

First Steps to Automatic Interpretation of Electroencephalograms

Valério Ramos Batista¹

Federal University of ABC, St André - SP, Brazil

Received on September 25, 2016 / accepted on October 25, 2016

Abstract

We present a hypnogram generator fully written in Matlab/Octave programming languages. Our software extracts data from brainwaves during sleep and then classifies wave stretches according to their predominant frequency. Differently from other programs, which rely on heavily numerical data analyses, our approach is the plain shape analysis of the waves. For this purpose we focus on properties of the waves that are purely geometrical. The result is a fast and reliable algorithm that was implemented with only 135 lines of source code.

Keywords: Image processing, shape analysis, electroencephalograms.

1. Introduction

The early history of Electroencephalography (EEG) is composed of many important names from Medical and Biological Sciences, specially Physiology and Psychiatry. Here we give a very brief exposition: in 1875 Richard Caton made the first electrical measurements of small mammals' brains. He could not record these measurements, which was later achieved by Vladimir Pravdich-Neminskii in 1913. The first machine dedicated to measuring and recording such electrical activities was finally invented in 1924 by Hans Berger, and he was also the first who applied it to humans. Details about the history of EEG can be found in [1, 2, 3].

In this work we deal with EEGs recorded at sleep and our databank is described in [4, 5]. They are supposed to cover the whole average sleep period of 8h. This is because many disorders can only be properly studied if compared with the so-called *normal cases* (see [6] for details). Figure 1 shows a standard graph of sleep stages. Mental disorders like amnesia, schizophrenia and Tourette syndrome deviate from the normal sleep cycle. The hypnogram is a helpful tool in the diagnosis of such disorders.

¹E-mail: valerio.batista@ufabc.edu.br

still depend on handwork because the error rate of automatic analyses can be considerably high. For instance, in [8] the authors compare outputs of the ASEEGA software with expert consensual scoring. They obtain up to 17.5% of discrepancy between them. However, in that work the authors show evidence of the good overall performance of ASEEGA. We shall discuss their work again in Section 8. Please access

<https://aseegaonline.com/pub/doc/FAQ.html>

for more information about ASEEGA.

In fact, when one considers the mostly used softwares for automatic analysis of EEG, they still present considerable discrepancies with experts' scorings. Moreover, experts can also come to quite distinct diagnoses. For instance, in [9] the authors draw statistical conclusions for the case of multiple sclerosis diagnosed by different specialists. Unfortunately, we have not found similar and *recent* studies for EEG in the literature yet, and therefore comparisons will be drawn at a minimum here.

Most software tools rely on the *numerical analysis* of data. In the case of brainwaves they can be studied via *Fourier spectral method*. This method requires the explicit function to be decomposed in frequencies. In EEG each channel captures the brainwave and its oscillation is then printed on a computer screen. The printed wave can be seen as the graph of a function but its equation is unknown. The software filters noise and then *interpolates* sample points of the graph within a certain time interval $[0, T)$, and this finally gives $f : [0, T) \rightarrow \mathbb{R}$.

Most of the time brainwaves are far from being periodic, even after filtering. However, for a spectral analysis of f we must extend it to a periodic function P . What softwares normally do is to take $P : \mathbb{R} \rightarrow \mathbb{R}$, $P(t + nT) = f(t) \forall n \in \mathbb{Z}$. Here we give a practical example that is also critical. Suppose $T = \pi$ and f is the sinus function. By taking the aforementioned P we have $P(t) = |\sin(t)| \forall t \in \mathbb{R}$. Hence the corresponding Fourier series is $P(t) = \frac{2}{\pi} - \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{\cos(2nt)}{4n^2 - 1}$, as can be readily checked in [10, Page 44]. However, for $T = 2\pi$ that infinite sum is replaced with a single term, namely $P(t) = \sin t \forall t$. Of course, the Fourier spectral method is a powerful tool in the case of *stationary signals*. But this is not characteristic of EEG-signals. An alternative could be the *Fast Wavelet Transform* (FWT), which still relies on numerical analyses (see [11] for details). However, we have not found a free hypnogram generator with the FWT yet in order to compare it with our method.

1.1. Contributions

Instead of a numerical analysis we propose a *shape analysis* of the brain-

waves. Our method is based on the theory of *Manifold Learning*, which is classically composed of *Machine Learning*, *Statistics* and *Dimensionality Reduction*. This is a relatively recent theory, of which the first publications are [12, 13] and [14]. Many general-purpose algorithms have been developed upon Machine Learning techniques. However, here we focus on a dedicated software that interprets brainwaves by attributing colour patterns along their image, and then computing the statistical distribution of these patterns. See Figure 2. In this sense manifold dimension is reduced from one to zero because we shall end up with an array of integers. However, machine learning is not applied because what counts is the direct relation between the types of brainwaves and the corresponding pattern distribution. For example, high frequency waves of small amplitude correspond to a high density of magenta pixels, as can be seen in Figure 2.

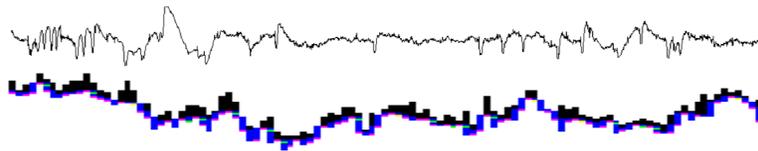


Figure 2 - Stretch of brainwave and a zoomed detail after colouring.

1.2. Related work

The aforementioned EMBLA and ASEEGA are professional softwares. Later we shall discuss EEGlab and EDFbrowser, which are both open source programs. In fact, our work aims at contributing to EEGlab for two reasons. First, EEGlab has no automatic hypnogram generator yet. Second, EEGlab can easily incorporate our program since both run in Matlab. Of course, our codes still need improvement beforehand, something we intend to achieve in a near future.

However, differently from any EEG software ours makes use of a discrete method, namely based on pixels. So it neither filters frequencies nor computes differentials. But, of course, the wave shape tends to be moulded by the predominant frequencies and amplitudes. That is exactly what the method is based on. Other softwares on EEG interpretation are based on numerical data analyses. Our approach uses plain shape analysis, and we claim that it is original in this sense.

2. Open-Source Codes

Figure 2 (top) was obtained from EEGlab, an open-source program originated from a Matlab toolbox launched in 1997 by Scott Makeig and his colleagues [15]. That toolbox applied the computational method of *Independent*

Component Analysis to process EEG-signals, and also included functions to plot EEG-data (time/frequency analyses, EEG field map, etc.) From that time on the number of EEGLab developers has grown consistently and this software is now in the version 13.4. As part of the GNU Project, EEGLab is freely distributed under the GNU General Public License. Its code is open and fully written in Matlab programming language.

Currently, EEGLab consists of circa 400 functions that sum up approximately 50 thousand lines of source code. Many of its users belong to the official Discussion List that is growing towards 4 thousand participants from nearly 100 different countries. More details including download, running and tutorials can be found at <http://sccn.ucsd.edu/eeglab>

3. Technical background

In spite of the innumerable capabilities of EEGLab, in our work we shall utilise only a few of them. We present our technique in 3 steps. **1st:** We start with EEG-datafiles that are either in the standard EEGLab format or can be converted into it. For instance, an EEG datafile in *European Data Format* named `brainwvs.edf` can be imported into datafiles `brainwvs.set` and `brainwvs.fdt` that are the standard formats used by EEGLab. It always works with these two formats, in which the SET-file contains metadata about channels, sampling and others, and the FDT-file has all the numerical brainwave data. **2nd:** For any given channel we use EEGLab to generate a pair of SET/FDT-files that show the brainwave from that single channel. Then we load the single-channel data in EEGLab, which plots them for each time interval of 30s through the functions `eegplot()` and `pop_eegplot()`. **3rd:** While each figure is displayed on the screen our program invokes the Matlab function `print('-dtiff',s)`, where `s` specifies a different TIF-file for each figure. Finally we have a folder containing several TIF-files that will be submitted to our shape-analyser. Its written in Octave and also runs in Matlab.

Since Octave is also a freely distributed open-source program of the GNU Project we decided to adopt it in our research as well. Moreover, there have been efforts to use Octave as a platform of EEGLab, although Matlab still leads in terms of programming facilities.

4. Preliminaries

As explained in the Introduction our approach is the shape-analysis of brain waves. In order to obtain a reliable and fast algorithm we must understand how to estimate the frequency of a wave just by the naked eye,

and then translate this task into colouring pixels along the wave. Roughly speaking, predominant frequency of waves at sleep and their classification could be summarised as low, medium and high for delta, theta and alpha, respectively. Just by looking at a wave like in Figure 2 we see that it is delta, even if interfered with noise. Our algorithm colours only the lower part of the wave in order not to penalise speed. Indeed, every oscillation is made of peaks and valleys. These latter are the approximate mirror image of that formers and vice-versa. Therefore, we can just consider half of the object in our analysis.

Figure 3 depicts the output of our frequency tracker. Notice that it captures only the predominant frequency along the time. Comparing it with Figure 2 will help understand our algorithm: the yellow pixels determine when this frequency changes value. Details will be given in the next section but for now Figure 3 serves as a good bird's-eye view.

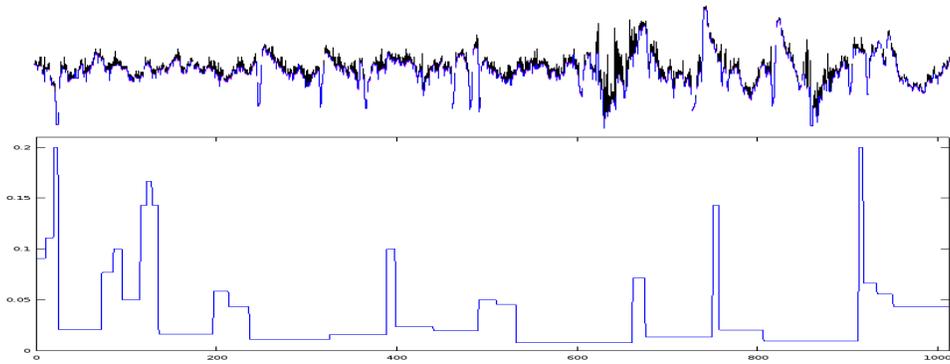


Figure 3 - Coloured 30s-stretch and its readout by our frequency tracker.

Anyway, notice that the highest frequency in Figure 3 is still lesser than 3Hz. As a matter of fact this figure shows only part of an NREM-IV state. Delta waves predominate in this case, and that is why the frequency remains so low.

5. The Algorithm

As mentioned in Section 4 our algorithm colours the lower part of the wave. As a matter of fact, the colouring works like leaving a trail while going along the lower contour of the wave. We only need four colours to reprint pixels, chosen as green, blue, yellow and magenta. In this paper we also use the fifth colour cyan but just for explanations. The predominant frequency of the wave on a certain stretch actually depends on the density

of yellow and green pixels there. The idea behind it is rather geometrical than numerical, so that our focus on *shape* is indeed in the strongest sense.

Figure 4 illustrates the geometrical meaning of the colours, which we explain right below.

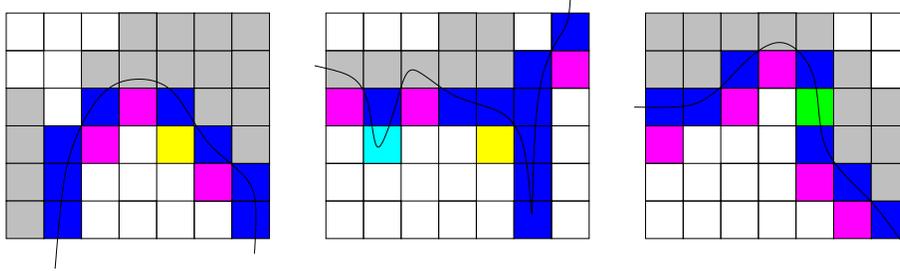


Figure 4 - Geometrical meaning of the coloured pixels.

Blue, green and cyan pixels are originated from black ones (*grey* in the picture), whereas magenta and yellow come from white pixels. By following the curves in Figure 4 with your right hand, the little finger will only pass through either black, blue, green or cyan pixels. After colouring, if a pixel remains black this means that it does not belong to the lower part of the wave. Colours and their respective meanings are summarised in Table 1.

COLOUR	CHARACTERISTIC	MEANING
Blue	mostly in large groups	smooth stretch
Cyan	always isolated	abrupt change of direction
Green	always isolated	beginning of a sharp descending
Magenta	isolated or in small groups	sharp curving or smooth ascending
Yellow	always isolated	beginning of a smooth descending

Table 1: Pixel colours, their characteristics and geometrical meanings.

According to Table 1 the higher the density of yellow and green pixels within a certain stretch, the higher the predominant frequency. As a matter of fact, cyan pixels are also green in the output of our program, but we do not count them because it is mostly noise that causes abrupt small oscillations in the wave. Some magenta pixels mark the beginning of a smooth ascending, but counting them would just double the number of yellow pixels. Moreover, since magenta has multiple meanings it would be hard to decide when to count them or not.

Although yellow and green are the only colours of interest in this present work, we have kept the four-colour scheme for two reasons: 1) it can be useful to future developments of our software, for instance the inclusion of

automatic interpretation of other EEG-features; 2) it makes our algorithm *shorter* than it would be with the two-colour scheme. Indeed, this second reason will be clear with the detailing of Algorithm 1 presented in our tutorial. See how to access it and also Algorithms 2,3 in the next section.

6. Experiments and Results

The main result is the software `hpg`, which uses our tool `tkp` for taking pictures. At <https://sites.google.com/site/vramos1970> in the link *Softwares* one can download the Matlab programs `tkp.m` and `hpg.m` in protected code by accessing the file `hg.zip` of that homepage. This file includes a tutorial with tests, so that here we can summarise our discussions.

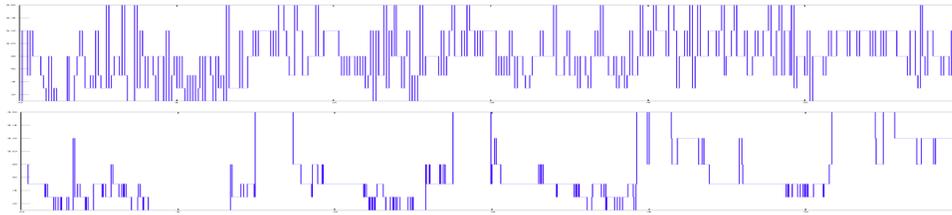


Figure 5 - Hypnogram of 6h30s of a patient's sleep time generated by our software (above), compared with the corresponding annotations by a polysomnographic technician in graphical output (below).

Now we must explain that Figure 1 is just didactic, because changes between sleep stages are at least as frequent as shown in Figure 5 (bottom). Comparisons with our output are drawn in the next section.

7. Conclusions

Our hypnogram is consistent with the fact that stages **III** and **IV** predominate at the beginning of the sleep, then R and W start appearing and alternating with the NREM states. Afterwards, the predominance remains between **II** and R, this latter becoming more frequent at the end of the sleep. The same behaviour is shown in Figure 1 (top and bottom). They do not match in general but there are more coincidences than mismatches. Here we list some possible reasons to explain the discrepancy: **1.** Polysomnographic technicians consider the whole Polysomnography (PSG) to establish the hypnogram. Besides the EEG, a typical PSG also includes Electrooculogram (EOG) and Electromyogram (EMG). Here we use only two combinations of EEG: channel 1 (Fpz-Cz) and channel 2 (Pz-Oz). **2.** As a matter of fact, we only took channel 1 to generate Figure 5 (top). This is because channel 2 presents too high variations of amplitude, as depicted

in Figure 6. **3.** However, one could still get a good approximation of the ideal hypnogram by sticking to essential data. As a matter of fact, in [8] the very authors discuss the inconvenience of working with too many data and present an overall good performance of the software ASEEGA by studying just *one* EEG channel. **4.** In our case, Figures 5 (top and bottom) differ with respect to variations between *neighbouring* stages. As a matter of fact, polysomnographic technicians must frequently resort to guessing the correct sleep stage at classifying them. That is why their annotations will not match perfectly, although the overall hypnogram is expected to be similar. **5.** However, it is clear that our automated method registers too many variations during the sleep. Even if realistic, maybe this depth of detailing can in practice be of less help than a plainer hypnogram as in Figure 5 (bottom).

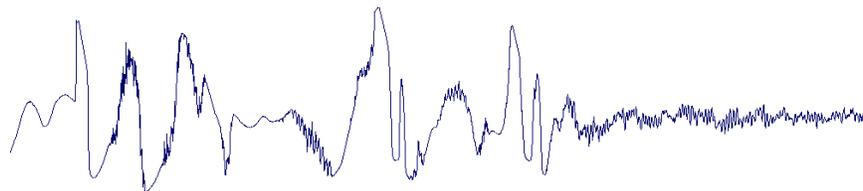


Figure 6 - Channel 2 at about 6:12am.

Anyway, we must emphasise that our software is rather devoted to research purposes than to diagnosis. It still needs improvement and validation, which is only possible after many tests and comparisons with expert scorings. Of course, we hope it will be useful in EEG analysis, but we actually focus in our *computational* method: instead of the costly numerical analysis we propose a *shape analysis* that is very simple, fast and reliable.

Acknowledgements. This work concludes my post-doctoral research in Computer Science, supervised by Prof Alexandre Levada at the Federal University of São Carlos. Valuable information about EEG was provided in details by the following members of the Psychobiology Dept of the Federal University of São Paulo: Prof Orlando Bueno, Prof Sabine Pompeia, Psychologist Silmara Batistela and Researcher Diego Mazzotti. My gratitude goes to all of them for their help, support and attentiveness to my studies.

References

- [1] Savoy, R.L. History and future directions of human brain mapping and functional neuroimaging, *Acta Psychologica* 107(1):9-42, 2001.

- [2] Swartz, B.E., Goldensohn, E.S. Erratum to: “timeline of the history of EEG and associated fields” [Electroencephalography and Clinical Neurophysiology 106(2):173-176, 1998]q, *Electroencephalography and Clinical Neurophysiology* 106:173-176, 1998.
- [3] Swartz, B.E., Goldensohn, E.S. Timeline of the history of EEG and associated fields, *Electroencephalography and Clinical Neurophysiology* 106(2):173-176, 1998.
- [4] Kemp, B. et al. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG, *IEEE Transactions on Biomedical Engineering* 47(9):1185-1194, 2000.
- [5] Goldberger, A.L. et al. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals, *Circulation* 101(23):e215-e220, 2000.
- [6] William, O.T., Benbadis, S.R., Husain, A.M., Kaplan, P.W. *Handbook of EEG interpretation*, 2014, Demos Medical Publishing.
- [7] Rechtschaffen, A., Kales, A. *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*, 1968, US Government Printing Office, US Public Health Service.
- [8] Berthomier, C. et al. Automatic analysis of single-channel sleep EEG: validation in healthy individuals, *Sleep* 30(11):1587, 2007.
- [9] Landis, J.R., Koch, G.G. The measurement of observer agreement for categorical data, *Biometrics* 33(1):159-174, 1977.
- [10] Spiegel, M.R. *Schaum’s outline of theory and problems of Fourier analysis: with applications to boundary value problems*, 1994, McGraw-Hill.
- [11] Strang, G. Wavelet transforms versus Fourier transforms, *Bulletin of the American Mathematical Society* 28(2):288-305, 1993.
- [12] Roweis, S.T., Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding, *Science* 290(5500):2323-2326, 2000.
- [13] Tenenbaum, J.B., Silva, V.D., Langford, J.C. A global geometric framework for nonlinear dimensionality reduction, *Science* 290(5500):2319-2323, 2000.
- [14] Seung, H.S., Lee, D.D. The manifold ways of perception, *Science*, 290(5500):2268-2269, 2000.
- [15] Makeig, S. et al. *Matlab toolbox for analysis of electrophysiological data*, 1997.