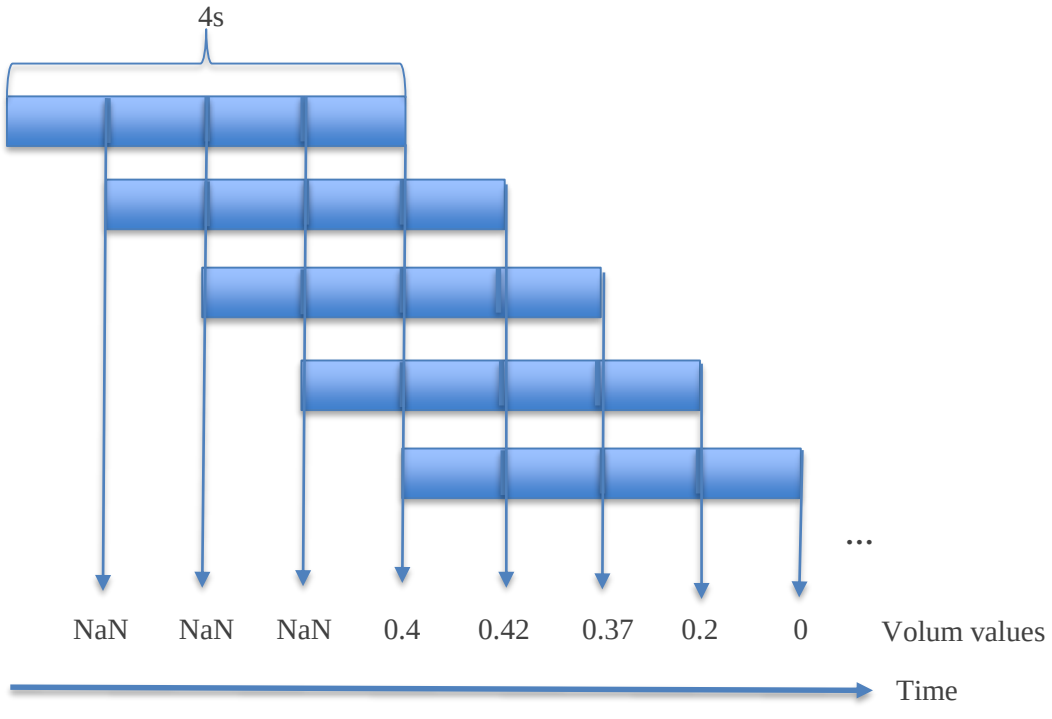
SNR

Report Names: Fanny Grosselin	Verification Name: Yohan Attal Date: Signature:
History of modifications: - *Creation 23/01/2017* : Fanny Grosselin Description of the code used to compute SNR and to transform it into volum scale.	

Introduction:

This document lists and describes the pipeline to compute SNR and to transform it into volum scale.
We compute SNR each second on a EEG signal of 4s with a sliding window of 1s. So we have 1 volum value by second but we need to wait for 3s at the beginning of the session.



1. Processes during calibration:

During calibration, we receive 2 EEG signals of 250 datapoints.
For each of these signal **(1s by 1s)**:

- Remove mean
- Filter the powerline noise (50 and 100Hz)
- Compute quality with the quality Checker. If the quality is bad (=0), we replace the 250 datapoints in the **raw signal (not in the filtered one)** by NaN values.

2. Processes at the end of the calibration:

At the end of calibration, we have 2 EEG signals of 250xY with Y the duration of the calibration in seconds.

a) Compute the thresholds for the outliers

For each of the **whole EEG signal from the calibration**:

- Remove mean
- Filter the powerline noise (50 and 100Hz)
- Apply a bandpass between 2 and 30Hz
- Set the thresholds for the outliers

b) Preprocessed calibration data

For each second, we preprocess a 4-second packet of EEG signal from calibration (sliding window):

- Remove mean
- Filter the powerline noise (50 and 100Hz)
- Apply a bandpass between 2 and 30Hz
- Detect outliers based on the whole calibration data and apply a linear interpolation on it.

c) Get the best channel in terms of quality and compute smoothed SNR

We get the whole EEG signal for each channel. Each whole EEG signal was preprocessed **4s by 4s with an overlap of 3s (sliding window)** (see 1.b). Apply MBT_ComputeCalibration with the preprocessed signals in order to get the best channel in terms of quality and to recover the smoothed SNR. Each second we have a 1 smoothed SNR computed with a 4-seconds EEG packet from the best channel during the calibration.

If MBT_ComputeCalibration is called without input, we set an index of -1 for the best channel and Inf values for SNR from calibration. An error message is also displayed on the console: "ERROR: MBT_COMPUTECALIBRATION CANNOT PROCESS WITHOUT GOOD INPUTS".

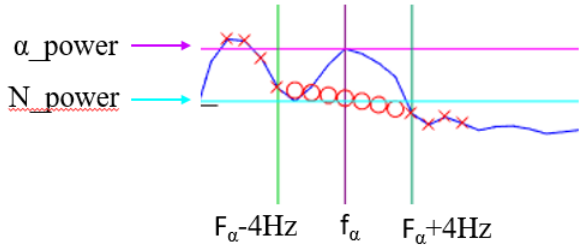
How MBT_ComputeCalibration processes?

➔ *To find the best channel:*

For each channel, we **keep only the 1-second EEG data which has a quality higher or equal to 0.5**. Then, for each channel we compute the averaged quality over the kept packets. The best channel is the channel which has the higher averaged quality. If they have both the same averaged quality, we consider the left channel as the best channel. If this best quality is lower than 0.5, we consider that we have too bad EEG signal in calibration.
In this case, we set an index of -2 for the best channel and Inf values for SNR from calibration. An error message is also displayed on the console: "ERROR: MBT_COMPUTECALIBRATION CANNOT PROCESS WITH ONLY BAD QUALITY SIGNALS".

➔ *To compute SNR*

Each second, for each **4-seconds packet** (from those which were kept in the previous step), we apply MBT_ComputeSNR (**sliding window**). This function does a spectral normalization with a differenciator function to remove the 1/f trend. Then, it finds the maximale value of the power (α_power) between 7 and 13Hz. It estimates the power of the noise (N_power) at the corresponding frequency by a linear interpolation:



X
4 points before F_α-4Hz
4 points after F_α+4Hz

O
To interpolate the points
[F_α-4Hz+1 : F_α+4Hz-1]

Finally, SNR = $\frac{\alpha\_power}{N\_power}$. If SNR<1, we set SNR to 1.

If MBT_ComputeSNR is called without input, SNR value is set to Inf and an error message is displayed on the console: "ERROR: MBT_COMPUTESNR CANNOT PROCESS WITHOUT SIGNALS IN INPUT".

➔ *To smooth SNR*

In order to translate SNR into volum during the session, we need to smooth the SNR values because SNR values change very fastly and this is not pleasant to hear. To rescale the SNR values during the session we need to normalize them with the mean and the standard deviation of the smoothed SNR values from the calibration. That is why we need to smooth SNR values from the calibration. To do that, for each SNR value computed from the calibration (**second by second on 4-seconds signals each time**), we apply MBT_SmoothRelaxIndex which averages the SNR value with the previous values:

- At the first packet of the calibration, we have not previous value, so we do not average it. *If the first smoothed SNR value is equal to NaN an error message is displayed on the console: "Four consecutive EEG packet of 1s have bad quality.". We say four consecutive EEG packet because we compute SNR on a signal of 4s. For having NaN SNR, each EEG data of 1s is also a NaN value.*
- At the second packet, we have one previous value, so we average the SNR value with the previous one. *If the second smoothed SNR value is equal to NaN, it means that the current SNR value (4s) and the previous SNR value (4s but 3s in common) are NaN. So in this case we have 5 consecutive NaN values and an error message is displayed on the console: "Five consecutive EEG packet of 1s have bad quality.".*
- At the third packet, we have two previous values, so we average the SNR value with the two previous values. *If the third smoothed SNR value is equal to NaN, it means that the current SNR value (4s) and the 2 previous SNR values (4s with 3s in common with the current studied packet and 4s with 2s in common with the current studied packet) are NaN. So in this case we have 6 consecutive NaN values and an error message is displayed on the console: "Six consecutive EEG packet of 1s have bad quality.".*
- For the fourth packet and after, we have at least three previous values, so we average the SNR value with the three previous values. *If the smoothed SNR value is equal to NaN, it means that the current SNR value (4s) and the 3 previous SNR value are NaN. So in this case we have at least 7 consecutive NaN values and an error message is displayed on the console: "At least seven consecutive EEG packet of 1s have bad quality.".*

If we have NaN or Inf values, we do not average SNR values with these values.
For example, if we have [1 1.5 1 2.2 2 1.8 1.4 NaN Inf 2.1] for previous SNR values and 1.8 for the current SNR value, the smoothed SNR value = (1.8 + 2.1) /2 = 1.95.

If MBT_SmoothRelaxIndex is called without input, smoothed SNR value is set to Inf and an error message is displayed on the console: "ERROR: MBT_SMOOTHRELAXINDEX CANNOT PROCESS WITHOUT GOOD INPUT".

If the current SNR value and the previous SNR values are equal to Inf, smoothed SNR value is set to Inf and an error message is displayed on the console: "ERROR: MBT_SMOOTHRELAXINDEX CANNOT PROCESS WITHOUT GOOD PASTRELAXINDEX IN INPUT".
For example, if we have [1 1.5 1 2.2 2 1.8 1.4 Inf Inf Inf] for previous SNR values and Inf for the current SNR value, the smoothed SNR value = Inf.
If we have [1 1.5 1 2.2 2 1.8 1.4 Inf 1.8 2.1] for previous SNR values and Inf for the current SNR value, the smoothed SNR value = (1.8 + 2.1) /2 = 1.95.

If the current SNR value and the previous SNR values are equal to NaN, smoothed SNR value is set to NaN.
For example, if we have [1 1.5 1 2.2 2 1.8 1.4 NaN NaN NaN] for previous SNR values and NaN for the current SNR value, the smoothed SNR value = NaN.
If we have [1 1.5 1 2.2 2 1.8 1.4 NaN 1.8 2.1] for previous SNR values and NaN for the current SNR value, the smoothed SNR value = (1.8 + 2.1) /2 = 1.95.

3. Processes online during the session:

We need a vector of float (pastRelaxIndex) to hold the relaxation indexes of the session.
During session, we receive 2 EEG signals of 250 datapoints. We first compute the **quality checker 1s by 1s**:

a) Compute the quality of the signals
- Remove mean
- Filter the powerline noise (50 and 100Hz)
- Compute quality with the quality Checker. If the quality is bad (=0), we replace the 250 datapoints in the **raw signal (not in the filtered one)** by NaN values.

These raw signals with NaN values will be stored in a matrix and will be used by Katerina to compute some statistics about the relaxation state.

To compute the relaxation index, we will work 4s by 4s with a sliding window of 1s like during for the calibration.
For each second with 4s-EEG signal, we apply all these steps:

b) Preprocessed session data
- Remove mean
- Filter the powerline noise (50 and 100Hz)
- Apply a bandpass between 2 and 30Hz
- Detect outliers based on the whole calibration data and apply a linear interpolation on it.

c) Get the SNR value of the session data

Apply MBT_ComputeRelaxIndex with the preprocessed signal of the session and the index of the best channel. This function allows to get the SNR value of the best channel (defined in the calibration) thanks to the call of MBT_ComputeSNR on the SNR value from the best channel. The SNR value is computed like in the calibration. So we have two EEG signals in input but only one SNR value (the one from the best channel). If all the EEG values from the session packet is equal to NaN (because of the quality checker which set the EEG values to NaN when the quality is bad), the SNR value is set to NaN.

If MBT_ComputeRelaxIndex is called without EEG packet in input or an index of -1 or -2 for the best channel, the SNR value from the session is set to Inf and an error message is displayed on the console: "ERROR: MBT_COMPUTERELAXINDEX CANNOT PROCESS WITHOUT GOOD INPUTS".

d) Smooth the SNR value of the session data

Push each new computed SNR value in pastRelaxIndex.
Apply MBT_SmoothRelaxIndex to smooth the last SNR value with the SNR values of the past. To smooth the SNR value, we do the same process as it is described for the calibration.

See the corresponding section to see the possible error messages and values to handle a problem.

e) Normalize the SNR value of the session data

Apply MBT_NormalizeRelaxIndex to normalize the smoothed SNR of the session with the smoothed SNR values from the calibration:

$$\text{Normalized smoothed SNR value} = \frac{Smoothed\ SNR\ value - mean(Smoothed\ SNR\ values\ from\ calibration)}{std(Smoothed\ SNR\ values\ from\ calibration)}$$

If MBT_NormalizeRelaxIndex is called without Smoothed SNR values from calibration in input, the normalized SNR value is set to Inf and an error message is displayed on the console: "ERROR: MBT_NORMALIZERELAXINDEX CANNOT PROCESS WITHOUT GOOD INPUTS".

f) Convert the smoothed normalized SNR value of the session data into volum

Apply MBT_RelaxIndexToVolum to transform the smoothed normalized SNR into volum. This function applies the hyperbolic tangent to put the SNR value in a range of [-1:1]. Then, we add +1 in order to have SNR value in a range of [0:2]. Then, we divide by 2 for having a SNR value in a range of [0:1]. Because we want no volum when the person is relaxed, we apply this formula: volum = 1 – SNR value.

If SNR value = Inf, we set the volum to -1 and we have an error message which is displayed on the console: "ERROR: MBT_RELAXINDEXTOVOLUM CANNOT PROCESS WITHOUT SMOOTHEDRELAXINDEX IN INPUT".

If SNR value = NaN, we set the volum to 0.5. This can appear when for example, we have at least 4 consecutive packets with bad signal. Indeed, the quality checker set to NaN values when the quality is bad.
When SNR value = NaN, a message is displayed on the console: "At least four consecutive EEG packets have bad quality."

See SchemaComputeSNR_v1.0_2017_01_30_FAG.pdf to see the schemas.