing pipeline



Collect EEG data

Import into EEGLAB

Import event markers and channel locations

Re-reference/ down-sample (if necessary)

High pass filter (~.5 – 1 Hz)

Examine raw data

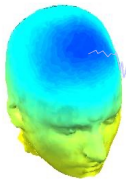
Identify/reject bad channels

Reject large artifact time points

Run ICA and reject components

Done

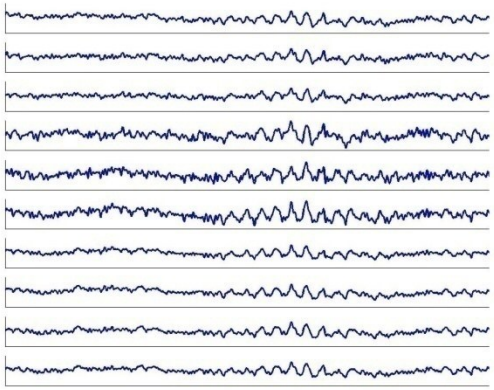
Independent Component Analysis



x = scalp EEG

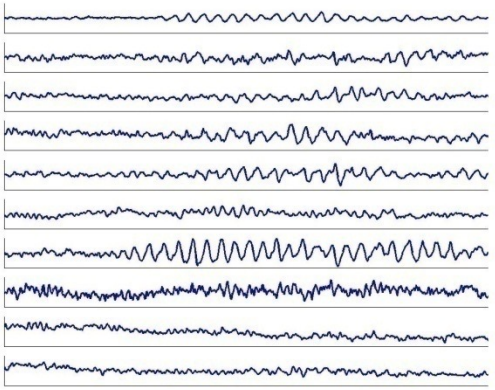
W = unmixing matrix

u = sources



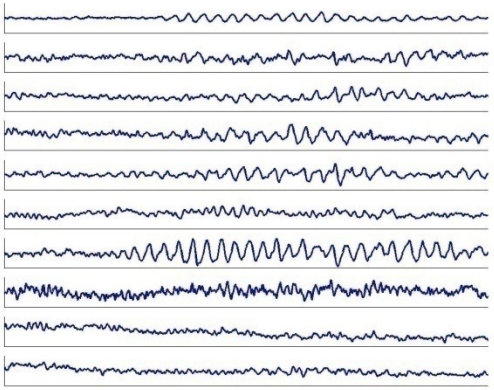
W*x = u

ICA



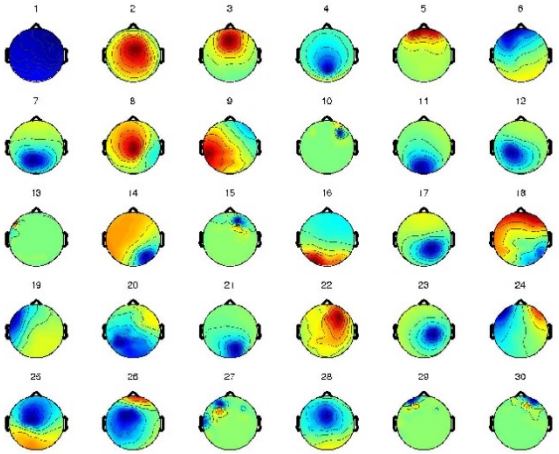
x = W^{-1}*u

u = sources

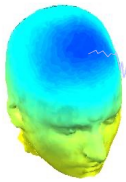


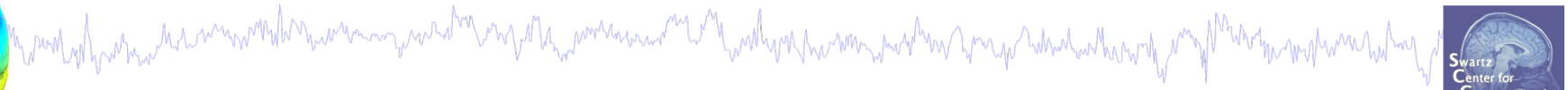
W^{-1} (scalp projections)

*



Running ICA





Swartz Center for Computational Neuroscience

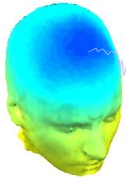
Task 1

Run ICA

Task 2

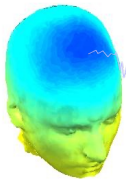
Evaluating ICA Components

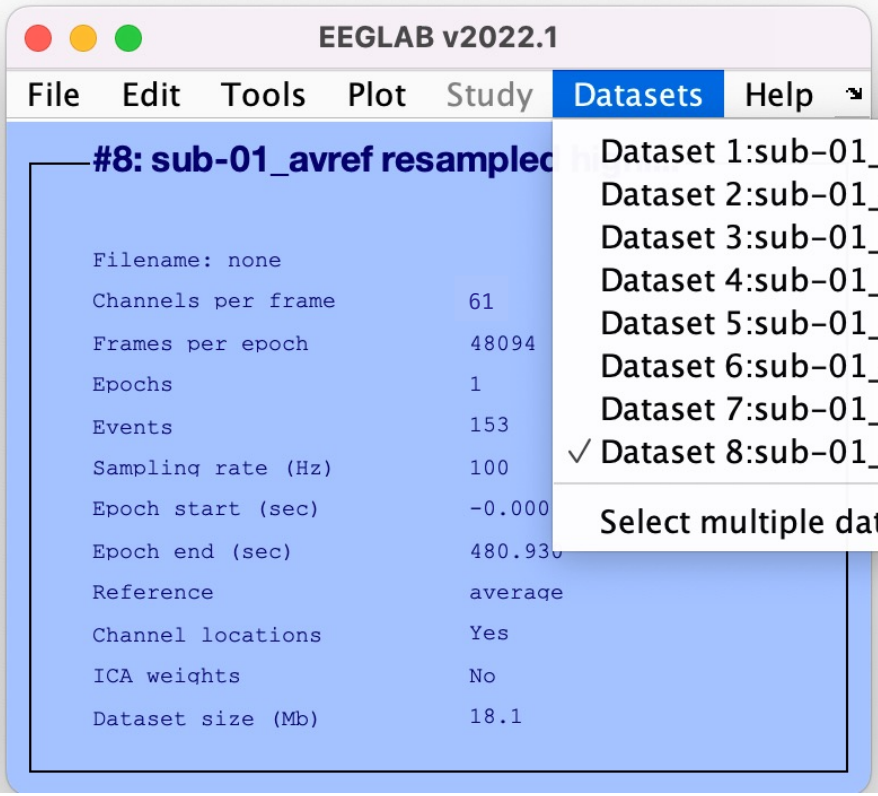
“Secrets” to a good ICA decomposition



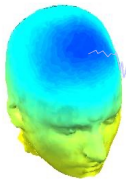
- Garbage in… garbage out (it’s not magic)
- Remove large, non-stereotyped artifacts
- Do you have enough data? (based mostly on time, not frames)
- High-pass filter to remove slow drifts (no low-pass filter needed)
- Remove bad channels
- Data must be in double precision (not single)
- Data should be full rank

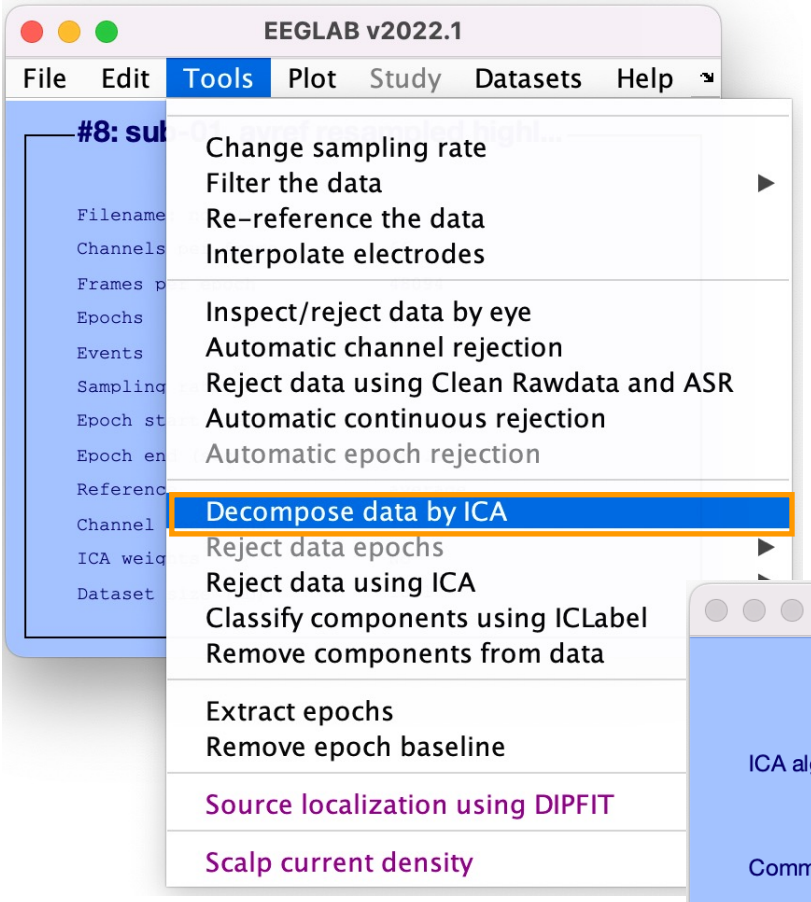
Preprocessed Dataset



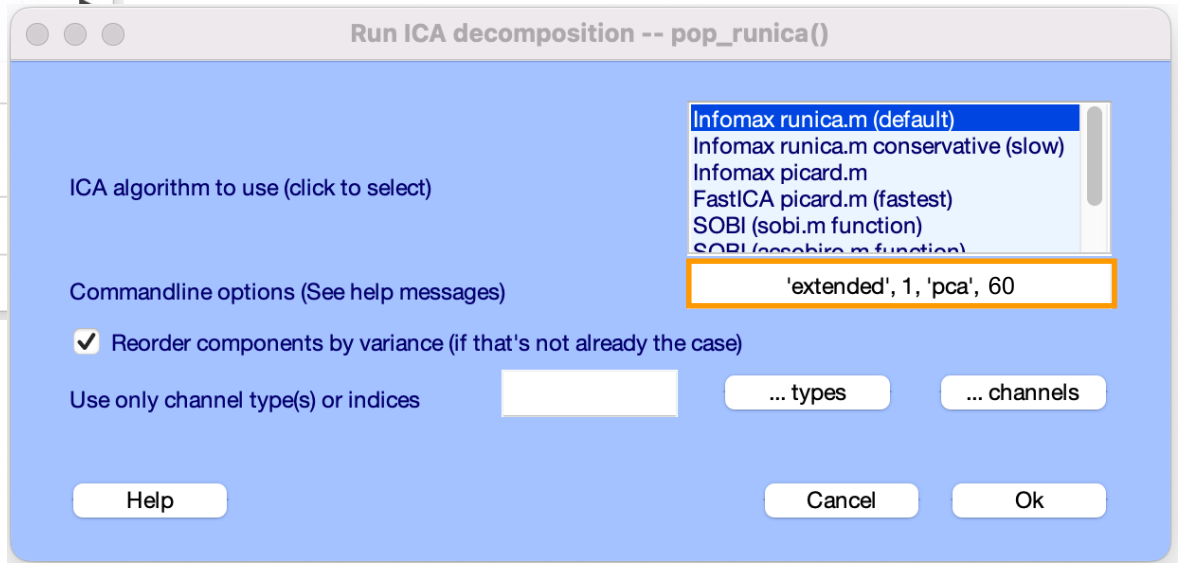


ICA options

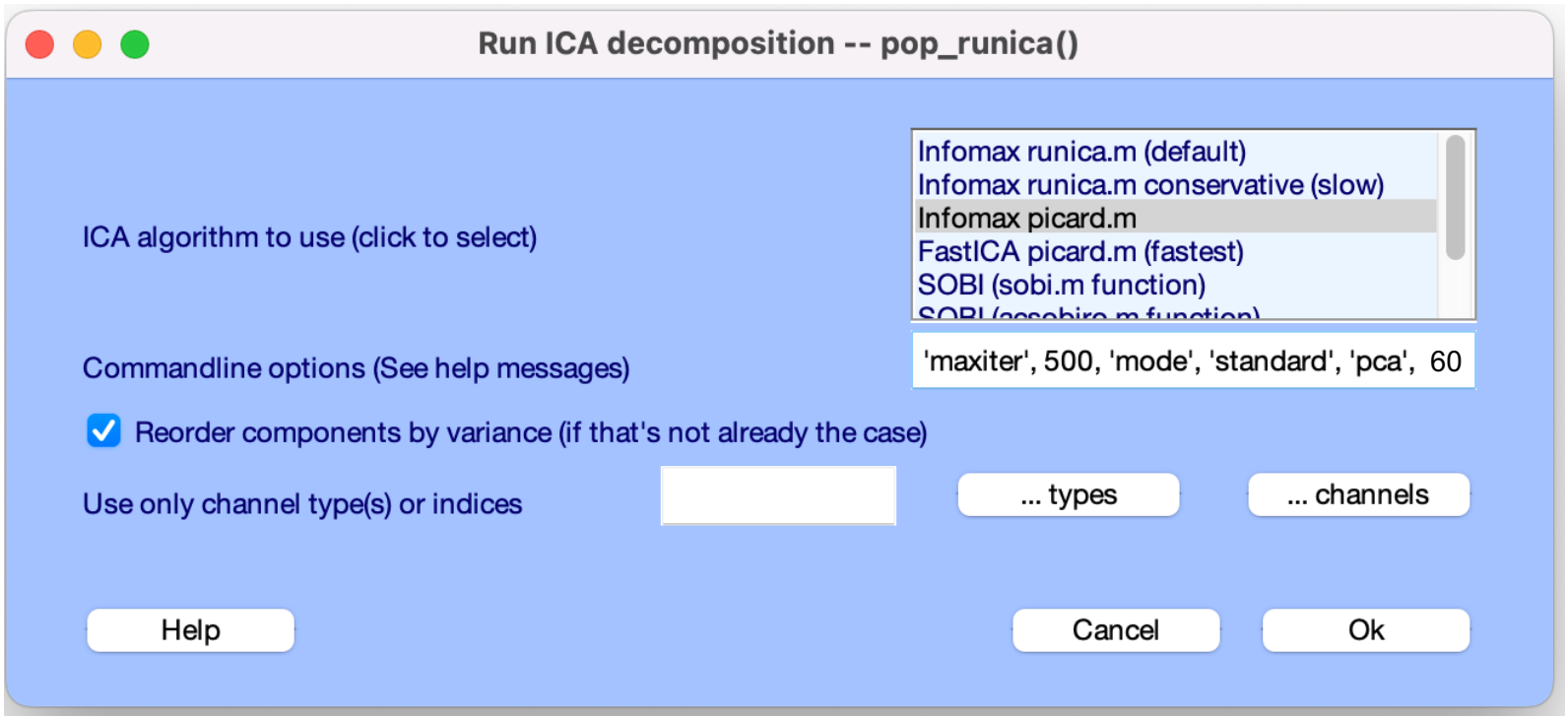




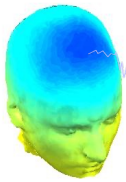
Option	Default	Comments
'extended'	0	1 is recommended to find sub-gaussians
'stop'	1e-7	final weight change → stop
'lrate'	determined from data	too small → too long… too large → wts blow up
'maxsteps'	512	more channels → more steps
'pca'	0 or EEG.nbchan	Decompose only a principal data subspace



Faster ICA option



ICA options

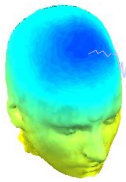


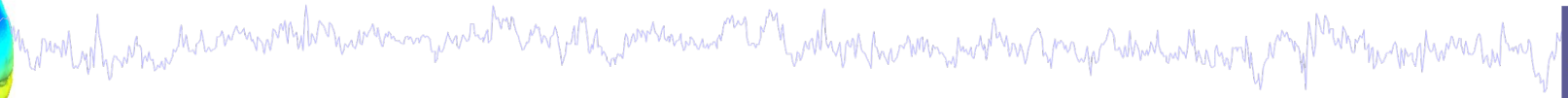
```
EEG = pop_runica( EEG , 'runica', 'extended',1, 'pca', EEG.nbchan-1);
```

Faster option:

```
EEG = pop_runica( EEG , 'picard', 'maxiter', 500, 'pca', EEG.nbchan-1);
```

Runica progress…





MATLAB R2021b - academic use

HOME PLOTS APPS

Search Documentation Sign In

New Script | New Live Script | New | Open | Find Files | Compare — FILE
Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace — VARIABLE
Favorites | Analyze Code | Run and Time | Clear Commands — CODE
Layout | Preferences | Set Path | Add-Ons — ENVIRONMENT
Help | Community | Request Support | Learn MATLAB — RESOURCES

/ ▸ Users ▸ johanna ▸ Library ▸ CloudStorage ▸ GoogleDrive-joa.wagn@gmail.com ▸ My Drive ▸ eeglab workshop ▸ PracticalMEEG_2022 ▸ MATLAB ▸

Current Folder

Name | Git
PracticalMEEG_Preprocess_Dat...
PracticalMEEG_Importa Data.m
PracticalMEEG_Import_Data_S...
PracticalMEEG_ERP_Analysis_S...
trash
Figures

Details

Select a file to view details

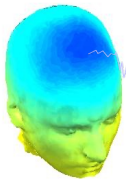
Command Window

```
>> EEG = pop_runica( EEG , 'runica', 'extended',1, 'pca', EEG.nbchan-1);

Saving current ICA decomposition in "EEG.etc.oldicaweights" (etc.).
              Decomposition saved as entry 2.
Warning: you have used PCA to reduce dimensionality so ICA
         is not modeling the entire data, only the PCA-reduced data.
Warning: fixing rank computation inconsistency (61 vs 60) most likely because running under Linux 64-bit Matlab
Attempting to convert data matrix to double precision for more accurate ICA results.

Input data size [60,47826] = 60 channels, 47826 frames/nAfter PCA dimension reduction,
  finding 60 ICA components using extended ICA.
Kurtosis will be calculated initially every 1 blocks using 6000 data points.
Decomposing 13 frames per ICA weight ((3600)^2 = 47826 weights, Initial learning rate will be 0.001, block size 54.
Learning rate will be multiplied by 0.98 whenever angledelta >= 60 deg.
More than 32 channels: default stopping weight change 1E-7
Training will end when wchange < 1e-07 or after 512 steps.
Online bias adjustment will be used.
Removing mean of each channel ...
Final training data range: -40.6563 to 130.95
Reducing the data to 60 principal dimensions...
Computing the sphering matrix...
Starting weights are the identity matrix ...
Sphering the data ...
Beginning ICA training ... first training step may be slow ...
step 1 - lrate 0.001000, wchange 29.18269909, angledelta  0.0 deg
step 2 - lrate 0.001000, wchange 0.88232071, angledelta  0.0 deg
step 3 - lrate 0.001000, wchange 0.70743793, angledelta 80.8 deg
step 4 - lrate 0.000980, wchange 0.60157838, angledelta 93.3 deg
step 5 - lrate 0.000960, wchange 0.55077526, angledelta 99.2 deg
```

Press button to interrupt runica()
Interrupt

Busy

Results of ICA Decomposition in EEG struct

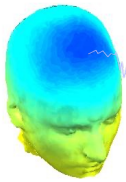


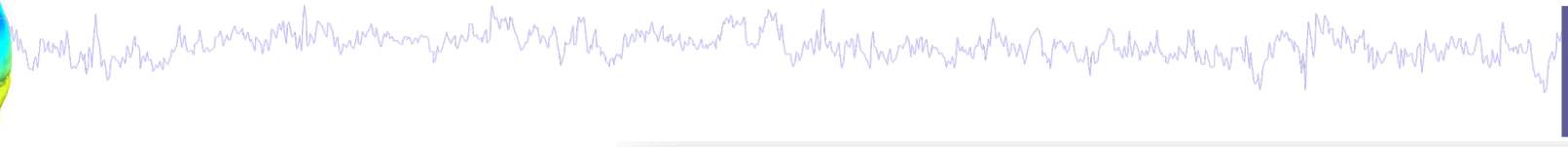
Command Window

```
EEG =

  struct with fields:

            setname: 'sub-01_ses-meg_task-facerecognit
           filename: 'wh_S01_run_01_preprocessing_data_
           filepath: '/Users/johanna/Library/CloudStora
            subject: 'sub-01'
              group: ''
          condition: ''
            session: []
           comments: 'Original file: /Volumes/GoogleDr
             nbchan: 61
             trials: 1
               pnts: 47826
              srate: 100
               xmin: 0
               xmax: 478.2503
              times: [0 10.0000 20.0000 30.0000 40.0000
               data: [61×47826 single]
             icaact: [60×47826 single]
            icawinv: [61×60 double]
          icasphere: [61×61 double]
         icaweights: [60×61 double]
        icachansind: [1 2 3 4 5 6 7 8 9 10 11 12 13 14
           chanlocs: [61×1 struct]
         urchanlocs: [74×1 struct]
           chaninfo: [1×1 struct]
                ref: 'average'
              event: [1×264 struct]
            urevent: [1×259 struct]
    eventdescription: {''  ''  ''  ''}
              epoch: []
    epochdescription: {}
             reject: [1×1 struct]
              stats: [1×1 struct]
           specdata: []
        specicaact: []
        splinefile: ''
     icasplinefile: ''
             dipfit: []
            history: '↵EEG.etc.eeglabvers = '2021.0'; %
              saved: 'no'
                etc: [1×1 struct]
                run: []
            datfile: ''
```

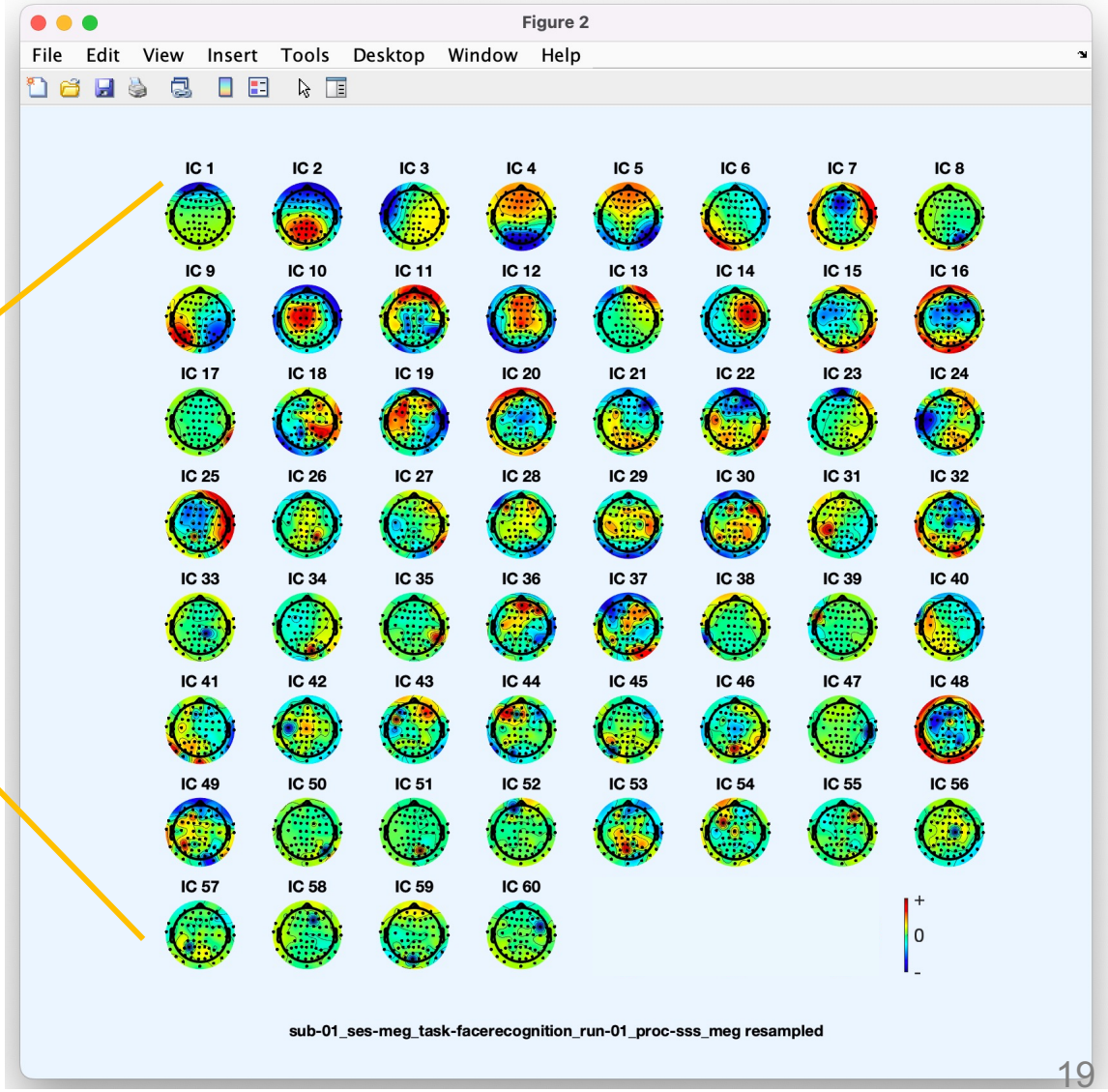
```
icaact: [60×47826 single]
icawinv: [61×60 double]
icasphere: [61×61 double]
icaweights: [60×61 double]
```

Results of ICA Decomposition in EEG struct

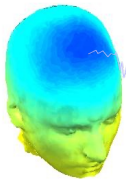





```
    icaact: [60×47826 single]
   icawinv: [61×60 double]
 icasphere: [61×61 double]
icaweights: [60×61 double]
```



English → MATLAB



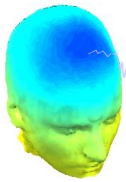
Source activation = unmixing * Channel data

Channel data = mixing (topo) * Source activation

```
EEG.icaact = (EEG.icaweights*EEG.icasphere) * EEG.data
```

```
EEG.data = EEG.icawinv * EEG.icaact
```

Running ICA



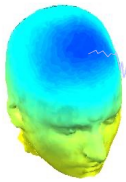
Task 1

Run ICA

Task 2

Evaluating ICA Components

Now what…?



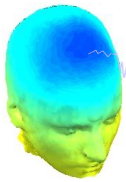
Part 1

Getting an overview of your ICs

Part 2

Classifying/Evaluating ICs

Now what…?



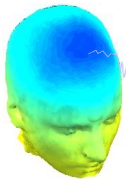
Part 1

Getting an overview of your ICs

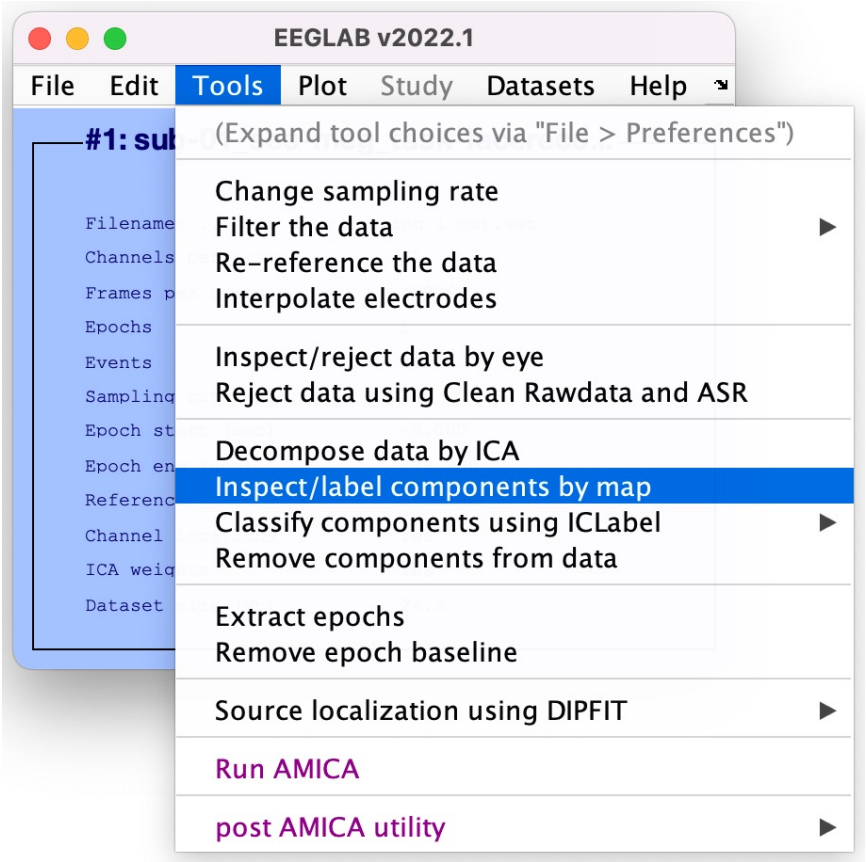
Part 2

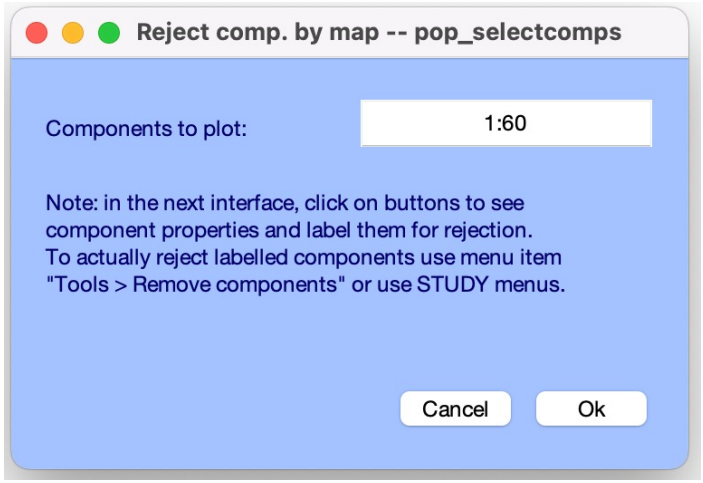
Classifying/Evaluating ICs

A convenient ‘trick’…

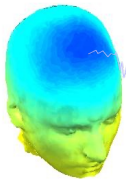


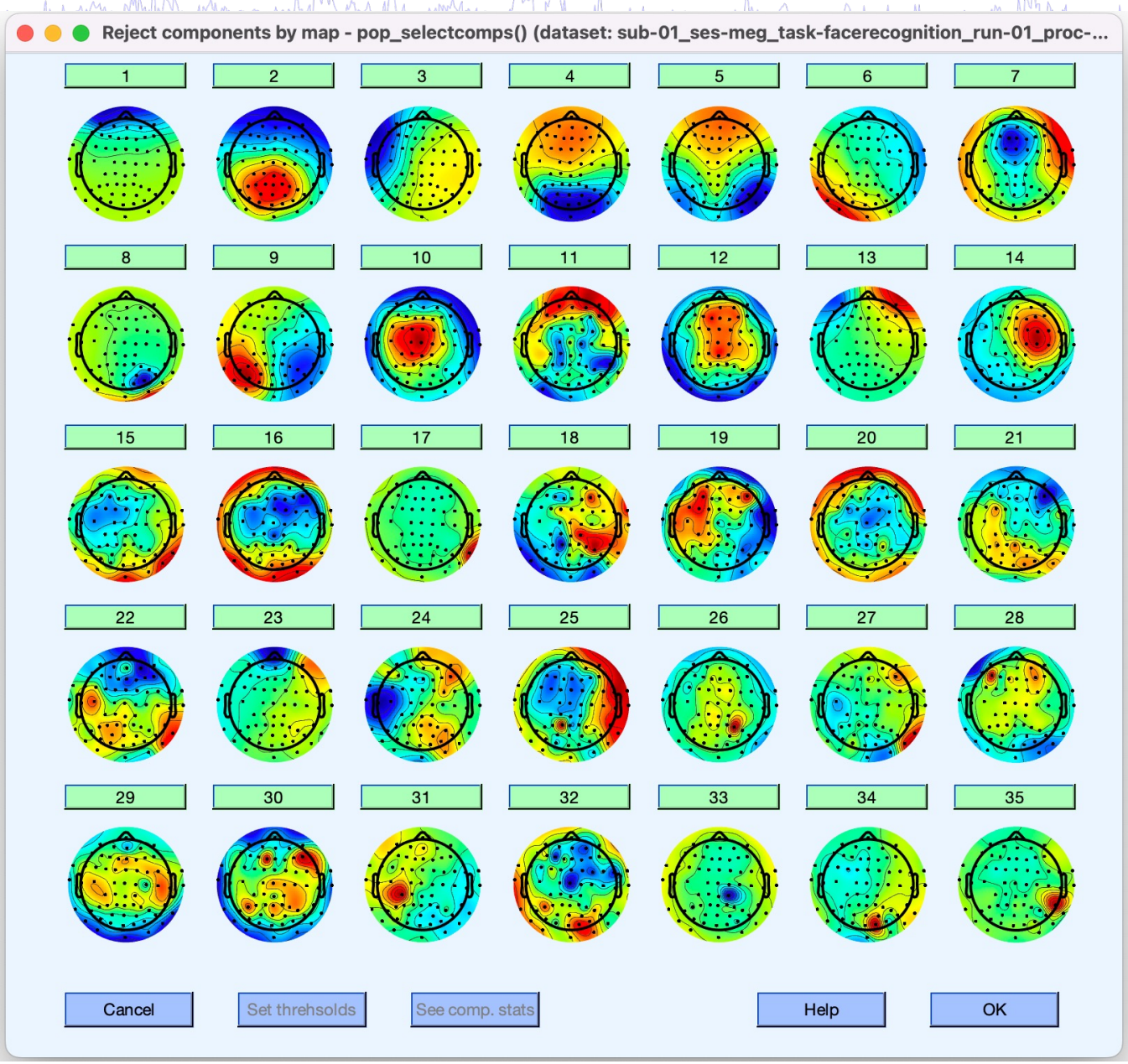
Use ‘Inspect/label components by map’ to survey components



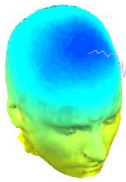


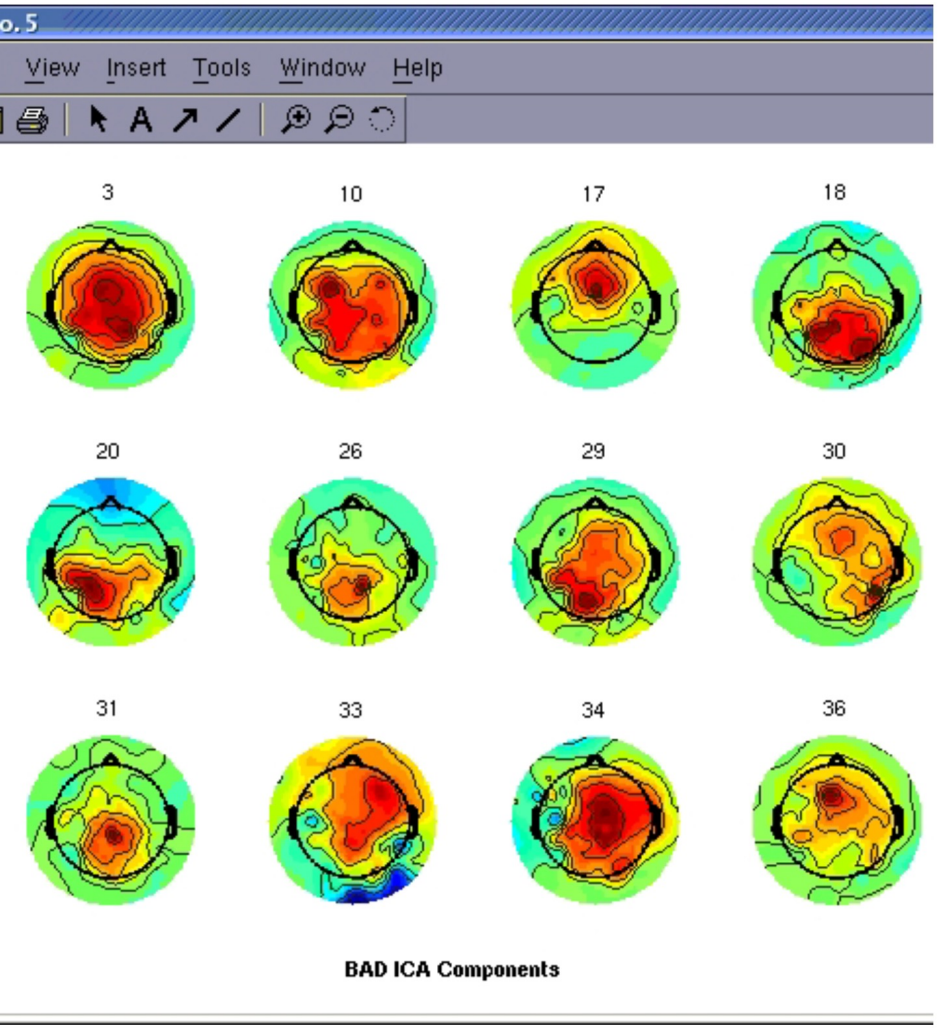
An interactive overview of ICs

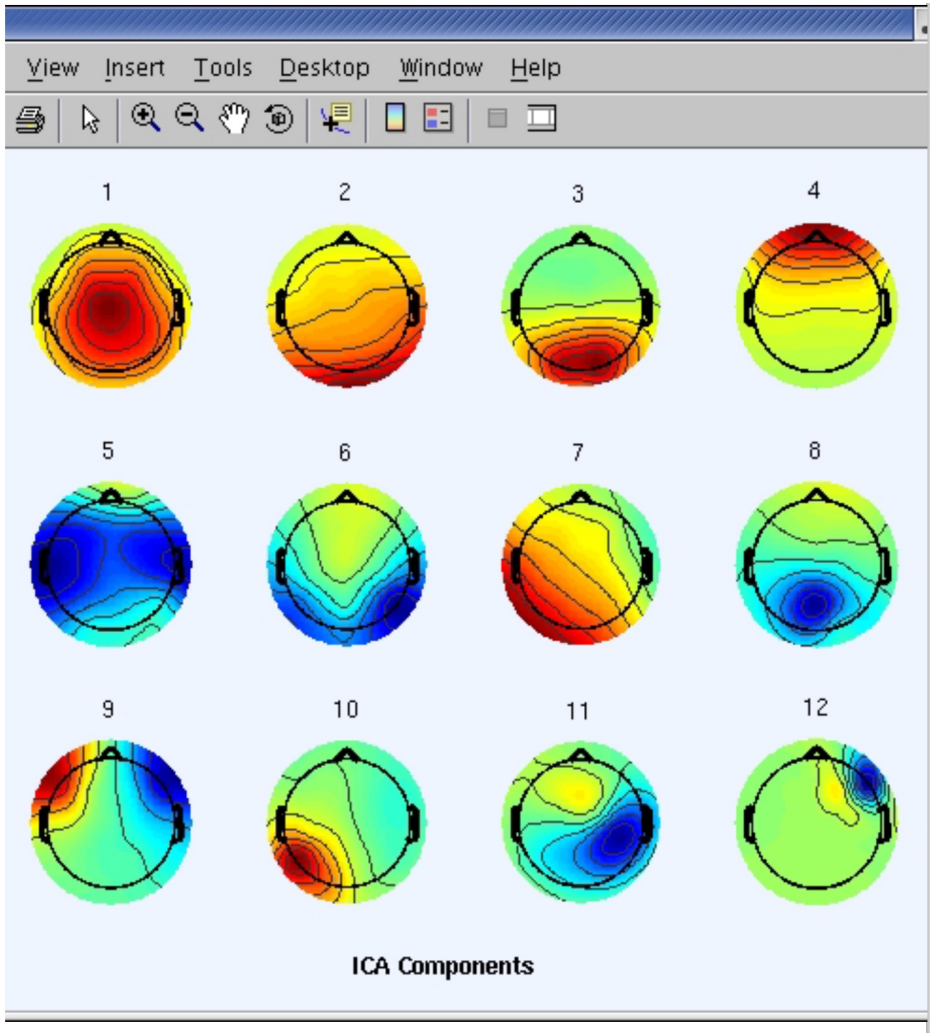




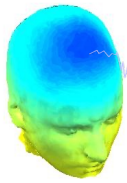
Step 0: Quality of Decomposition

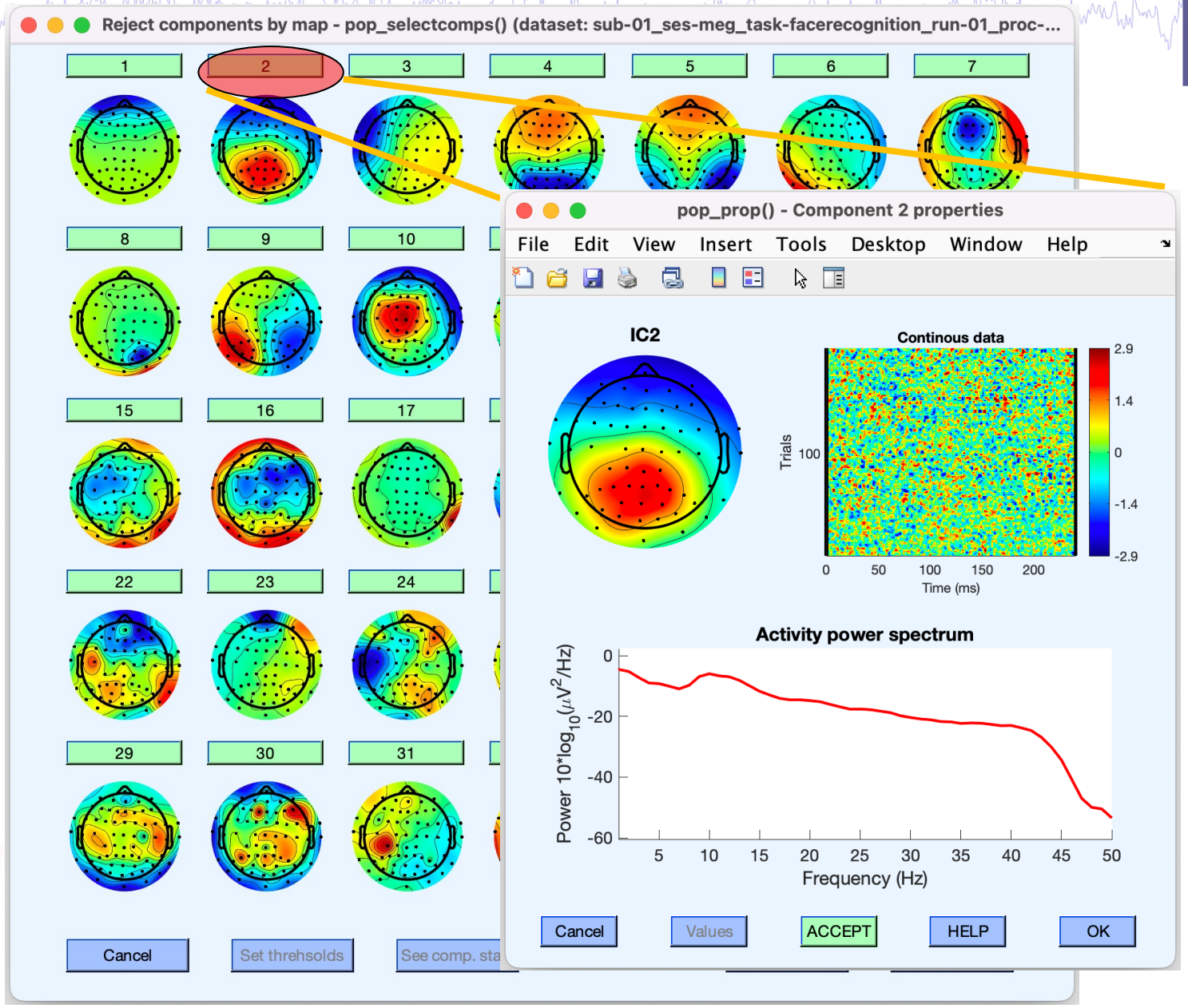




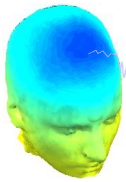


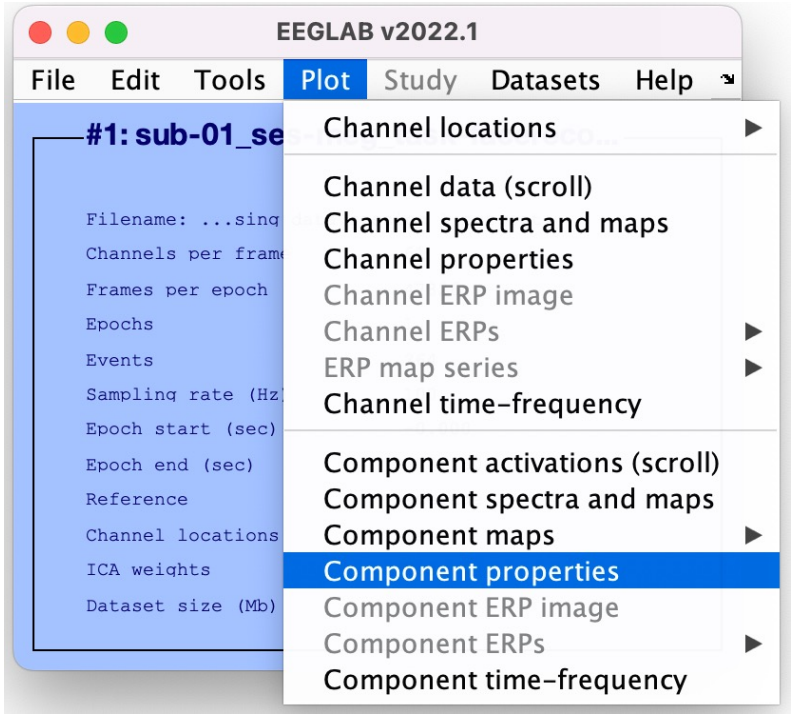
Examining IC Properties

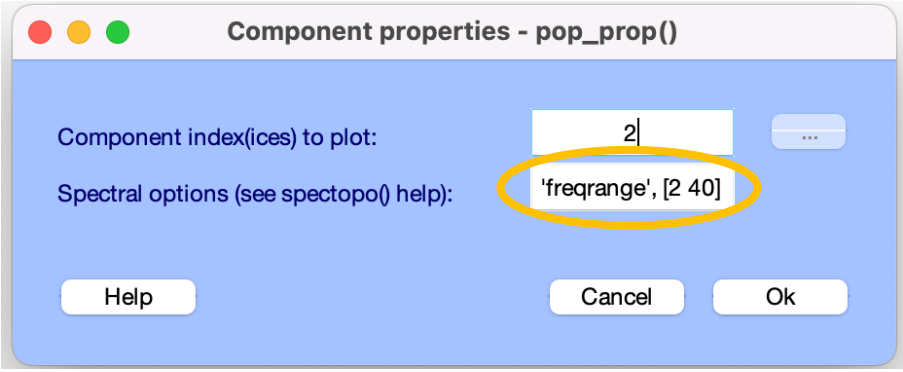




Plot IC Properties

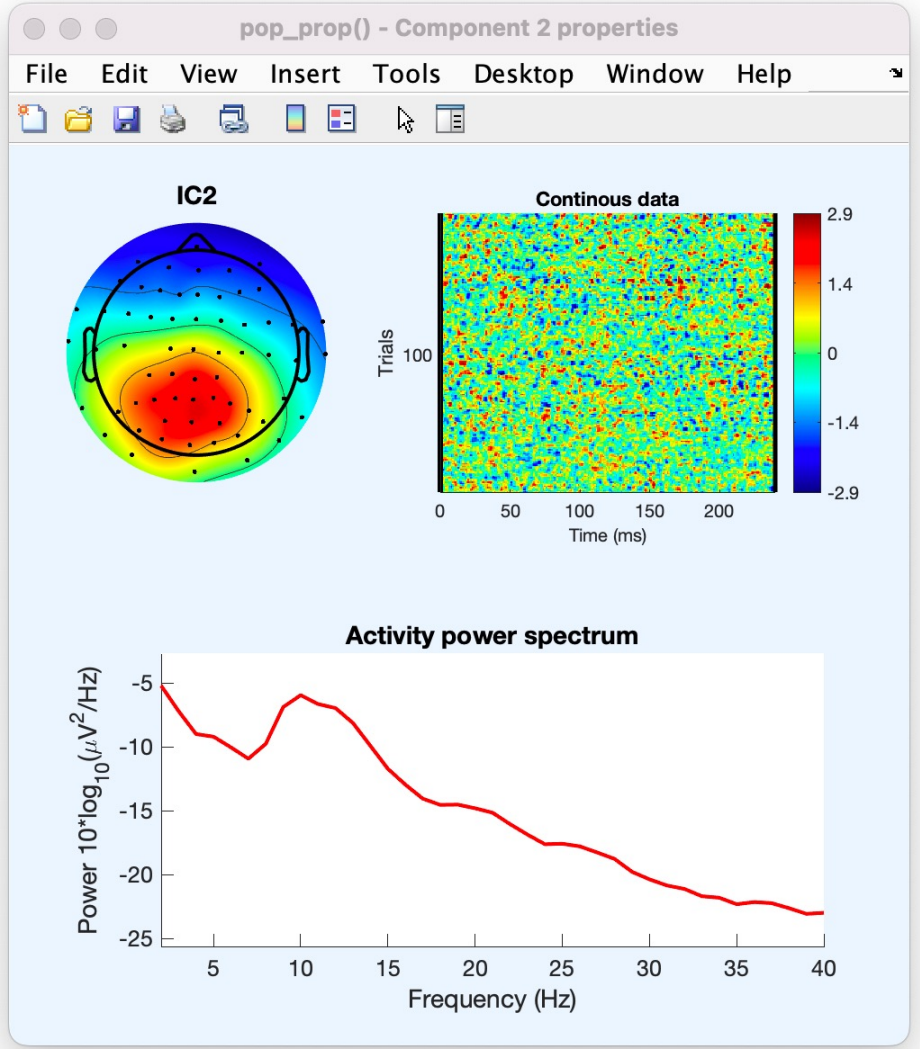






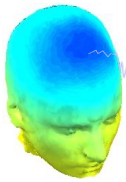
IC Properties

IC Topography
`topoplot()`



ERP Image
&
ERP
`erpimage()`

Power Spectrum
`spectopo()`



Part 1

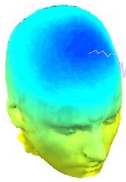
Getting an overview of your ICs

Part 2

Classifying/Evaluating ICs

- Eye Artifacts
- Muscle Artifacts
- Other Artifacts

Evaluating ICs



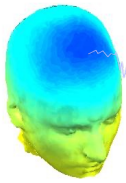
- Over time, most EEGLAB users develop a *heuristic* sense of which ICs might be brain vs. artifact.

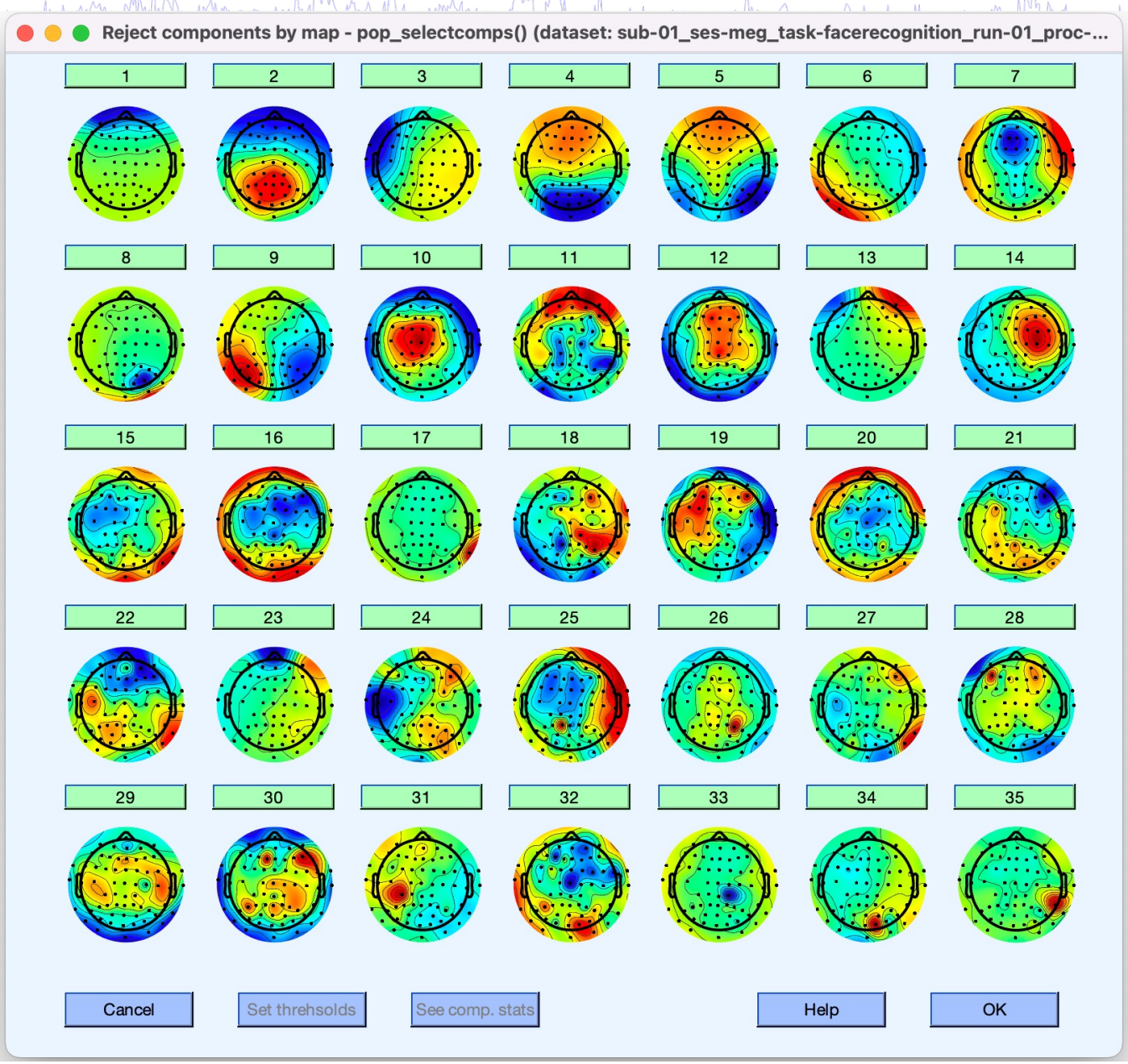
- Heuristics are generally based on:
  - Topography
  - Component Activities (scroll)
  - ERP
  - Power Spectrum

- IC Classification can be used to 'clean' data—study likely brain activity without artifacts

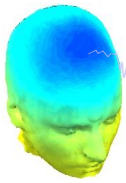
- *There are new efforts to automate this process, but doing it by hand is a good place to start to build intuition* – **IC Label plugin**

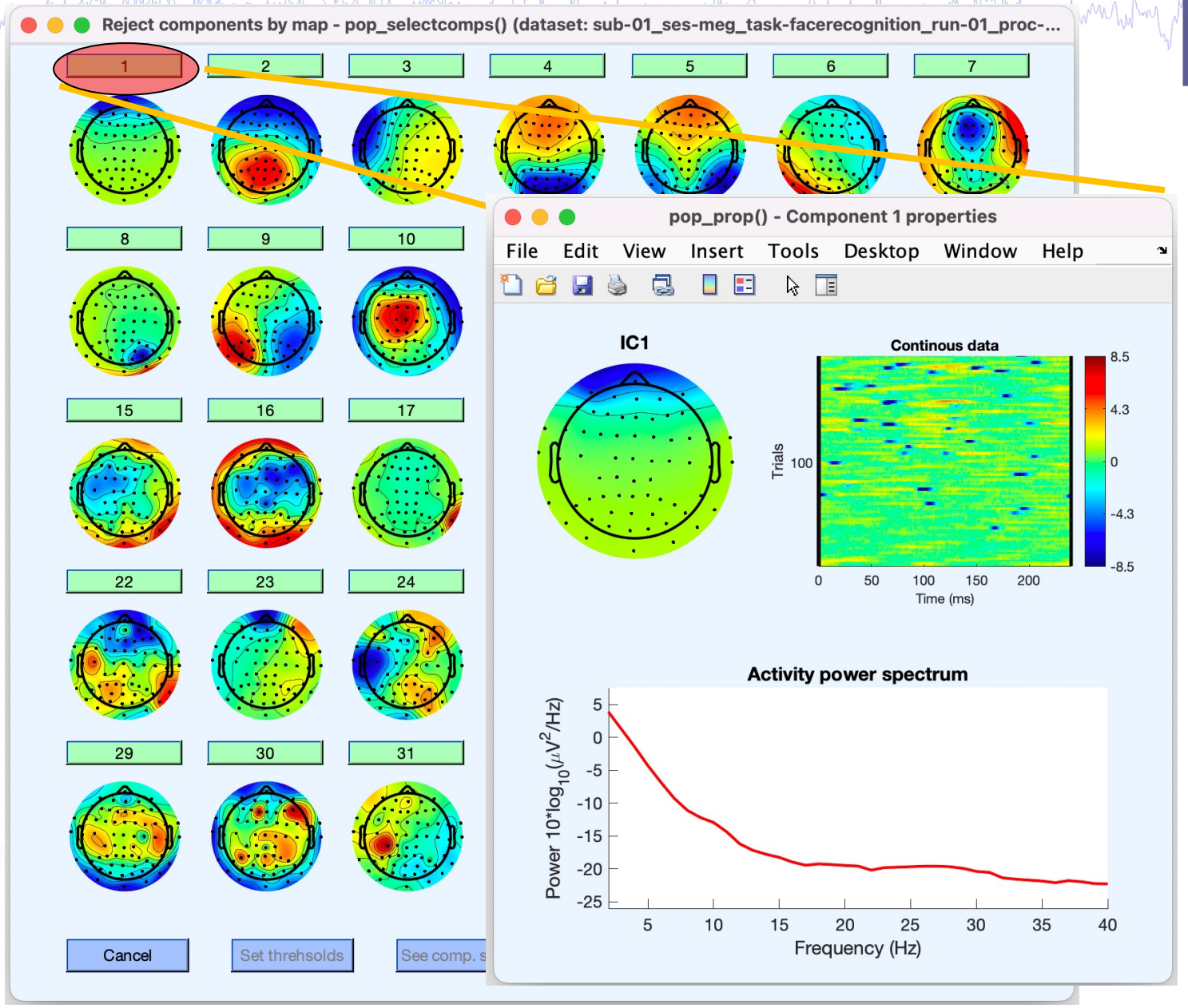
Topography



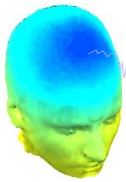

Reject components by map - pop_selectcomps() (dataset: sub-01_ses-meg_task-facerecognition_run-01_proc-...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
Cancel
Set threhsolds
See comp. stats
Help
OK

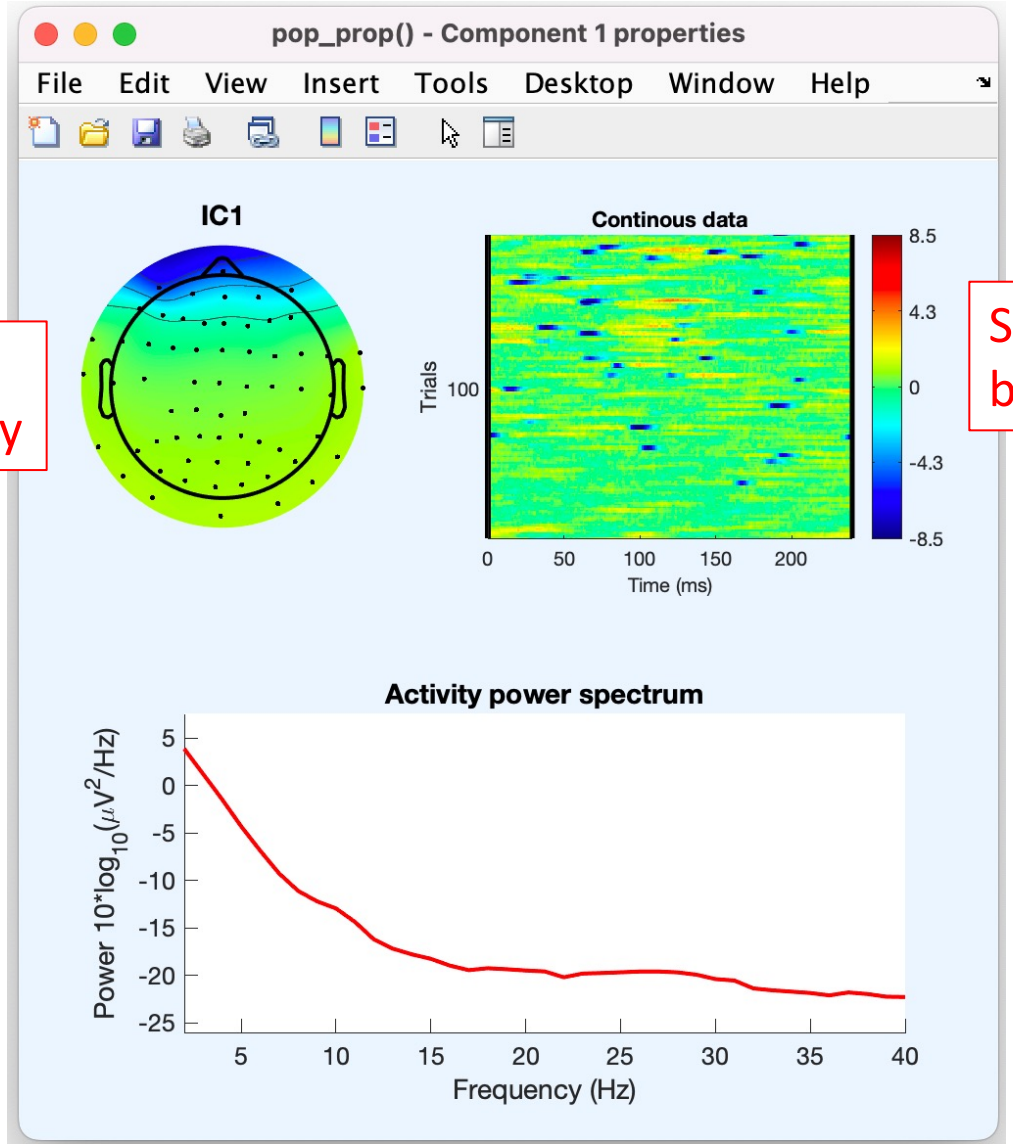
IC1 - eyeblink





IC1 - eyeblink



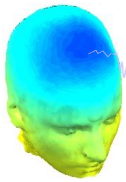


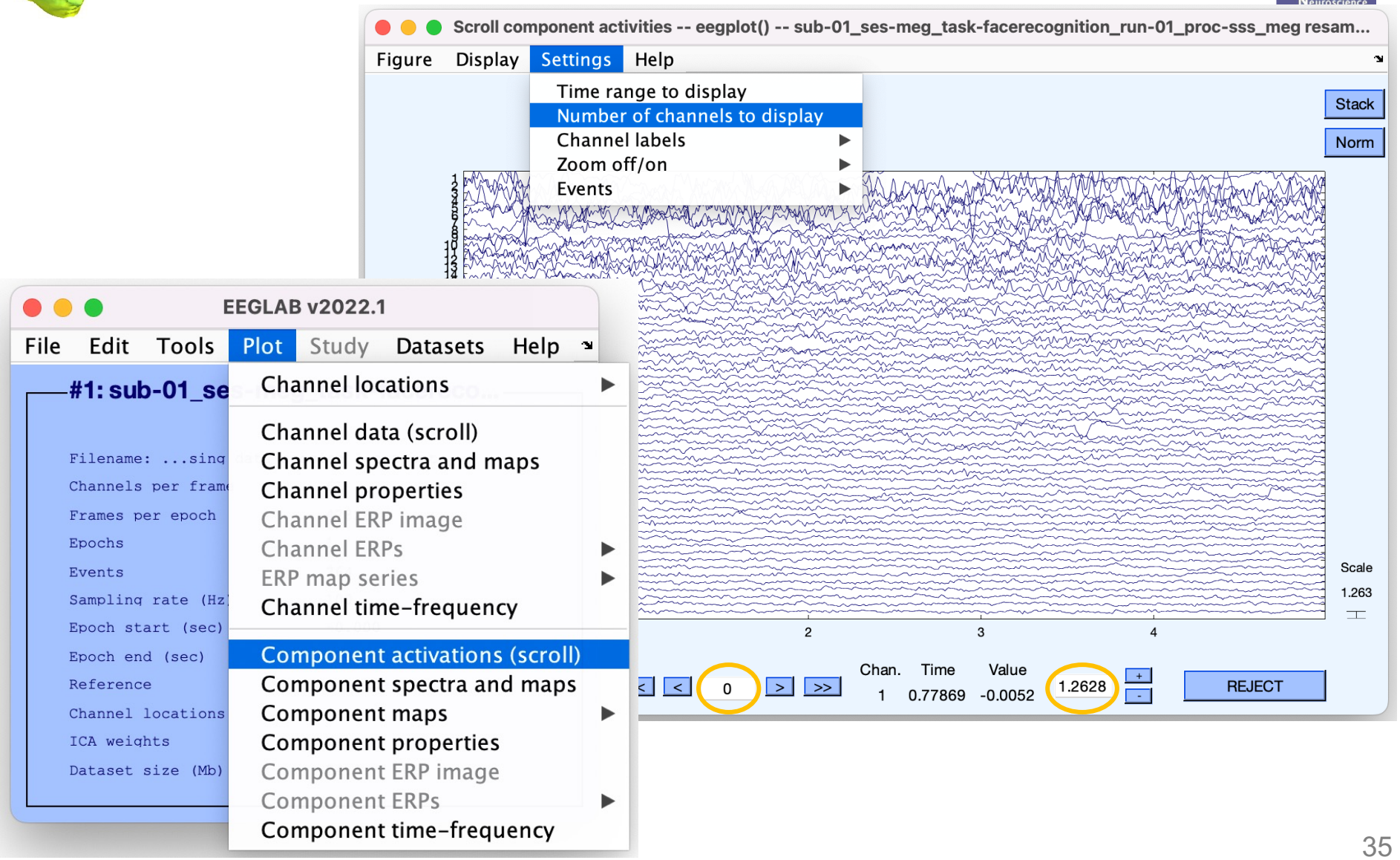
Classic frontal eye-blink topography

Sporadic large biphasic pulses

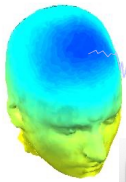
Low frequency activity

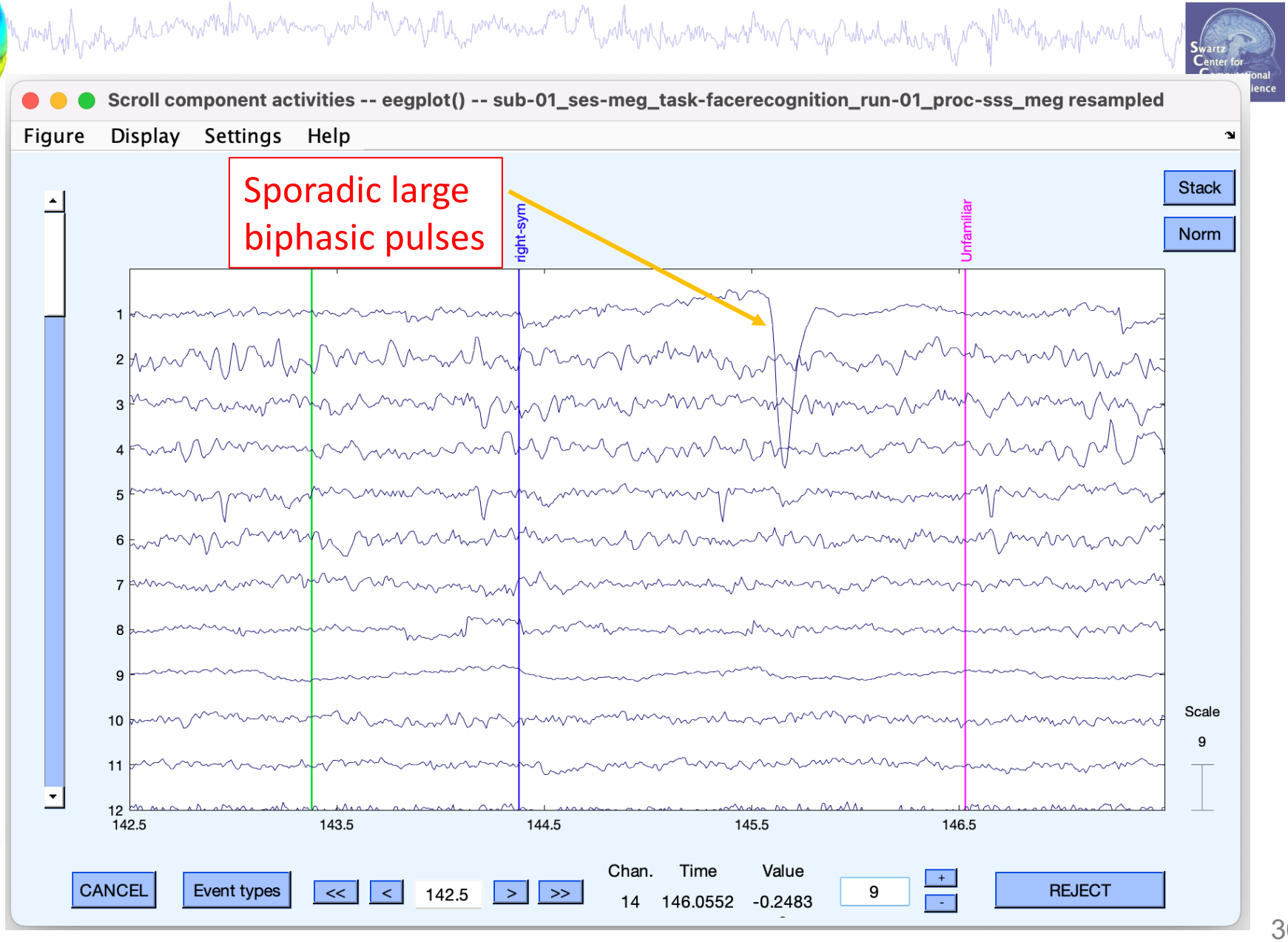
Plot → Component Activations (scroll)



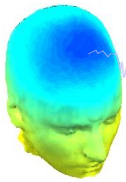


IC 1 Activation – eyeblink



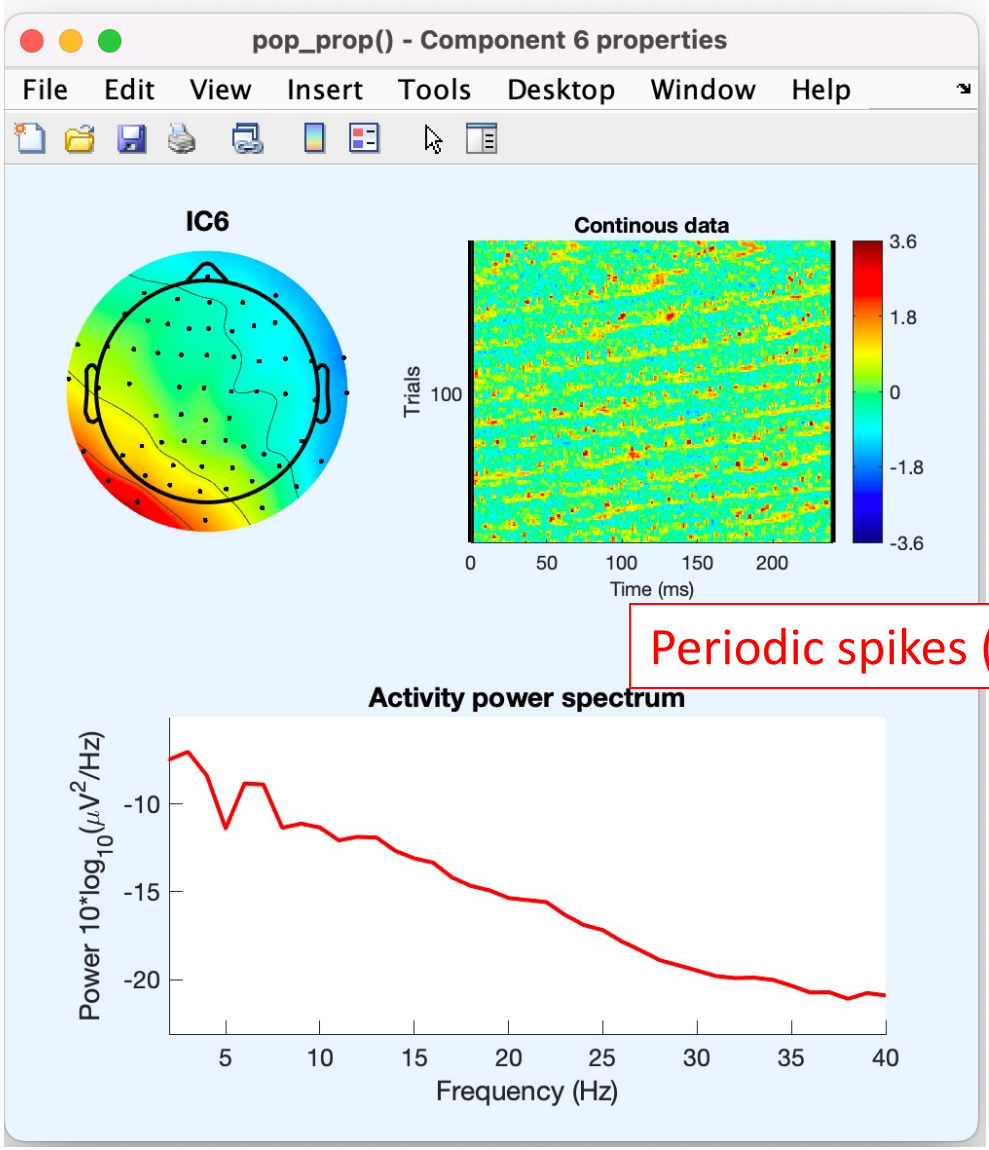


IC 6 - cardiac



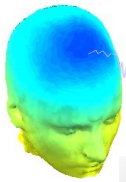
Cardiac-like topographies:
Shallow gradient =
extremely distant source

Unusual, peaky spectrum
(often peaks ~5, 10 Hz)



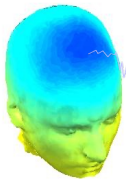
Periodic spikes (~1 / sec)

IC 6 - cardiac

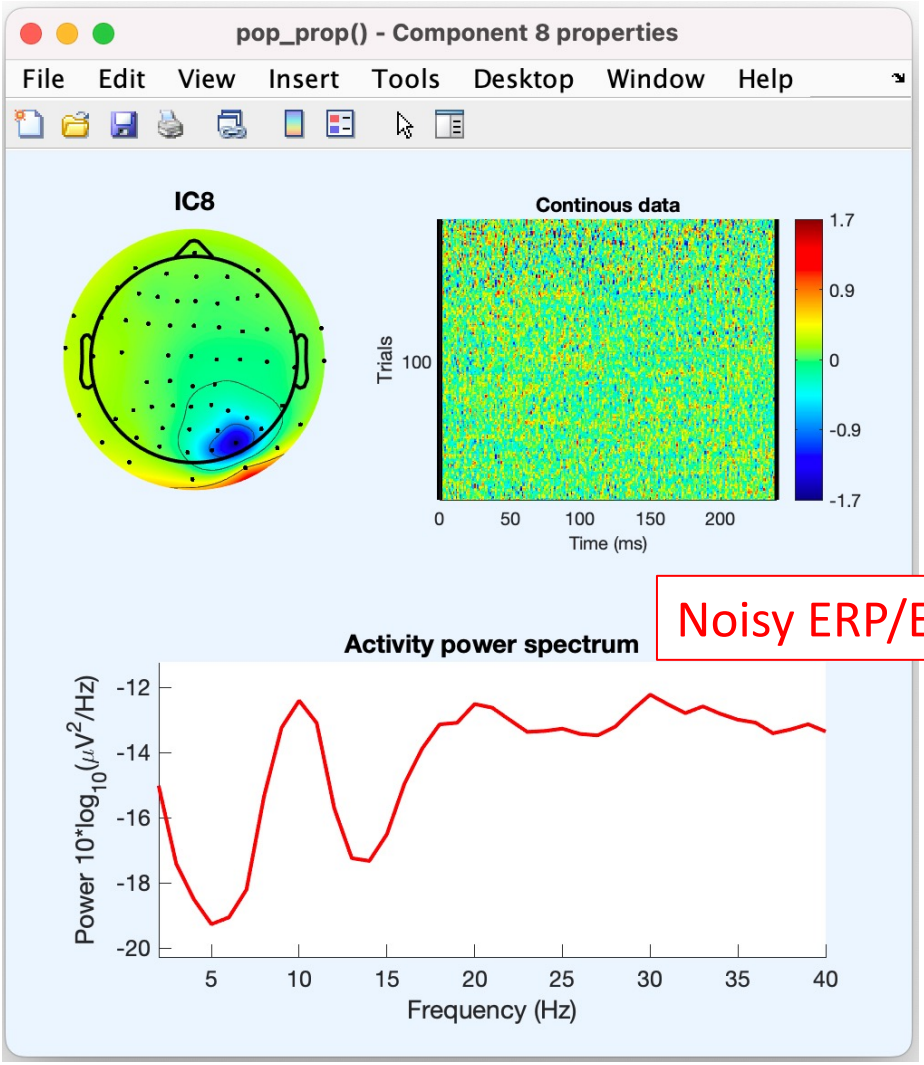



Scroll component activities -- eegplot() -- sub-01_ses-meg_task-facerecognition_run-01_proc-sss_meg resam...
Figure
Display
Settings
Help
Time range to display
Number of channels to display
Channel labels
Zoom off/on
Events
Stack
Norm
Scrambled
left-nons...
Famous
right-sym
Scrambled
right-sym
Periodic spikes (~1 / sec)
Scale
9
CANCEL
Event types
<<
<
108
>
>>
Chan.
Time
Value
1
111.9798
-0.859
9
+
-
REJECT

IC 8 – Muscle



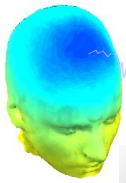
Narrowly spaced dipolar topography (consistent with superficial source)



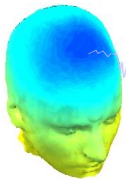
Noisy ERP/ERP Image

High frequencies dominate power spectrum

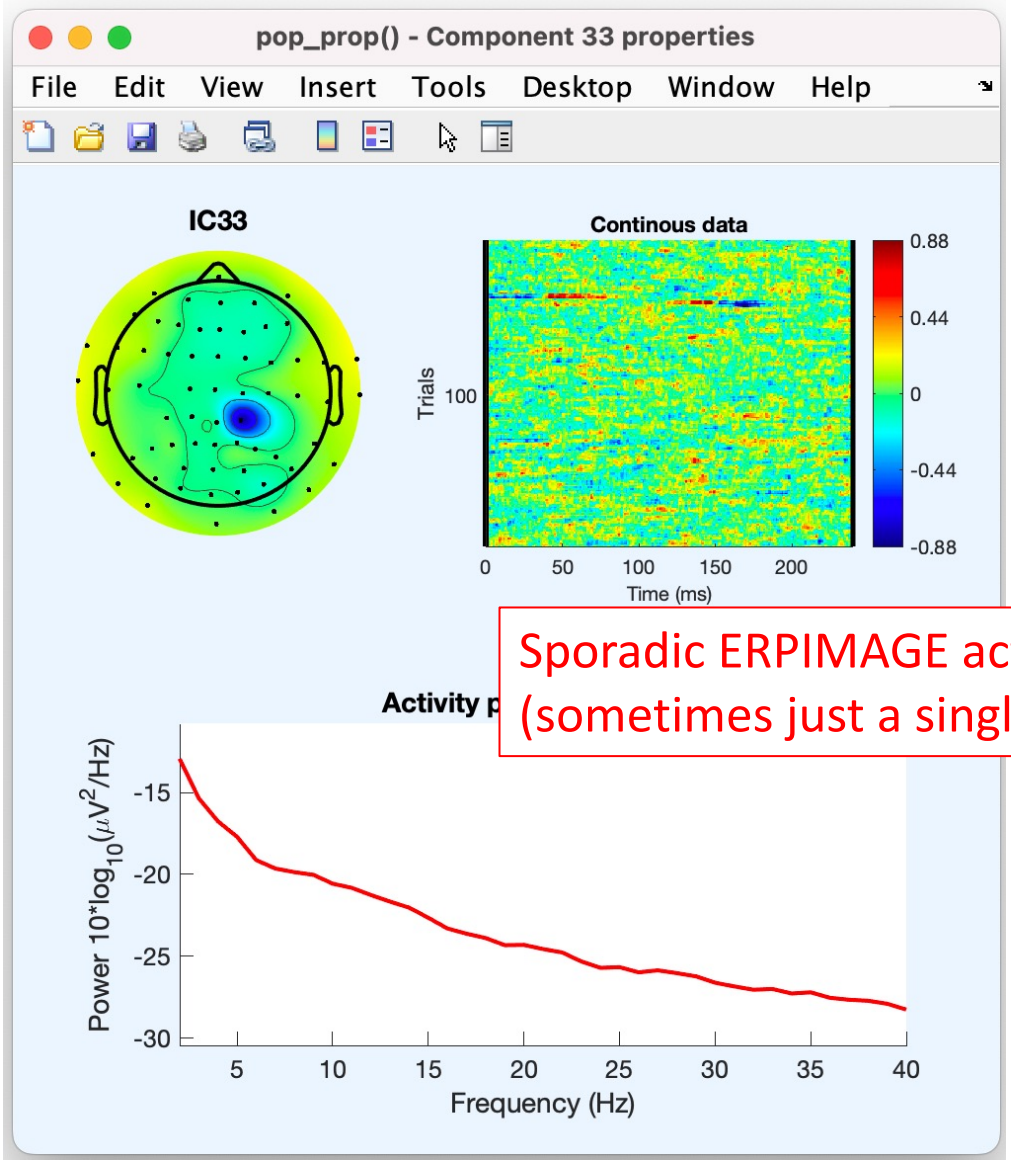
IC 8 Activation – Muscle



IC 33 – Bad channel

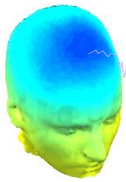


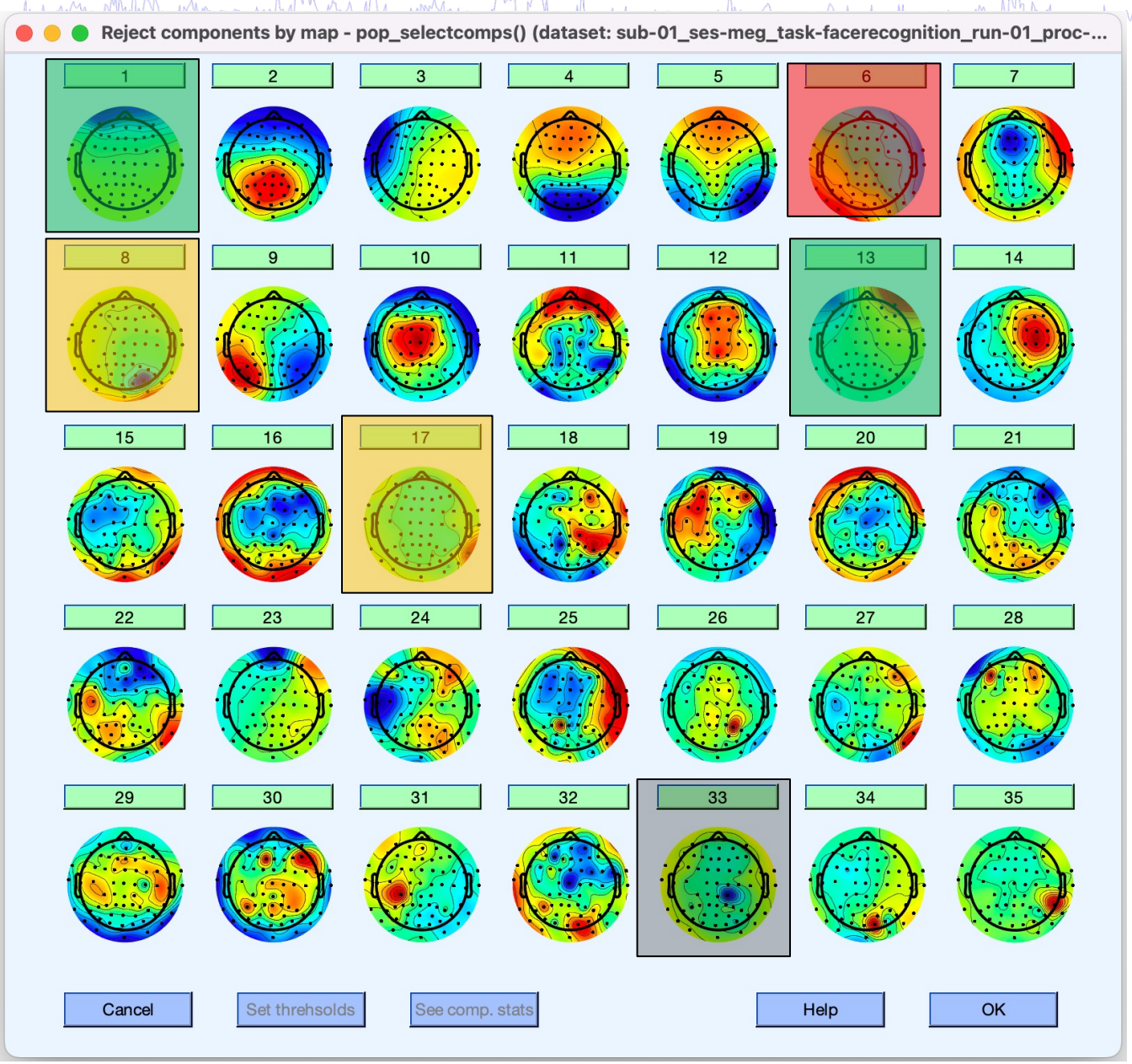
Punctate topography (single channel)



Sporadic ERPIMAGE activity (sometimes just a single large spike)

IC classification so far…





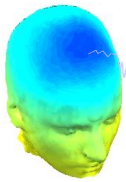
Eye

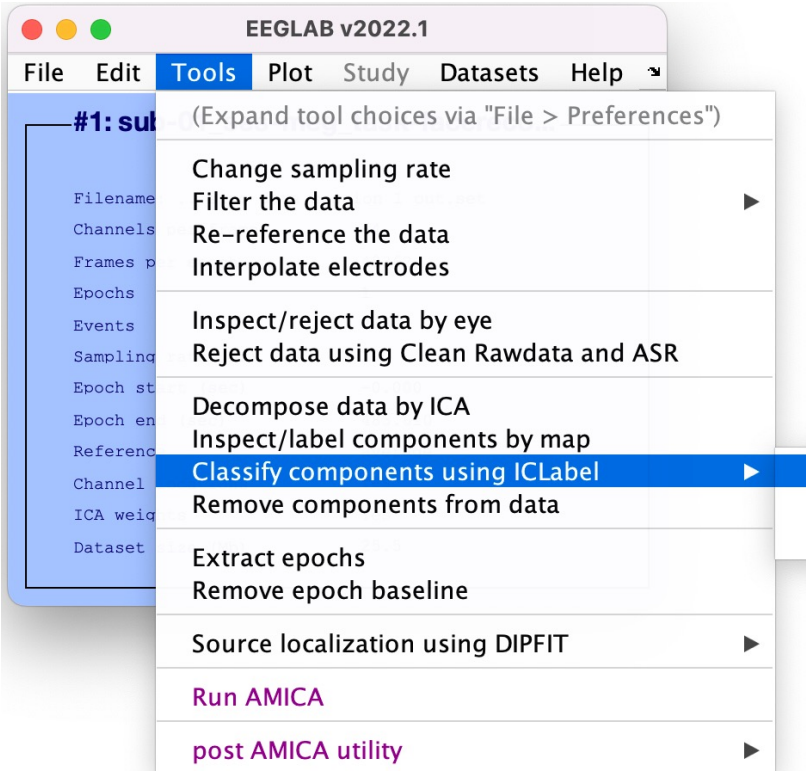
Muscle

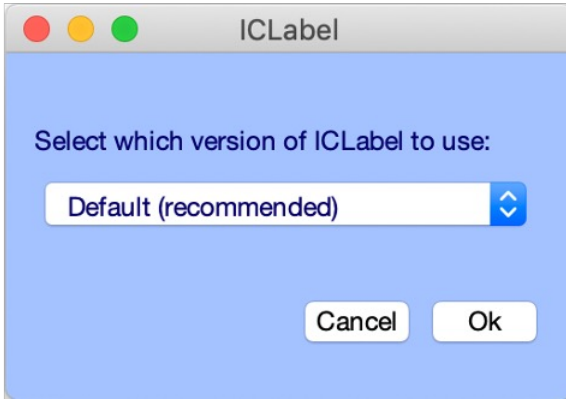
Cardiac

Badchan

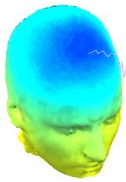
IC Label plugin

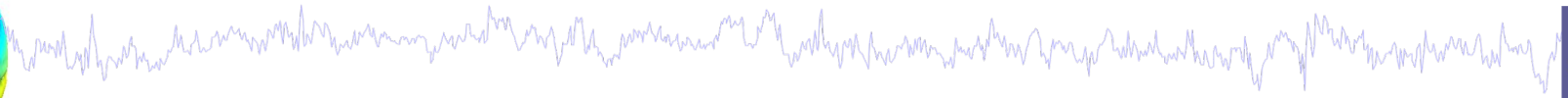






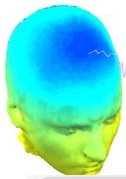
IC Label plugin

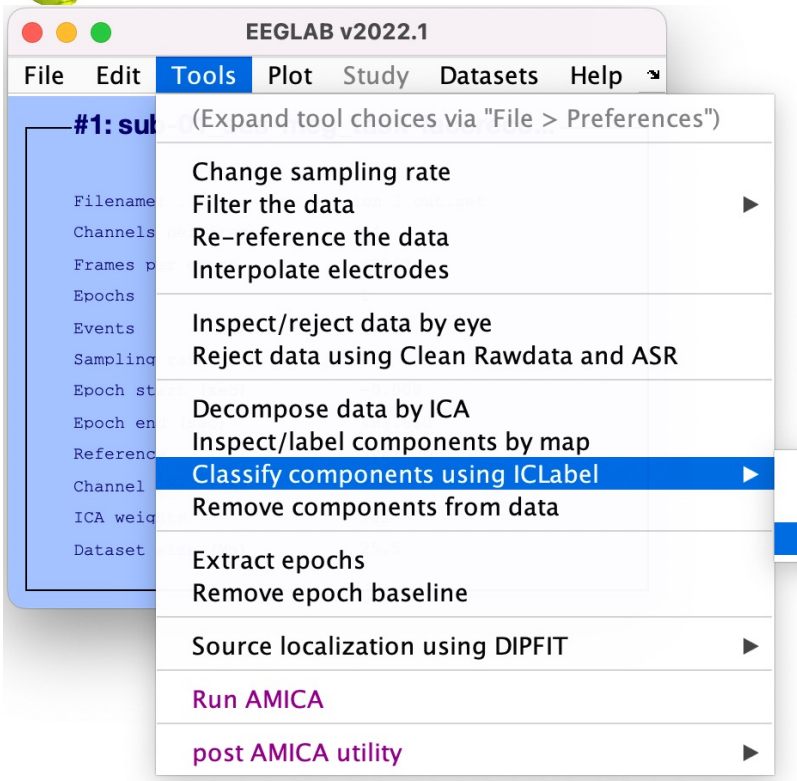


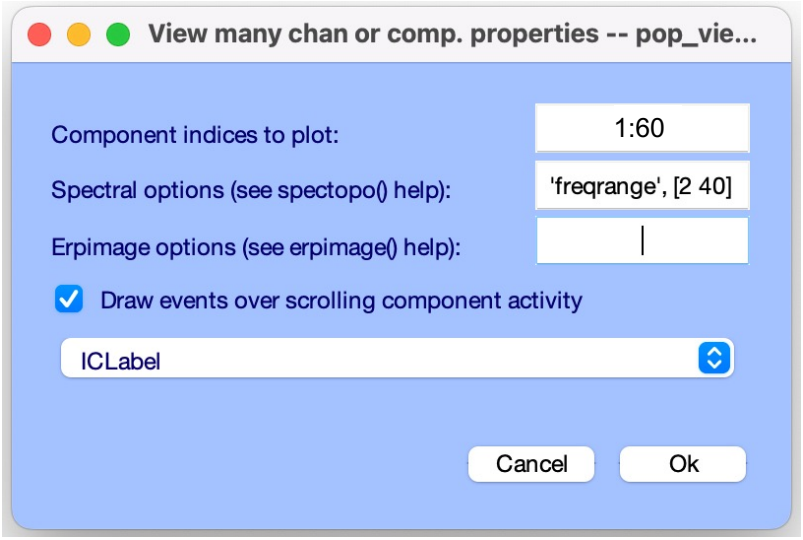


```
>> EEG.etc.ic_classification.ICLabel

ans =

  struct with fields:

          classes: {'Brain'  'Muscle'  'Eye'  'Heart'  'Line Noise'  'Channel Noise'  'Other'}
    classifications: [63×7 single]
          version: 'default'


    >> EEG.etc.ic_classification.ICLabel.classifications

    ans =

      63×7 single matrix

        0.0003    0.0009    0.9871    0.0006    0.0000    0.0011    0.0100
        0.9992    0.0001    0.0000    0.0001    0.0002    0.0000    0.0004
        0.9989    0.0000    0.0001    0.0001    0.0001    0.0000    0.0008
        0.9998    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001
        0.9963    0.0001    0.0000    0.0001    0.0009    0.0000    0.0026
        0.2034    0.0008    0.0004    0.7153    0.0024    0.0013    0.0763
        0.8176    0.0021    0.0002    0.0098    0.0213    0.0003    0.1487
        0.3447    0.5817    0.0021    0.0046    0.0003    0.0174    0.0492
        0.9953    0.0002    0.0000    0.0004    0.0019    0.0001    0.0022
```

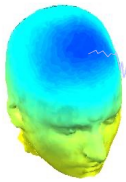
View extended component properties

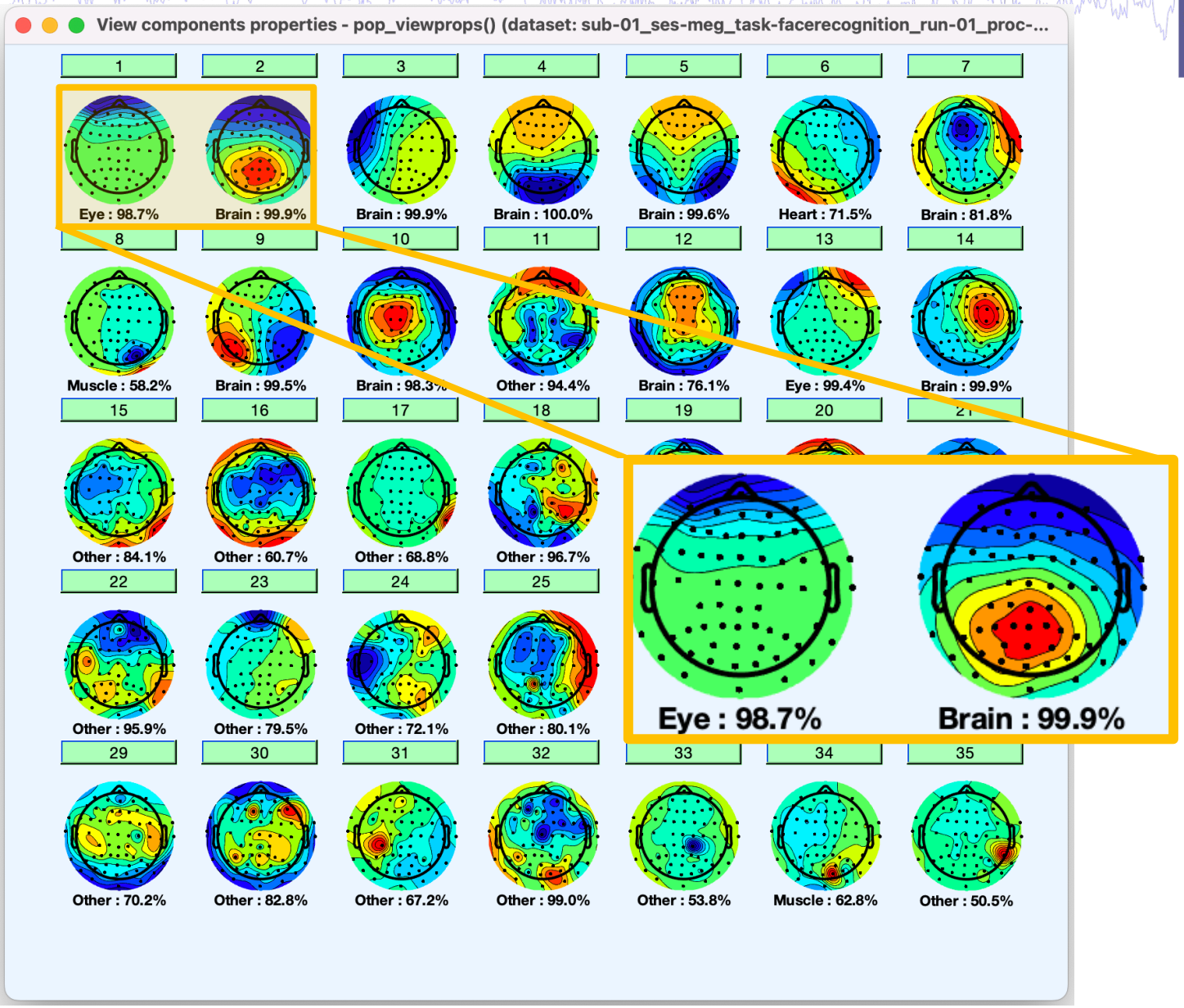




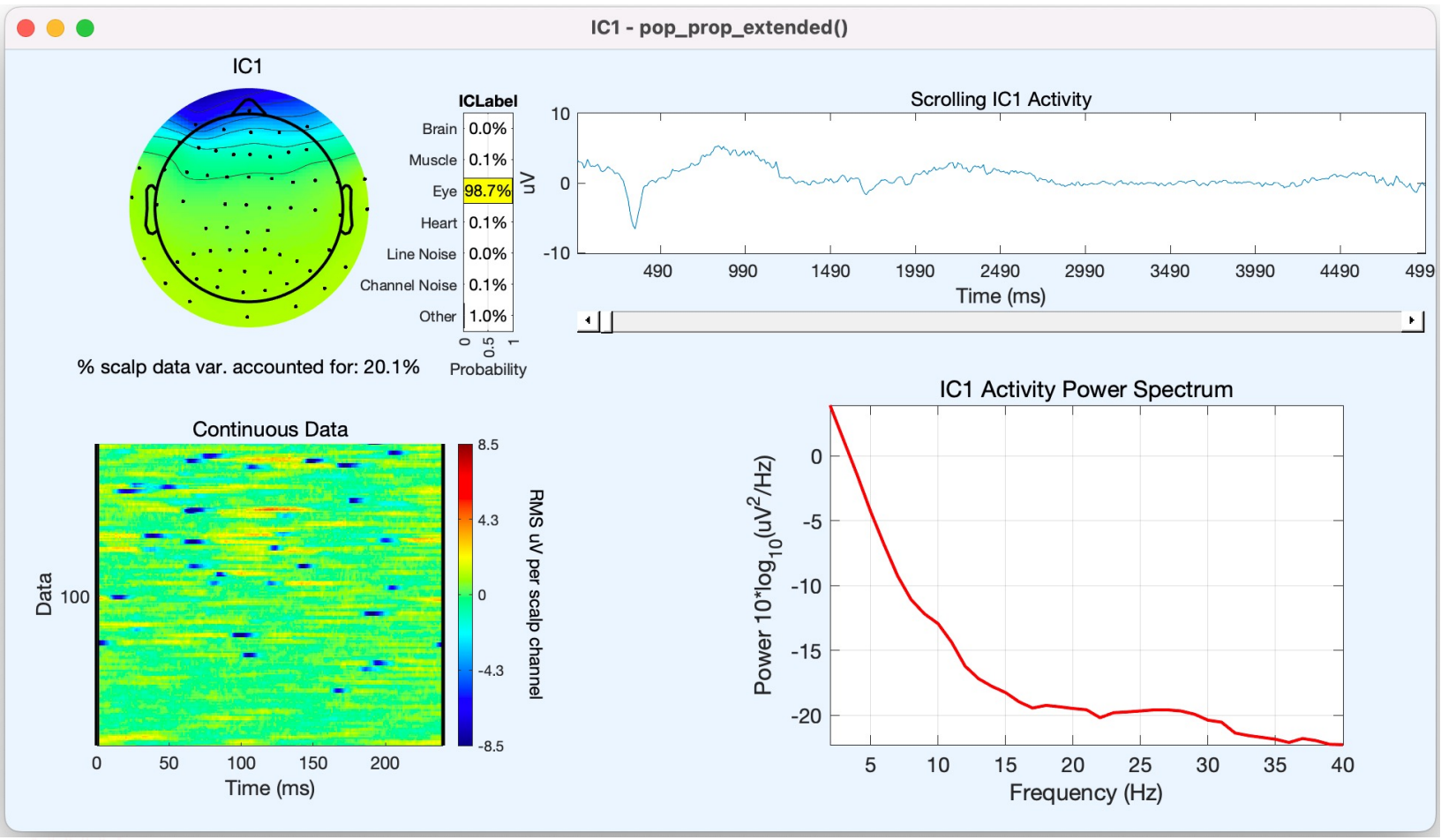


View extended component properties



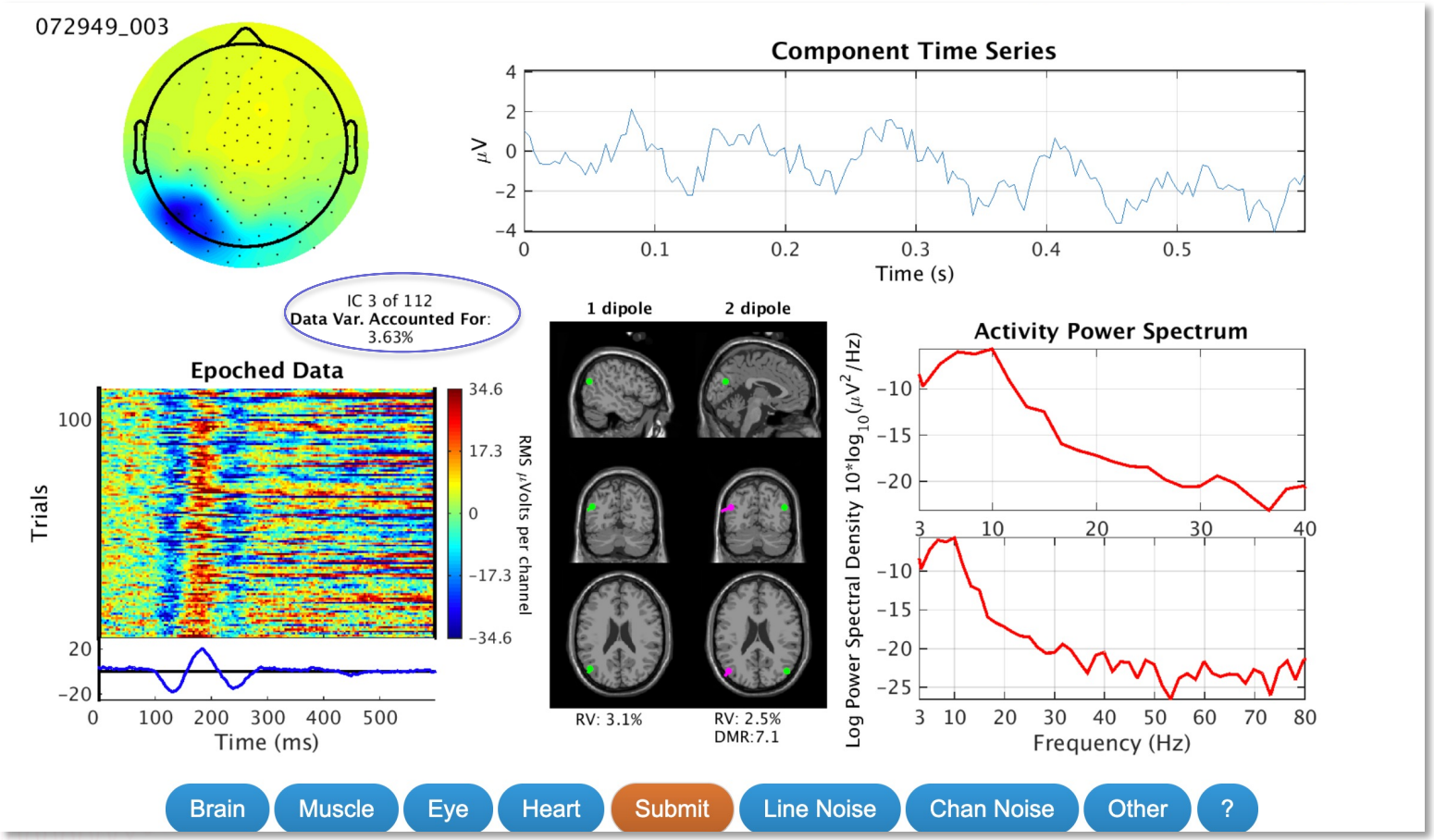


IC Label plugin



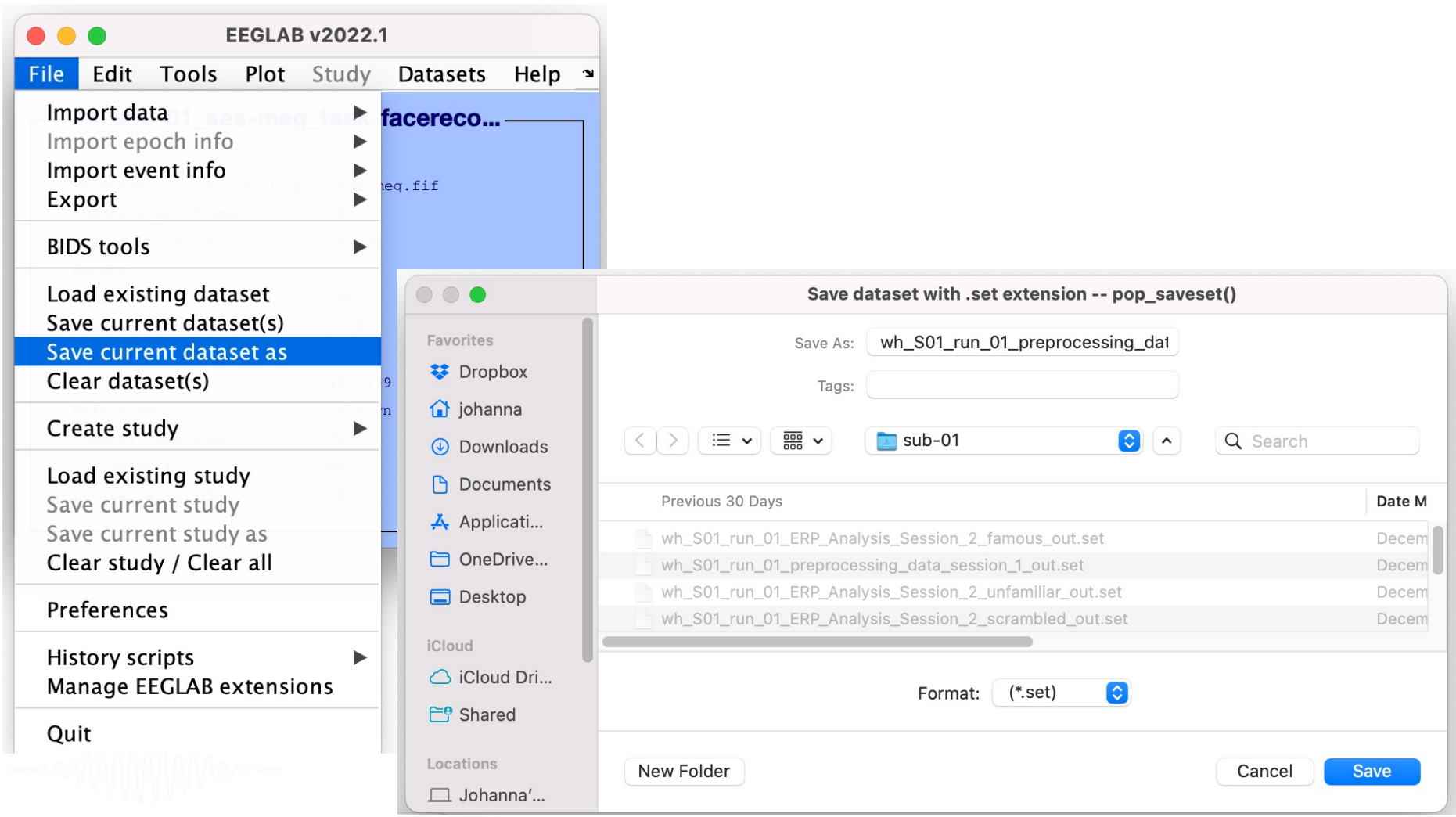
Practice Labeling…

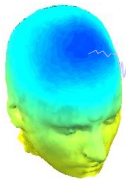
https://labeling.ucsd.edu/tutorial



Luca Pion-Tonachini, 2019

Save dataset





Swartz Center for Computational Neuroscience

Thank You!