EEG signal quality estimation

The Dreem headband has a set of 4 EEG channels. Two are recordings between the forehead and the mastoids while the two others are recording signal on the forehead.

In a medical environment, wet electrodes are used to record EEG. They are glued to the head skin so that the signal is recorded all night long and has a good quality.
On the Dreem headband, such a process of sticking electrodes is obviously excluded. Instead, dry electrodes are used. They are pieces of conductive polymer sensitive enough to record EEG. The quality of such signal depends on the quality of the contact between the electrode and the skin. The contact can vary throughout the night if the sleeper is moving. The illustration bellow shows a window of 30 second of EEG signal recorded with the Dreem headband on the 4 channels. The quality of this signal is not the same for every electrodes.

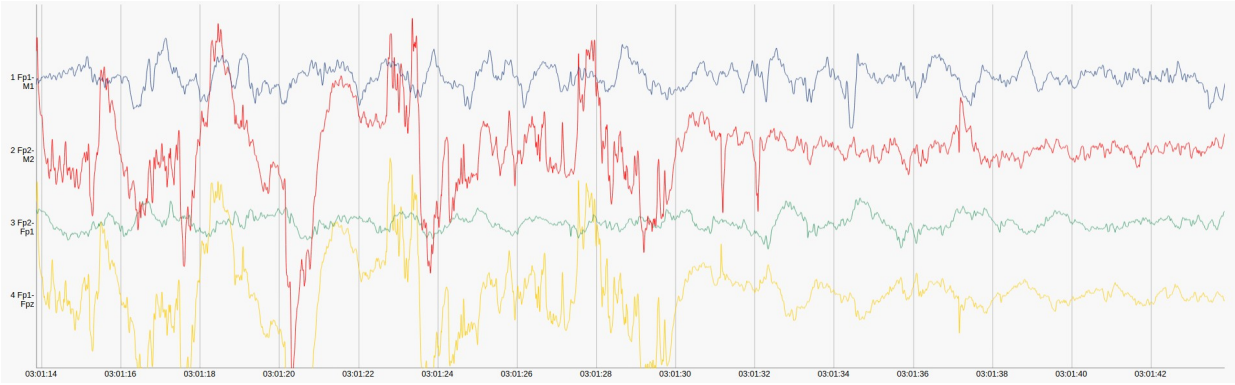

Illustration: Signal 1 (blue): frontal-mastoid filtered EEG signal. This signal has good quality.
Signal 2 (red): frontal-mastoid EEG filtered signal. This signal has band quality at the beginning.
Signal 3 (green): frontal-frontal filtered EEG signal. This signal has good quality.
Signal 4 (yellow): frontal-frontal filtered EEG signal. This signal has bad quality at the beginning.

To have good performance and robustness on sleep staging and sound stimulation along the night (dreem headband purpose), we need to detect when the signal has good quality otherwise we could misinterpret the signals. Which is evil.

Expert approach to detect the quality (like using threshold on amplitude, derivation etc.) is tedious and fixing parameters can be a real problem. This is why we use data-driven models to estimate this quality.

The exercise we propose here is to implement an algorithm for quality detection.

You are provided:
- 'quality_dataset.h5' a dataset of windows of 2 second of EEG signal labeled 0 (bad quality) or 1 (good quality). Columns description are provided in the file but in short they are a concatenation of 2 seconds of raw signal (250 Hz, 2 seconds = 500 columns) and the corresponding 2 seconds of filtered signal (250Hz, 2 records = 500 columns also) for a total of 1000 columns. Raw signal is the output signal of the electronic side. Filtered signal is the raw signal filtered between 0.5 and 20 Hz thus removing DC signal and uninteresting frequencies. For visualization we use the filtered signal (like in the figure above).
- 'record1.h5' a record with the signal for the 4 channels.
- 'record1_good_quality.csv': a CSV file with 4 column corresponding to the 4 signals of record1.h5. The value is 1 when the quality is GOOD. (0 means no information, it can be good or bad quality.)

- 'record1_bad_quality.csv': a CSV file with 4 column corresponding to the 4 signals of record1.h5. The value is 1 when the quality is BAD. (0 means no information, it can be good or bad quality.)
- 'record2.h5': one more record with 4 channels. For this one no CSV are attached but you can test your algorithm and visualize the result.
- 'quality_detector.py' a file where you can implement your algorithm
- 'easy_import.py' to easily open 'record1.h5' and 'record2.h5' raw and filtered signal

Instructions
- Use 'quality_dataset.h5' to train a data-driven model that you can use to estimate the quality of 'record1.h5' and 'record2.h5' at each instant.
- Use python and comment your code when it is important.
- Make a report describing your method, ideas results. The report is crucial for us, we want to see how you think and what you think. Be verbose and draw pictures!
- Use git (github or bitbucket preferably) to manage and version your code so that we can follow your progress. (You can also include your report in the repo). Do not hesitate to commit frequently.

Advice:

1. Use h5py to manage '.h5' files “http://docs.h5py.org/en/latest/quick.html”.
If you have trouble installing it upgrade your pip version before:
> pip install pip –upgrade
> pip install h5py

2. http://rythm.s3-website-us-west-1.amazonaws.com/rythm_whitepaper.pdf You can find information about our technology
3. Don't try to overkill your model training, your approach to solve the problem is as important as your results.
4. Of course, read the instruction carefully.