HAPPE README File

the Harvard Automated Processing Pipeline for EEG

Laurel Gabard-Durnam
Adriana Mendez Leal
Carol Wilkinson
April Levin
9/30/2017

Table of Contents

What is HAPPE?

- A pipeline for taking unprocessed EEG data and automatedly processing it

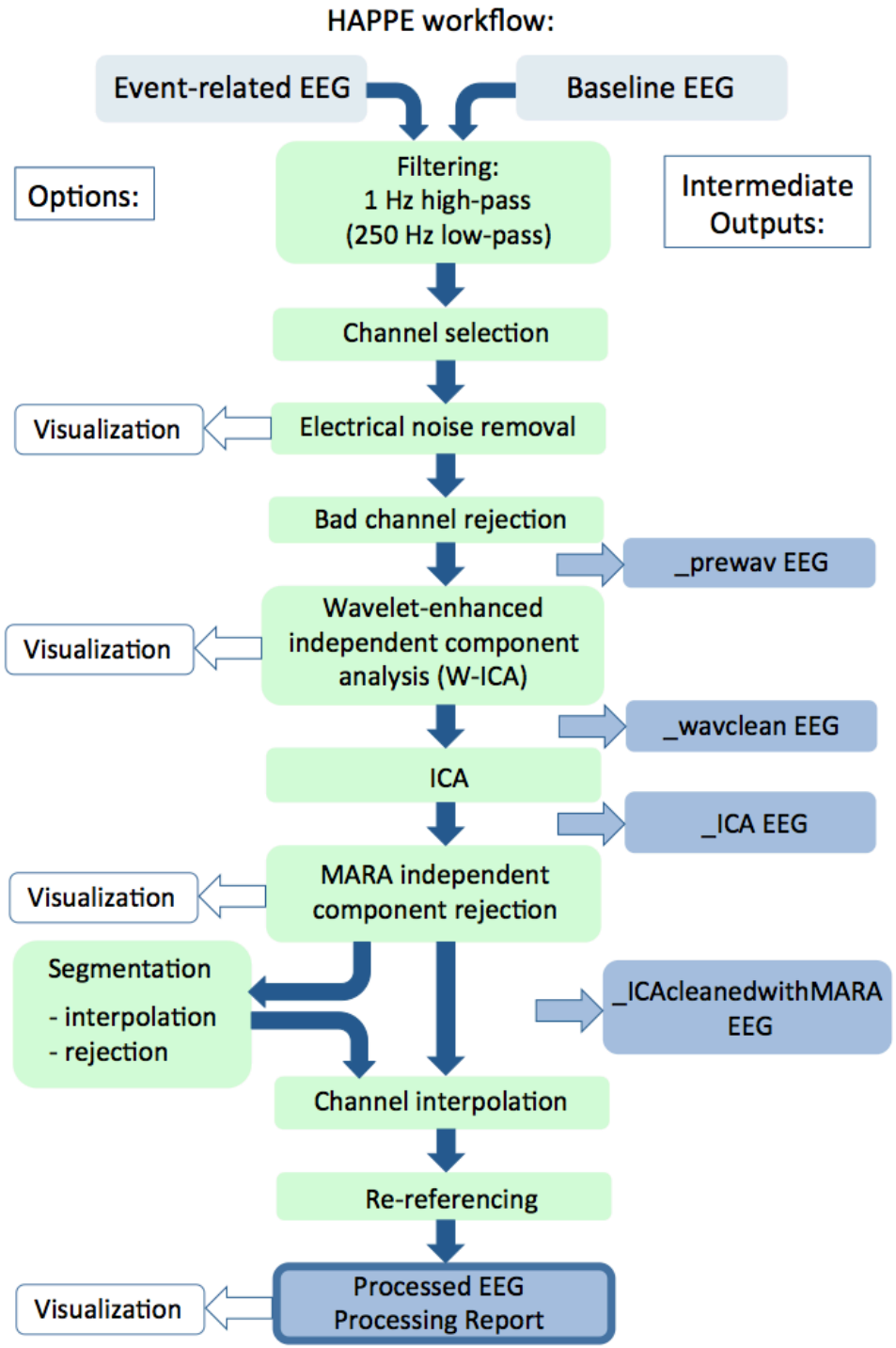
- Translates recent advances in adult EEG processing to developmental data context

- Implements wavelet-enhanced-ICA + ICA approaches to EEG artifact removal

- Agnostic to the program for running analyses afterwards

  Compatible with:

  - BEAPP
  - EEGLAB
  - Matlab
  - Anything that can read a .txt file of EEG data as input


HAPPE workflow:
Event-related EEG
Baseline EEG
Options:
Filtering:
1 Hz high-pass
(250 Hz low-pass)
Intermediate Outputs:
Channel selection
Visualization
Electrical noise removal
Bad channel rejection
_prewav EEG
Visualization
Wavelet-enhanced independent component analysis (W-ICA)
_wavclean EEG
ICA
_ICA EEG
Visualization
MARA independent component rejection
Segmentation
- interpolation
- rejection
_ICAcleanedwithMARA EEG
Channel interpolation
Re-referencing
Visualization
Processed EEG Processing Report

What does HAPPE require?

1. Matlab version 2014 or newer (NOTE: this version is not compatible with NetStation version 4.5, so you cannot run HAPPE on the same computer as NetStation 4.5.)

2. The matlab toolboxes (all should be standard for academic licenses):
    - Signal processing toolbox
    - Optimization toolbox
    - Statistics toolbox

3. Some free programs (included with the HAPPE download):
    - EEGLAB
    - Cleanline EEGLAB plugin
    - MARA EEGLAB plugin

4. The HAPPE scripts

HAPPE runs happily on both Macs and PCs

How to get HAPPE

I. Get EEGLAB running

EEGLAB comes with the download of the HAPPE package. If for some reason you need to re-download EEGLAB, here are the details:

- Download EEGLAB version 14.0.0b from here: ftp://sccn.ucsd.edu/pub/daily/
- It is the February 10, 2017 release
- Theoretically, the pipeline is compatible with newer releases unless you let me know otherwise ☺

1. Move the EEGLAB folder into your Matlab folder

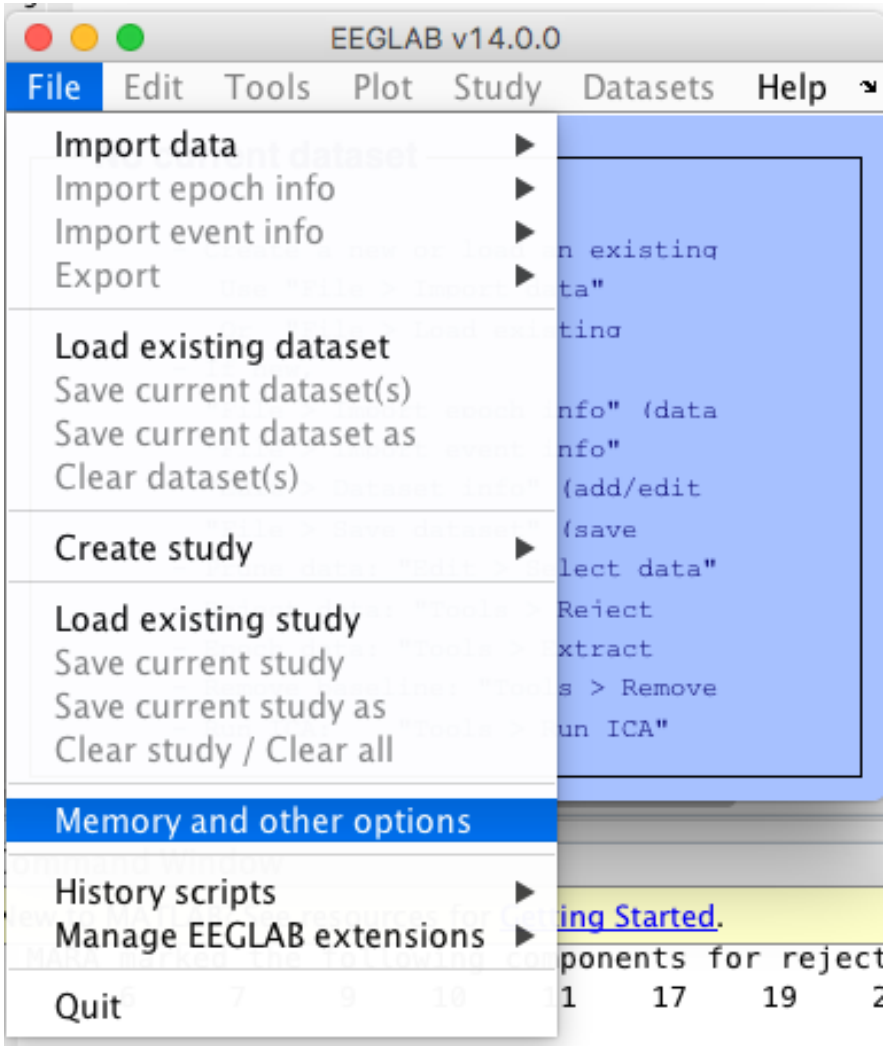
2. Open matlab and navigate into the EEGLAB folder (e.g. cd EEGLAB_14_0_0b)

3. In matlab's command line, type: EEGLAB

How to get HAPPE

II. Configure EEGLAB

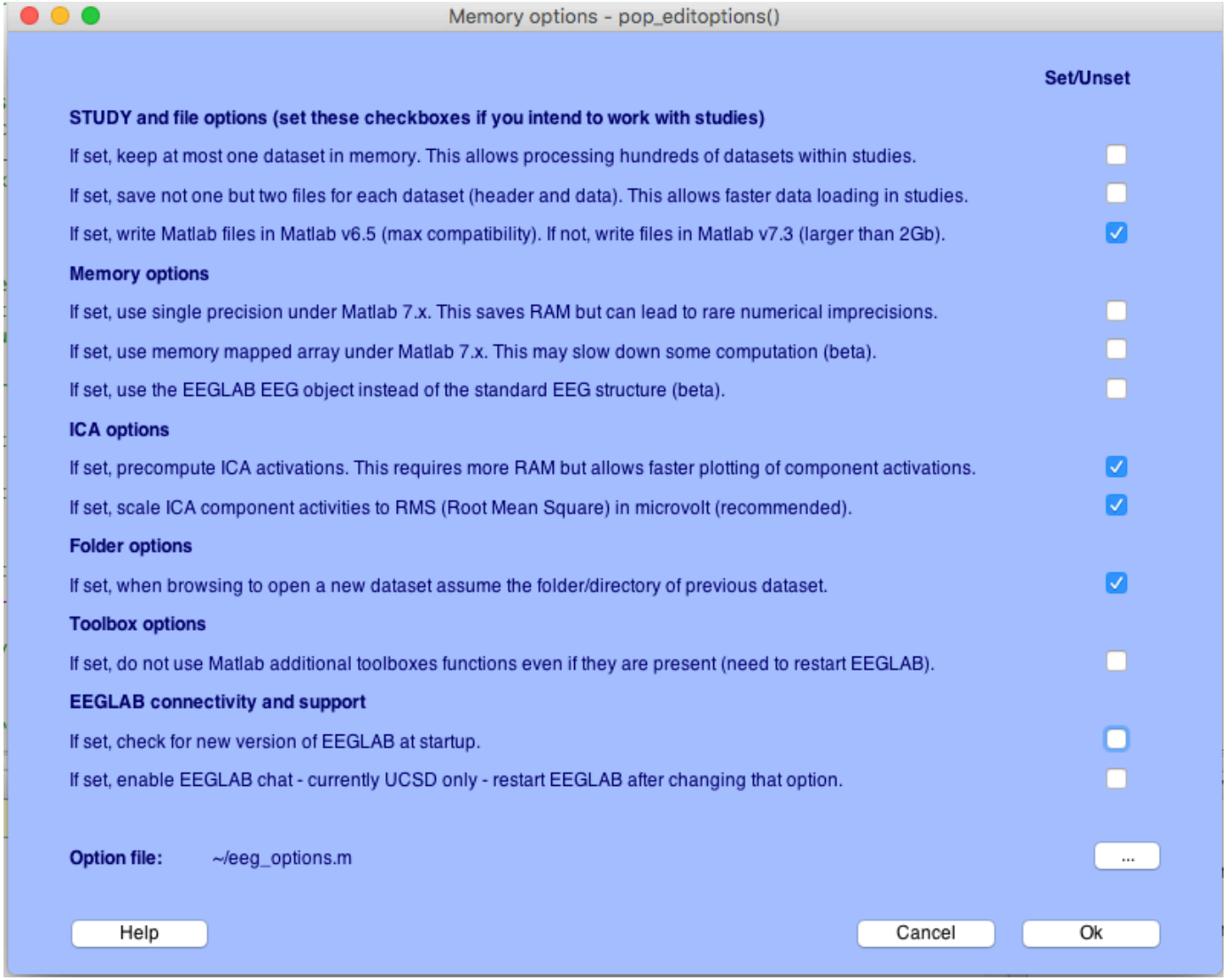
1. A periwinkle GUI will pop up. Go to File -> Memory and other options



How to get HAPPE

II. Configure EEGLAB

2. Match the checked and unchecked boxes to this image:



3. Click ok, and you are done configuring. Close out of the GUI.

Re-downloading software

In case you need to re-download the other software HAPPE calls on:

For re-downloading cleanline:

1. Download cleanline here: https://www.nitrc.org/projects/cleanline/

2. You should then move the cleanline plug-in folder into the EEGLAB_14_0b/plugins/ folder

For re-downloading MARA:

1. Download MARA plug-in from here: https://github.com/irenne/MARA
   -click the green "clone or download" button on the right & download

2. Unzip the MARA download and move the unzipped folder into the plugins folder that is inside the EEGLAB folder: Matlab/EEGLAB_14_0/plugins/

What data format works with HAPPE

RESTING-STATE DATA:

Current (version 1.0) scripts expect resting-state data to be in the NetStation Matlab format for input.

You can generate a tool within netstation to export your raw EGI data as a Matlab (.mat) format.

NOTE: the Netstation default is to export the file as .seg.mat if you are segmenting out a portion of the data, but EEGLAB will see .seg in the name and try to read in multiple trials, each as long as your file length (this will probably cause a memory error).
***PLEASE export your Netstation files as .s.mat instead to avoid this potential error.

EVENT-RELATED DATA:

Current (version 1.0) scripts expect event-related data to be in the NetStation Simple Binary format for input.

**Place all exported raw files in a single folder with nothing else, and you're ready to run HAPPE!

How to run HAPPE:

*Note

I find it's easiest to ensure that all the appropriate EEGLAB paths are added to the matlab path you are on by just opening the eeglab GUI once and closing out before I try to run any scripts that call on EEGLAB functions.
To do this, you will need to be inside the Matlab/EEGLAB_14_0_0b/ folder, and you will need to command-line type in matlab:

eeglab

You can close out of the EEGLAB GUI as soon as it opens.
Yes, I know this is a semi-superstitious way to go about the path addition, but it works and I've never had a problem this way, so I'll keep using it.

How to run HAPPE:

1. Open the HAPPE pipeline matlab script and edit the 10 user inputs at the top of the script (see next section for details about each option)

2. Run the pipeline script (green “play button” at top of screen, or type the name of the pipeline into the command line and hit enter)

3. Go do something more fun while your data is processed. Like other work haha.

HAPPE user specific inputs:

```
folder_name='/Users/laurelgabard-durnam/Desktop/mydatafolder';
```

1. Folder where the raw data lives.

    - Tell the pipeline where your data is!
    - The folder with the data should ONLY include the raw files you wish to run through the pipeline. No other folders or files should live inside that folder.
    - Here, I have put my raw data in a folder on the desktop called “mydatafolder”.

HAPPE user specific inputs:

chan_locations='/Users/laurelgabard-durnam/Documents/MATLAB/EEGLAB14_0_0b/sample_locs/GSN-HydroCel-128.sfp';

2. Channel locations.
    - You should only run the pipeline on batches of files that all used the same channel layout (since the electrode names for your ROIs will differ across layout configurations).

    - For a batch of files, EEGLAB needs to know what kind of channel layout you used and where each channel is located in space. Above, it is the 128 channel HydroCel net from EGI.

    - This information exists in the "sfp" files included in the HAPPE folder.

    - You just need to pick the sfp file that corresponds to the net you used for this batch of files, and edit the path above to match where the sfp file "lives" on your own computer.
        - If you click on the sfp file and ask for "info" about that file, the computer will show you in a pop up GUI what the path for that file is on the computer.

HAPPE user specific inputs:

chan_IDs={'FP2' 'Fz' 'FP1' 'F3' 'F7' 'C3' 'T3' 'T5' 'PZ' 'O1' 'O2' 'P3' 'T6' 'C4' 'T4' 'F8' 'F4' 'P4' 'E39' 'E40' 'E46' 'E50' 'E51' 'E41' 'E47' 'E115' 'E109' 'E102' 'E101' 'E97' 'E98' 'E103'}; example for 128-channel net

3. Channel selection. You will probably need to pick a subset of channels to run, to give ICA enough samples/channel to decompose the data robustly. Once you have selected channels, ONLY those channels will exist for future steps (you cannot recover other channels' data later if they are not in that list).

- There are a set of 18 electrodes that should always go in the channel list. MARA needs these channels (the 10-20 electrodes without Cz) to evaluate the ICA components that you create. These channels are listed in the HAPPE script and below.

- The 10-20 electrode names are assigned to the equivalent channels for the 64 or 128 channel nets. These channels are listed below for your reference.
    - 128 net channels labeled with 10-20 names: 22 (FP1), 9 (FP2), 33 (F7), 24 (F3), 124 (F4), 122 (F8), 36 (C3), 104 (C4), 58 (T5), 62 (Pz), 96 (T6), 70 (O1), 83 (O2), 45 (T3), 108 (T4), 87 (P3), 92 (P4), 11 (Fz)
    - 64 net channels labeled with 10-20 names: 11 (FP1), 6 (FP2), 15 (F7), 13 (F3), 62 (F4), 61 (F8), 17 (C3), 54 (C4), 27(T5), 34 (Pz), 49 (T6), 37(O1), 40(O2), 24(T3), 52(T4), 3(Fz), 28(P3), 46(P4)
    - NOTE for the 64 channel net, we chose electrode 3 as the Fz equivalent (Fz corresponds to both electrode 3 or 8), and channel 28 for P3 (the equivalent with the closest distance to P3), and 46 for P4 (again, closest distance to P4).
- Check if any of your ROI electrodes include the 10-20 electrode equivalents that MARA uses. If one of your ROI electrodes is also a 10-20 equivalent electrode, no need to add it twice to the channel list, it will be included already.
    - In fact, if you try to add a channel that is one of the 10-20 equivalents, the script will stop and give you an error that it could not find that channel (because you've already selected it earlier in the list, so it is not available to be selected again). If you see this error, check your channel list carefully!

HAPPE user specific inputs:

chan_IDs={'FP2' 'Fz' 'FP1' 'F3' 'F7' 'C3' 'T3' 'T5' 'PZ' 'O1' 'O2' 'P3' 'T6' 'C4' 'T4' 'F8' 'F4' 'P4' 'E39' 'E40' 'E46' 'E50' 'E51' 'E41' 'E47' 'E115' 'E109' 'E102' 'E101' 'E97' 'E98' 'E103'}; example for 128-channel net

3. Channel selection continued. To decide how many channels besides the 10-20 channels you can afford to select, follow this formula:

- (the number of channels)^2 * 30 data samples = how many total samples in time you need to decompose that number of channels
- For example, an EEG acquired with a 128-channel net and sampling rate of 500 Hz (500 samples/ second) would need at least 491,520 samples (128^2 * 30 samples), that is, 983.04 seconds of recording (491,520 samples/500 Hz) to be reliably decomposed with ICA.
- Use this formula to work backwards to how many channels you can decompose, given the length of your datasets

- For whole brain coverage with 128 channels and short recordings:
  - Option 1:select ROIs distributed across the scalp in addition to the 10-20 electrodes already included. This is akin to using a 64-channel net instead of 128. Sorry ☹
  - Option 2: process a whole session of data at once (e.g. resting-state and task data combined) to use all 128 channels. For example, if you are interested in the resting-state data, after the pipeline runs, segment out the restingstate data chunk for analysis.
  - Option 3: run HAPPE several times, selecting different channel subsets, until you have covered the whole brain.

HAPPE user specific inputs:

```
pipeline_visualizations_semiautomated = 1;
vis_freq_min = 2;
vis_freq_max = 57;
freq_to_plot = [6 10 20 30 55];
```

4. Do you want to run HAPPE in the semi-automated setting, with several visualizations (see the HAPPE workflow figure above for which steps include visualizations) during the pipeline for every file? This semi-automated version requires user input near the end (to accept the component rejection selections by MARA).

- If semi-automated, please select the frequency range you want HAPPE to plot in the processed data's power spectrum (vis_freq_min and max).
- If semi-automated, please select any particular frequencies you would like to see topoplots for (spatial distribution of power in that frequency across the scalp) in the same image as the power spectrum for that file.

Change this setting to = 0 to run the fully-automated HAPPE setting when you're ready to batch process your whole dataset without checking each file.

HAPPE user specific inputs:

```
task_EEG_processing = 0;
task_conditions = {’house’ ‘face'};
potential_eeg_var_names = {'Category_1_Segment1','Category_1'};
```

5. Are you processing event-related data (task_EEG_processing = 1) or resting-state data (=0)

If task data, please tell HAPPE the names of the stimuli conditions so it knows what events to store in the file throughout processing

If resting-state data, please tell HAPPE what the matlab file variable name(s) are that specify the EEG data within the matlab file.
If you aren’t sure what the variable is, load one of the matlab files of your data in matlab, And open the file to see the variable options (hint: the EEG variable will be very large in size probably).

HAPPE user specific inputs:

```
segment_data = 1;
task_segment_start = -0.5;
task_segment_end = 1.5;

segment_length = 2;
```

THIS IS AN OPTIONAL STEP

6. Do you want to segment your data? (optional!) if yes, segment data = 1; if you do not want to segment your data, set segment_data = 0.

If you are segmenting event-related data: please tell HAPPE how you want your segments to be structured around the event marker (IN SECONDS). Here, the segment (which should include your "baseline" period and your post-stimulus time period of interest) is -.5 seconds before the stimulus to 1.5 seconds after the stimulus. NOTE: HAPPE does not do baseline correction of the data to facilitate calculating event-related power through various approaches, some of which do baseline correction at different levels of analysis (single trial vs. trial average, vs. not at all). (task_segment_start, and task_segment_end)

If you are segmenting resting-state data, please say how long you want the regularly-spaced segments to be (IN SECONDS). Here, 2-second long segments are generated from the start to the finish of the resting-state data file. (segment_length = 2)

HAPPE user specific inputs:

THIS IS AN OPTIONAL STEP

epoch_interpolation = 1;

7. Interpolation of channels that are bad within an epoch (from FASTER program)

    - This option allows you to evaluate within each epoch whether any channels have bad data for that segment-Using only the channels that have been marked “good” channels overall from the channel rejection step. Channels flagged with bad data for that segment will then have their data interpolated only for that segment as in FASTER program.

    - The index of which channels were interpolated for each epoch in the data is stored in the intermediate3 and processed EEG outputs, in the field EEG.etc.epoch_interp_info, as rows of epoch#: channels interpolated (e.g. 34: 22 97 means in epoch 34, channels 22 and 97 were interpolated). To access a file’s details, load the file and command line type: EEG.etc.epoch_interp_info

    - Epoch_interpolation = 1 means yes, do this
    - epoch_interpolation = 0 means no

HAPPE user specific inputs:

epoch_interpolation = 1;

7. Interpolation of channels that are bad within an epoch (from FASTER program)

    - Criteria for flagging a channel as bad for a segment are 3SDs from the channel's own mean for: deviation from channel's mean across all epochs, variance across all epochs, maximum amplitude difference within that epoch (amplitude range) relative to all other epochs, median gradient (slope within the epoch) relative to all other epochs

    - For more info, see: ftp://ftp.egi.com/pub/Summer%20School%202014/Core%20EEG%20Skills%20handouts/Thursday/FASTER.pdf

HAPPE user specific inputs:

```
segment_rejection = 1;
reject_min_amp = -40;
reject_max_amp = 40;
```

THIS IS AN OPTIONAL STEP

8. Optional segment rejection

Instead of interpolating data within segments, users can instead select to reject segments that are determined to be artifact-contaminated still.

Criteria for rejection include amplitude-based criteria, and joint-probability criteria (as in EEGLAB).
Users must set the amplitude to be used for determining artifact-segments. Because both W-ICA and ICA with component rejection tend to reduce the signal amplitude, a more conservative threshold of ~40 – 50 microvolts is suggested for remaining artifact.

Outlier segments (over 3 standard deviations) on the joint probability criteria (e.g. how likely is this segment activity given the activity of other segments for that channel, and also other channels' activity for the same segment) will also be removed. The assumption is that artifact segments should be the rare (over 3 SD) segments relative to the rest of the data at this point in the processing stream.

HAPPE user specific inputs:

```
average_rereference = 1;
NO_AVERAGE_REREF_channel_subset = [20, 22];
```

9. Select how you want to re-reference the EEG data after the bad channels have been interpolated.

Average_rereference = 1 will do a re-reference across all the user-input channels.

Average_reference = 0 means you would like to re-reference the data to a single channel or a subset of channels, which you must specify in the NO_AVERAGE_REREF_channel_subset variable.

HAPPE user specific inputs:

Save_as_format = 0

10. how do you want to save the final processed data from HAPPE?

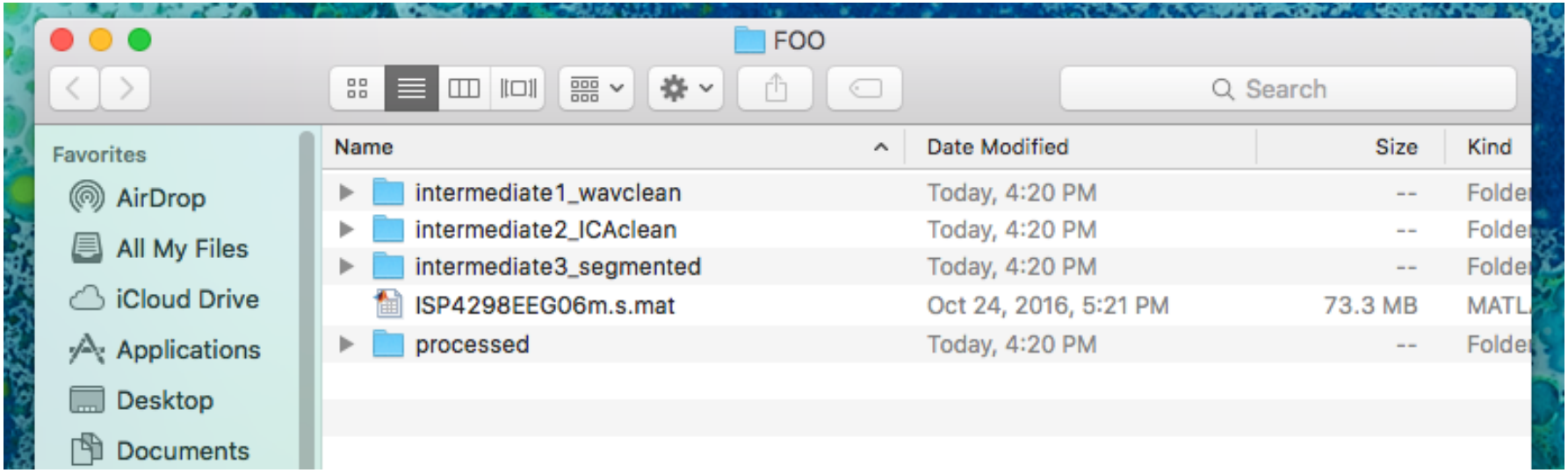
save_as_format = 2 will save the preprocessed data as a .mat file (matlab format)
save_as_format = 1 will save the preprocessed data as a .txt file
(electrodes as columns, time as rows)

save_as_format = 0 will save the preprocessed data as a .set file (EEGLAB format)

What does HAPPE output?

- Within the same data folder you specified as a user input, HAPPE will create:

1. SubFolders with intermediate file stages so you can go back and rerun some part of the pipeline or check the data at that stage
    - Wavclean contains the prewavelet-cleaned file (prewav) that is the least processed file besides the raw data, and the post-wavelet cleaned file (wavclean)
    - ICAclean contains the data with ICA components calculated on the wavcleaned data (ICA), and the data whose ICA components have been evaluated and cleaned with MARA (cleanedwithMARA)
    - If you choose to segment your data, segmented will appear and will include the segmented data and data from either post-segmentation option (interpolation or rejection) if you choose those
    - Processed is the folder with the fully-processed data



What does HAPPE output?

- Within the same data folder you specified as a user input, HAPPE will create:

2. Within the "processed subfolder", there will be an "All_subs_output_table" with the date the pipeline finished that contains data quality metrics.

Example output table:

filename	file_length_i	number_cha	number_goo	percent_goo	interpolated	number_ICs_	percent_ICs_	Percent_vari	median_artif	mean_artifac	range_artifac	min_artifact_	max_artifact_
ISP.4305.EEG	413.998	36	36	1	none	19	0.52777778	45.1209426	0.13398362	0.16565404	0.4429226	0.00259265	0.44551525
ISP4286EEG(	160.996	36	34	0.94444444	E101 T10	9	0.26470588	83.2129854	0.09582966	0.13808744	0.38942038	0.00582917	0.39524955

Helpful outputs for evaluating data quality include:
-number and percent of selected channels that were "good"
-channel IDs for the interpolated channels
-number and percent of Independent Components (Ics) that were rejected for that file by MARA
    -very high (e.g. ~75% or more, perhaps examine the file to decide keep or not)
-percent_variance_kept_of_post_waveleted_data –how much variance in the data was lost during the removal of components by MARA.
    -very low (~25%, evaluate the file, or set a threshold for your whole study)

- Artifact probability of remaining components from MARA (mean, median etc) …how much artifact contamination remained in the components that MARA kept?
    - Very high (~.25, .3, evaluate the file or set a threshold for your whole study)

***Reminder: The index of which channels were interpolated for each epoch in the data is stored in the field EEG.etc.epoch_interp_info, as rows of epoch#: channels interpolated

How to evaluate HAPPE's performance with your data?

-OBJECTIVE criteria:

-use the all_subs_output_table to evaluate the pipeline's performance across your data (in addition to using the table to make individual subject-level decisions).

How to evaluate HAPPE's performance with your data?

SUBJECTIVE criteria:

-use the semi-automated version of the pipeline to evaluate several pipeline outputs and make sure the pipeline is working well on your data before running it as batch processing

-as the "with visual" version of the pipeline runs, it will generate several images to help guide you:

1. original and cleaned spectra for selected channels Figure
    - Generated during the Cleanline line-noise removal step
    - You can see the frequencies where Cleanline removed line noise and what the post-cleaned power for those frequencies looks like
    - For data generated in the USA, you should see the red hump in the figure indicating line noise that was removed around the 60 Hz point (frequency is plotted on the x axis)

How to evaluate HAPPE's performance with your data?

SUBJECTIVE criteria:

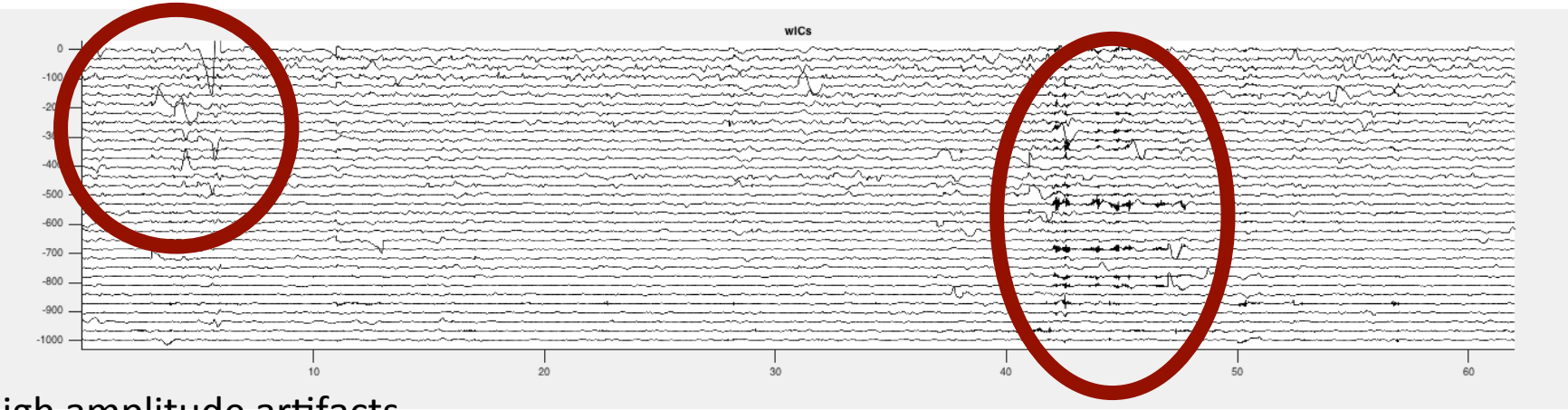
2. "Figure 1"
    - Generated during the wavelet-cleaning process
    - Top panel shows the independent components (from ICA run before the wavelet cleaning), check that the component timeseries look ok, and none have outrageously large amplitudes. Also check the first 2 or 3 components timeseries (first 3 components from the top of the graph), to make sure they are not incredibly noisy and mirror images of each other in time. If you encounter a file where the first few components look like mirror images of each other and/or have very high amplitudes, the ICA may have encountered serious problems during the decomposition. Note: I have yet to see this in any data that I have tested, so it should be very, very rare.

How to evaluate HAPPE's performance with your data?

SUBJECTIVE criteria:

2. "Figure 1"
    - Middle panel shows the artifact timeseries for each Independent Component that the wavelet cleaning is going to remove from the data
    - These timeseries should look very flat (low amplitude) except for timepoints with artifact. You should be able to see artifacts in these timeseries (high amplitude short duration peaks, or thick chunks of (EMG artifact), for example)
    - Ignore the lower panel, which just plots the difference between the two other panels, but is very hard to read visually

Example middle panel showing artifact timeseries:

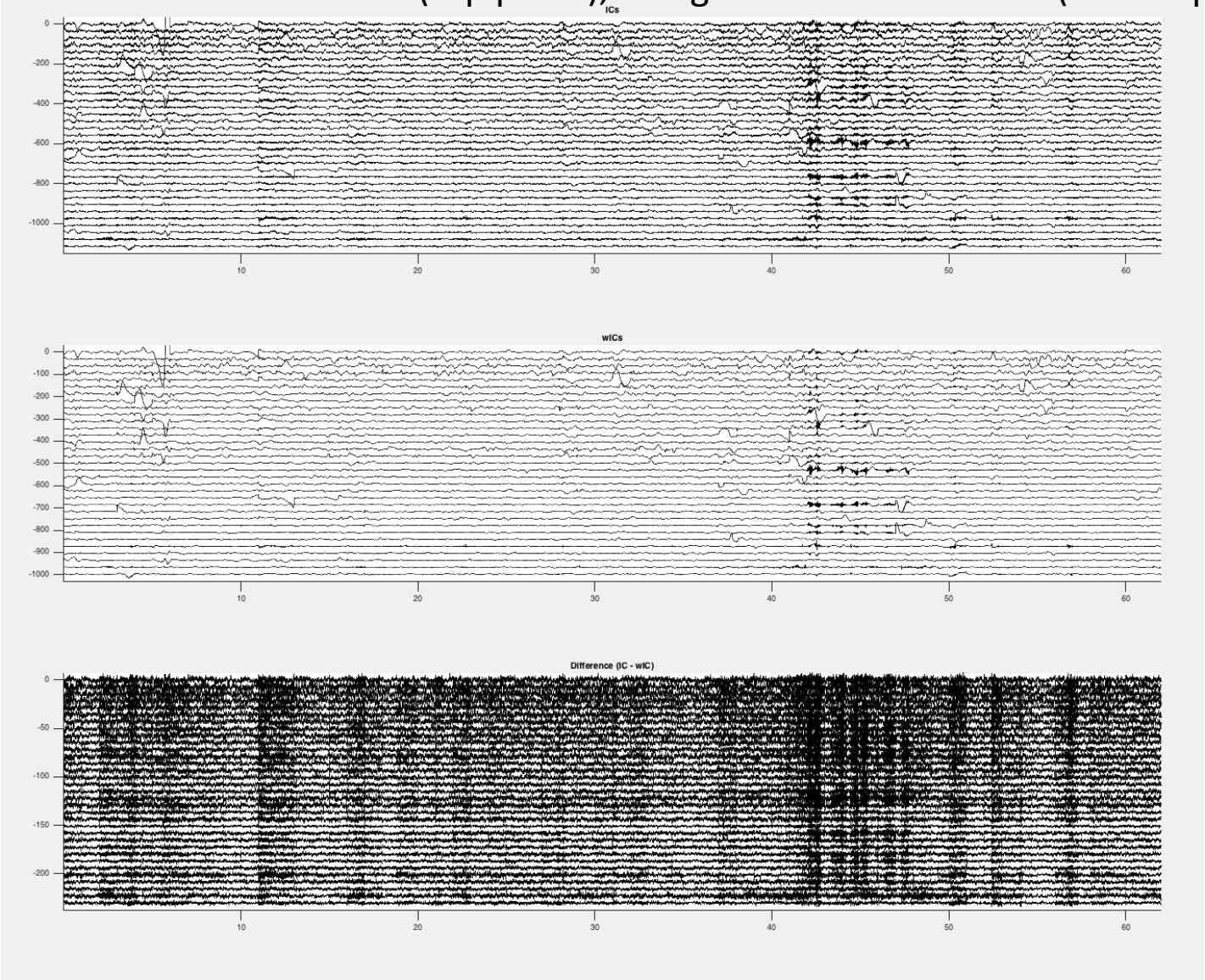


High amplitude artifacts

EMG artifacts

How to evaluate HAPPE's performance with your data?

Example of Figure 1 with successful ICA (top panel), and good artifact removal (middle panel)



How to evaluate HAPPE's performance with your data?

SUBJECTIVE criteria:

3. "ICs" Figure
    - Generated during the post-wavelet-cleaned ICA cleaning of the data
    - Each ICA-derived Independent Component of the data will be graphed (number of components will equal the number of good channels for a participant)
    - Each component's spatial topography and power spectrum are shown as figures
    - Each component's probability of being an artifact component as decided by the machine-learning classifier (MARA) will also be listed above the component's figure (e.g. Artifact Probibility = 0.02 means the probability of that component being an artifact component is .02, so it's definitely a component containing data signal, not artifact)
    - If the component is has a blue checkbox next to it, MARA has flagged it as artifact to remove from the data. The threshold for flagging is set (default) to .5 probability of being an artifact component.
    - Even in adult data, it is normal for ~50% components to be flagged as artifact. (this does NOT mean 50% of your data is thrown out. A component could account for only 1% of your data…like a single sneeze artifact in the data).

How to evaluate HAPPE's performance with your data?

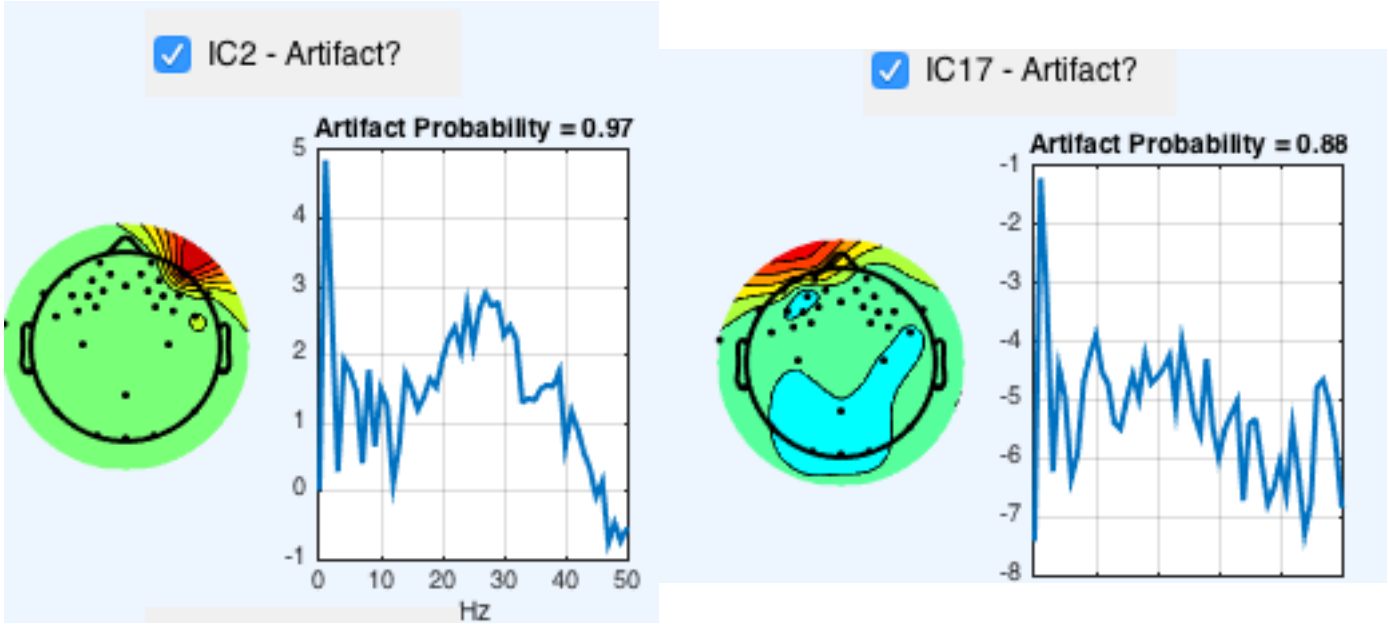
SUBJECTIVE criteria:

3. "ICs" Figure

- Evaluate the spatial distribution of the components and their power spectrum, and how MARA's classification did for the components.
    - Look for some components with single "dipole" spatial distributions, these are good data components
    - Look for components with very concentrated heat maps (red/ orange colors, or dark blue colors) around the ears, front or back of the scalp, projecting out beyond the head. These are likely EMG/artifact related components. If there is an increase in the power spectrum between 20-30ish Hz relative to the frequencies around it for that component, it is almost certainly capturing EMG artifact.
    - Check components' power spectrums. Components with good data signal will have spectrums that slope down to the right (higher frequencies), with higher values on the left (lower frequencies), and a peak in the power spectrum anywhere from 4-12 Hz, depending on the age of your participants.
    - To get more experience with classifying ICA components yourself if you'd like, you can practice here: http://reaching.ucsd.edu:8000/labelfeedback

3. “ICs” Figure:

EMG artifact component examples:



note: -the topoplot for these components has power distribution that takes up very little of the head spatially

-the topoplot for these components has power located in the front, continuing outside of the head

-the power spectrums have humps from 20-40 Hz, that is the peak EMG frequency range

-there are NOT really peaks in the power spectrum under 10Hz higher than the other frequencies

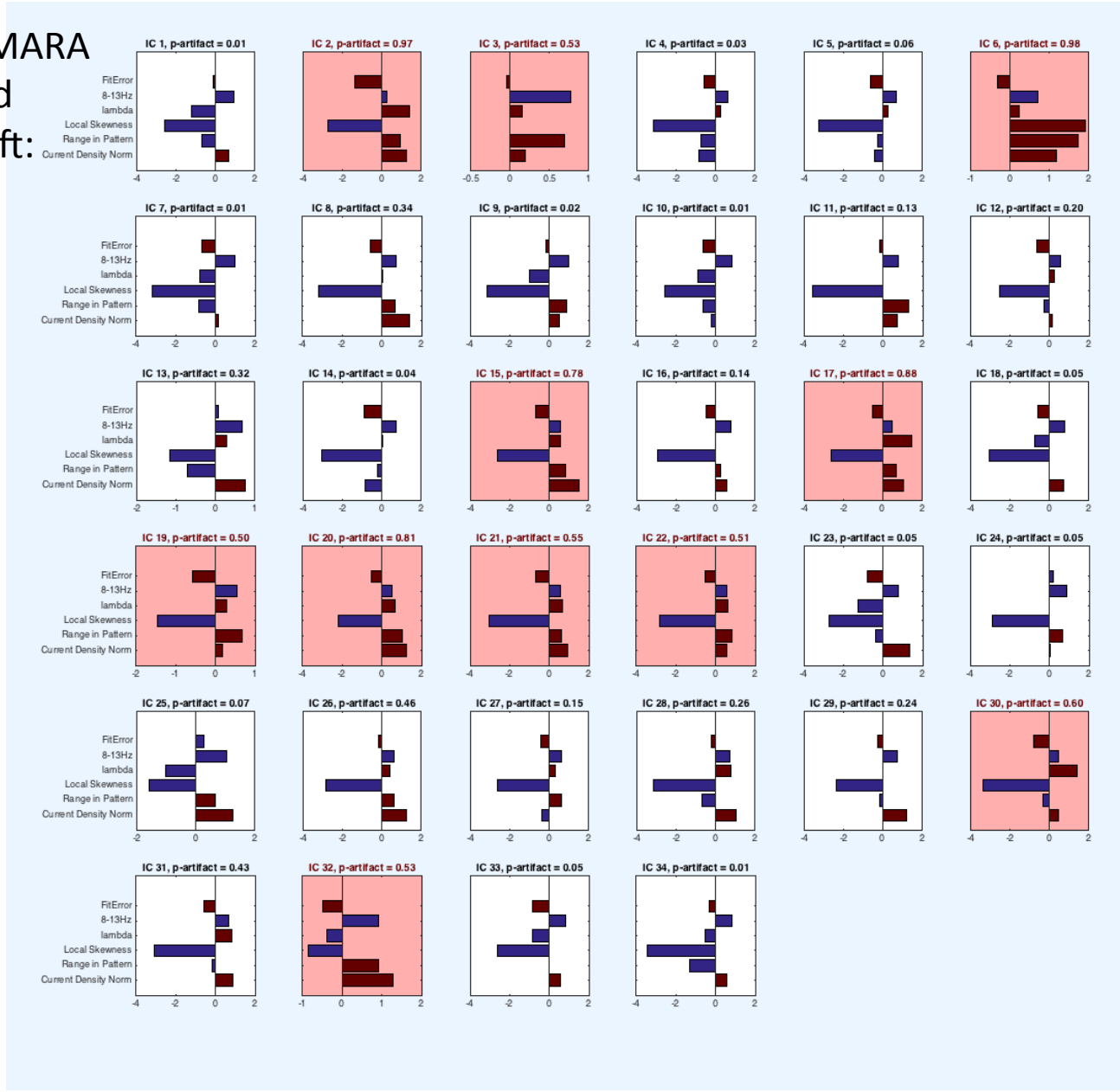
How to evaluate HAPPE's performance with your data?

SUBJECTIVE criteria:

4. "MARA criteria" Figure
    - Generated during the MARA evaluation of each ICA component as artifact or signal
    - MARA uses 6 metrics to evaluate each component and assign it the artifact probability you saw in the "ICs" Figure
    - This figure will show you how each component scored on each of the 6 criteria
        - BLUE bars indicate the component scored as GOOD on that metric
        - RED bars indicate the component scored as BAD on that metric
        - The length of the bar indicates how well/badly the component did on that metric (e.g. long blue bars = did really well on that metric)
    - Check if ALL or MOST of your components scored BAD (red bar) on any of the 6 metrics across several subjects…if you see consistent failing on at least 2 metrics, it's possible that MARA isn't doing it's best job classifying your data. You can consider changing the threshold for component rejection to be more lenient in that case (from .5 to .7 for example). That should only be done after reading all of the MARA documentation and verifying that there isn't something funky about your data first. Changing the MARA rejection threshold should be a last resort.You must go into the HAPPE script and find the MARA section to change this setting.

4. MARA Criteria example:

Metrics that MARA Uses are listed Here at the left:



IC 5 scored Really well On local Skewness

IC 6 scored Really badly On local Skewness, Range in Pattern, And current Density norm

How to evaluate HAPPE's performance with your data?

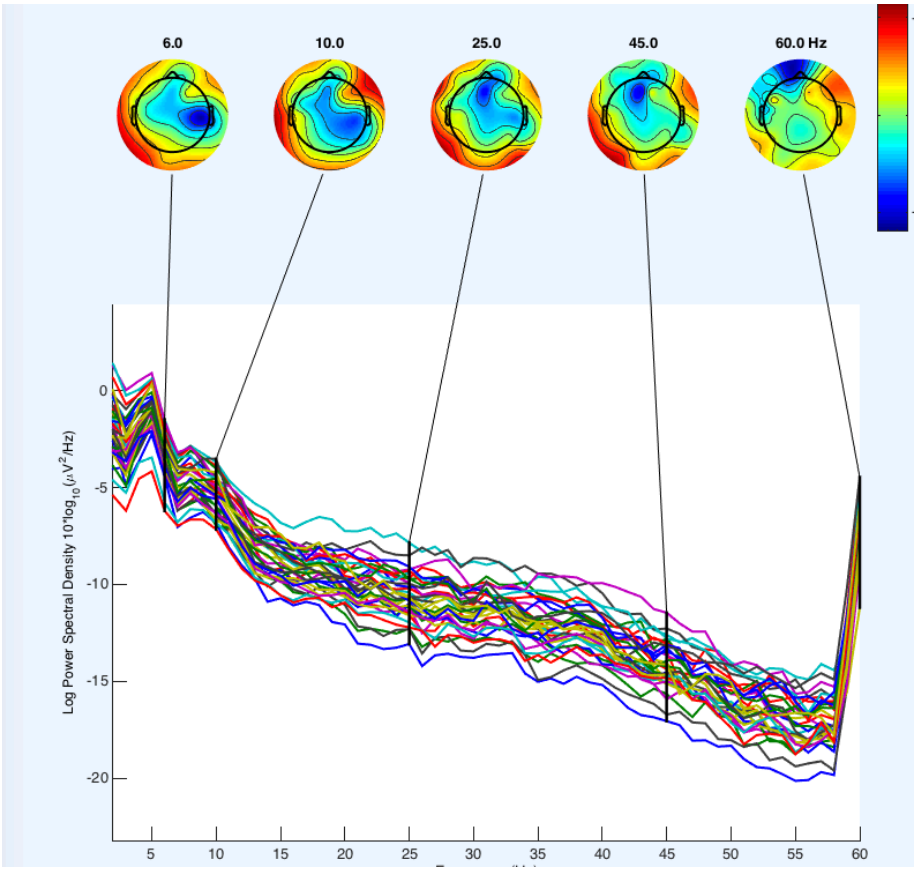
SUBJECTIVE criteria:

5. "subID_processedspectrum.jpg " Figure within the Processed Folder
    - Generated from the fully processed data
    - Each channel will be graphed with it's power spectrum in the chart.
    - Use to evaluate how clean the data looks in terms of power and spatial distribution of that power on the scalp
    - The expected power spatial distribution will vary with age
        - More posterior power for infants < 9 months is expected (warm colors in the back of the head on the map)
        - Some frontal power (warm colors in front of the head on the map) from 6 months and older

How to evaluate HAPPE's performance with your data?

5. "subID_preprocessedspectrum.jpg " Figure

GOOD DATA example



-note: alpha peak

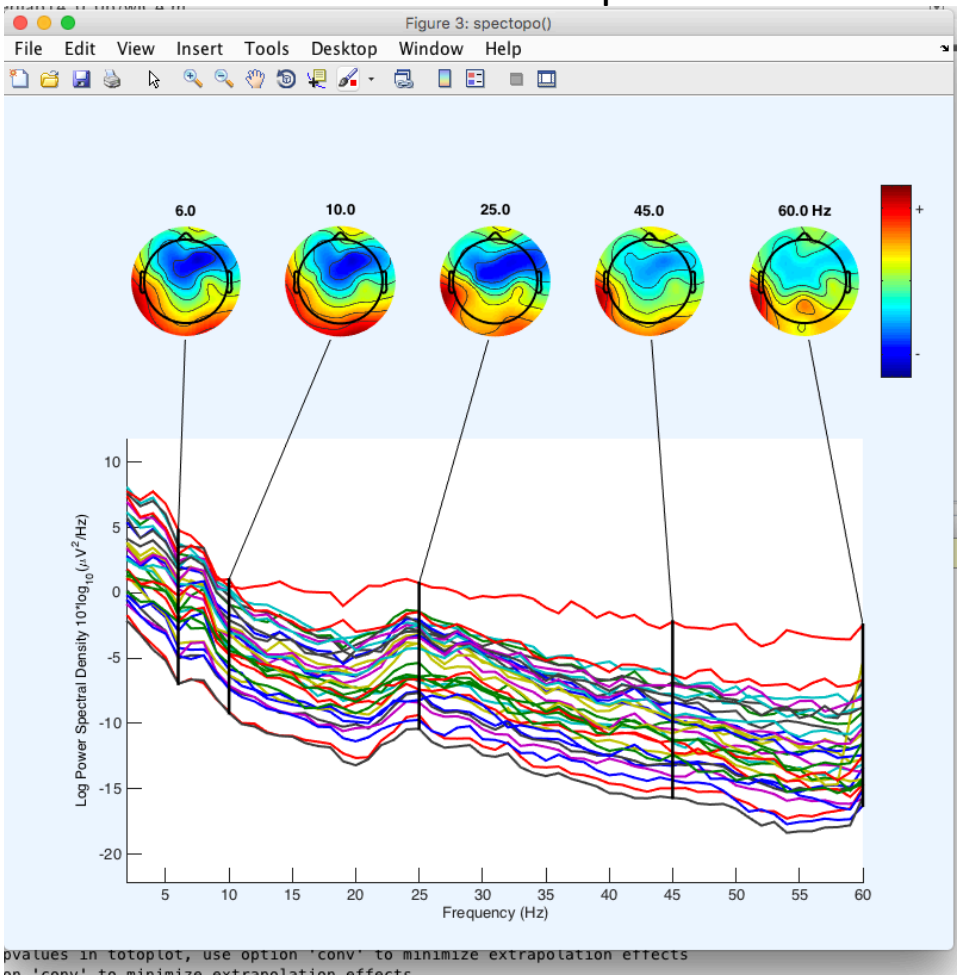
good slope downwards after the alpha peak

channels all clustered tightly with their spectrums

How to evaluate HAPPE's performance with your data?

5. "subID_preprocessedspectrum.jpg " Figure

BAD DATA example 2



-note: hump from 20-30 Hz may indicate EMG

one channel is separated from the others and may be very noisy still

HAPPE citations and references

EEGLAB:

A Delorme & S Makeig (2004) EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. Journal of Neuroscience Methods 134:9-21

Cleanline program for line noise removal from Tim Mullen:

Mullen, T. (2012). NITRC: CleanLine: Tool/Resource Info. Available online at: http://www.nitrc.org/projects/cleanline.

ICA extended Infomax algorithm justifications:

Glass, Kevin, et al. "A framework for evaluating ICA methods of artifact removal from multichannel EEG." Independent component analysis and blind signal separation (2004): 1033-1040.

Bridwell, et al. “Spatiospectral Decomposition of Multi-subject EEG: evaluating blind source separation algorithms on real and realistic simulated data.” Brain Topography (2016). doi:10.1007/s10548-016-0479-1

HAPPE employs an iteration of w-ICA first described in:

Castellanos, Nazareth P., and Valeri A. Makarov. "Recovering EEG brain signals: artifact suppression with wavelet enhanced independent component analysis." Journal of neuroscience methods 158.2 (2006): 300-312.

HAPPE employs modified code from the wICA matlab function written by Jordan Sorokin (2015) to implement w-ICA

HAPPE citations and references

MARA program:

Winkler, Irene, et al. "Robust artifactual independent component classification for BCI practitioners." Journal of neural engineering 11.3 (2014): 035013.

Winkler, et al. “Automatic Classification of Artifactual ICA-Components for Artifact Removal in EEG Signals.” Behavioral and Brain Functions 7:30 (2011).

Faster program (if you choose epoch-level channel interpolation):

Nolan*, H., Whelan*, R., & Reilly, R.B. (2010). FASTER: Fully Automated Statistical Thresholding for EEG artifact Rejection. Journal of Neuroscience Methods, 192, 152-162.

Justification for wavelet-thresholding followed by ICA rejection (note, HAPPE employs a different strategy for combining wavelet-thresholding and ICA, but this article provides empirical support for the framework of wavelet thresholding followed by ICA rejection):

Bono, Valentina, et al. “Hybrid wavelet and EMD/ICA approach for artifact suppression in pervasive EEG.” Journal of Neuroscience Methods 267 (2016).

Citation for applying ICA (in any form) in infant data:

Piazza et al. “ICA-derived cortical responses indexing rapid multi-feature auditory processing in six-month-old infants.” NeuroImage 133 (2016).

Reference (unpublished) for applying a form of w-ICA to infant data:

Zima et al. “Robust removal of short-duration artifacts in long neonatal EEG recordings using wavelet-enhanced ICA and adaptive combining of tentative reconstructions.” submitted to Physiological Measures (2017)