An Open-source Toolbox For Standardized Use Of PhysioNet Sleep EDF Expanded Database

Syed Anas Imtiaz and Esther Rodriguez-Villegas

Abstract— **PhysioNet Sleep EDF database has been the most popular source of data used for developing and testing many automatic sleep staging algorithms. However, the recordings from this database has been used in an inconsistent fashion. For example, arbitrary selection of start and end times from long term recordings, data-hypnogram mismatches, different performance metrics and hypnogram conversion from R&K to AASM. All these differences result in different data sections and performance metrics being used by researchers thereby making any direct comparison between algorithms very difficult. Recently, a superset of this database has been made available on PhysioNet, known as the Sleep EDF Expanded Database which includes 61 recordings. This provides an opportunity to standardize the way in which signals from this database should be used. With this goal in mind, we present in this paper a toolbox for automatically downloading and extracting recordings from the Sleep EDF Expanded database and converting them to a suitable format for use in MATLAB. This toolbox contains functions for selecting appropriate data for sleep analysis (based on our previous recommendations for sleep staging), hypnogram conversion and computation of performance metrics. Its use makes it simpler to start using the new sleep database and also provides a foundation for much-needed standarization in this research field.**

I. INTRODUCTION

Human sleep is divided in to multiple stages that manifest in a cyclical fashion throughout the night. Including wake and movement, there are seven stages of sleep according to R&K [1] classification and five according to the more recent AASM [2] classification. A typical sleep study involves monitoring of neural, eye movement and muscle activity by recording multiple EEG (electroencephalogram), EOG (electrooculogram) and EMG (electromyogram) signals for the duration of sleep. These signals are then segmented in to epochs of 30 second and are subsequently scored by human experts, assigning a sleep stage to each epoch. To make sleep medicine more accessible, reduce analysis time and lower the financial burden, it is desirable to have an automated system that mimics the human expert and classifies the recordings based on established rules. These motivations has led to a steady growth in different algorithms for automatic scoring of sleep stages being proposed by various research groups. These work by extracting a wide range of characteristic features from the signals and classifying them in using different methods including decision trees [3], [4], artificial neural networks [5], [6], support vector machines [7] and many others. The performance of these algorithms are usually evaluated using data that has been acquired as part of their research work or by using publicly available sleep databases. The advantage of using the latter approach is that it not only helps to show a method's own detection performance but also makes it possible to compare it with other existing methods that have been tested using the same database.

PhysioNet Sleep EDF database [8], consisting of eight recordings, has been the most popular publicly available sleep database used by multiple automatic sleep staging algorithms for development. One of the problems with the use of this database has been the lack of consistency in which data is extracted. This is because there are two kind of recordings in the database: one set containing only overnight recordings while the other consisting of recordings over a period of 24 hours. Different criteria have been used by research groups to establish the sleep start and end times for extracting the sleep part from the 24 hour recordings. This results in different sections of data being used. Together with varying performance metrics, this makes the comparison between various algorithms very difficult.

More recently, the Sleep EDF database has been made deprecated and a much larger superset of this database, known as the PhysioNet Sleep EDF Expanded database [9] (also referred to as Sleep EDFx database), has been made available which includes 61 well annotated sleep recordings. Some other freely available databases have also been made available in the recent past including [10] and [11] with 200 and 20 recordings respectively. It is still early days in the use of the new EDF Expanded database. Results of analysis using this database have slowly started coming in with only four papers published so far [12]–[15] using this database. However, with the existing popularity of PhysioNet it is likely that EDF Expanded database will remain very popular in the coming years.

In our earlier work [16] we proposed a set of recommendations to minimize the aforementioned problems and encouraged the use of public databases amongst sleep researchers. Of course, it is not possible to go back and try to standardize the practices in already published work. However, the availability of a new database, which is potentially going to be widely used, presents a unique opportunity to establish a set of rules for its usage, particularly in the context of automatic sleep scoring. In this paper, we build upon the previously proposed recommendations, by providing their implementation and making it freely available as an open source toolbox for MATLAB [17]. This will make the sleep data more acces-

S. A. Imtiaz and E. Rodriguez-Villegas are with the Circuits and Systems Group, Electrical and Electronic Engineering Department, Imperial College London, United Kingdom. Email: ({anas.imtiaz,e.rodriguez}@imperial.ac.uk).

The research leading to these results has received funding from the European Research Council under the European Community's 7th Framework Programme (FP7/2007-2013) / ERC grant agreement no. 239749.

sible to researchers starting in this area, result in consistent data being used and the final algorithm performances also being uniformly assessed across different research groups. This toolbox is available at `www.imperial.ac.uk/rodriguez-villegas-lab/research`. Section II of this paper describes the functions for easily downloading and saving the data and hypnogram files. Section III explains how the start and end times for data are established, the functions to load data recordings and to report the performance metrics of an algorithm. Finally, Section IV discusses some further advantages of this approach and potential improvements for future development.

II. Getting the Data

A. Downloading PhysioNet data

The first, and often the most tedious, step in using the Sleep EDF Expanded database is to download each of the 61 recordings separately from the PhysioNet website. These data files are in EDF format [18] and require conversion to a suitable format before being used for any further signal processing. To save time and simplify this initial step, all of the EDF data files can be downloaded from the PhysioNet website by executing a single function, shown below, and optionally providing a location (`download_dir`) where the files will be saved. The files are downloaded in newly created directories, within this location, having the same name as the recording. The function returns a `saved_file` list with the full path of each downloaded file and its corresponding `status` to indicate whether it has been fetched successfully.

```
[saved_file,status]=downloadEDFxData(
   download_dir)
```

B. Conversion to Matlab

Once the EDF data files are successfully downloaded they need to be converted to a format that is readable in Matlab environment. This is achieved using the `convertEDFxToMat()` function that takes the full path of a `saved_file` and its download `status` as arguments.

```
convertEDFxToMat(saved_file, status)
```

The conversion of EDF files requires the use of EEGLAB, an open source toolbox for EEG analysis [19]. An error will be shown if EEGLAB is not found on the user's system. The resulting *.mat* data files are stored in a folder named *matlab* under the test directory. Additionally, the list of channels for which data are available is also saved in the same folder.

C. Downloading and processing annotations

The next step is downloading the hypnogram annotations for each recording. This is the file that consists of sleep stage score corresponding to each epoch of a test case. Multiple methods are available on PhysioNet to get the hypnogram annotations and we used the files as obtained from PhysioBank ATM due to easier processing. This file has some other important information as well including the start time of the recording, the sample number in data to be considered as the first valid sample (with reference to the start time) and the duration (in seconds) for which each sleep stage has been classified until the end of recording. An example section of this file is shown in Fig. 1

In this annotation, the start time is stamped as `16:13:00.000` with the sample number being `0` at that time. It also shows that data has been classified as `Sleep_stage_W` for `30630` seconds (until time `00:43:30.000`) and then as `Sleep_stage_1` for `120` seconds followed by `Sleep_stage_2` for `390` seconds. Considering an epoch size of 30 seconds, it can be seen that from the starting point 1021 epochs are classified as Wake followed by 4 epochs as Stage 1 and 13 epochs as Stage 2. This is used later to create a vector in Matlab for the hypnogram where each value in the vector represents the sleep stage assigned to a 30-second epoch. The hypnogram annotations are downloaded using the `downloadEDFxAnnotations()` function. It fetches not only the annotation files but also extra information including the text files with *lights off* time so that they do not have to be read off the spreadsheet. Files with *end time* are also provided to follow a consistent convention for data end times where these are not available (explained in Section III).

```
downloadEDFxAnnotations()
```

Once downloaded, the `hyp_file` is then processed using the function below which returns the `hypnogram` as a one dimensional Matlab vector with the same code for each sleep stage as the original on PhysioNet.

```
hypnogram = processEDFxHypnogram(hyp_file)
```

The process of initial downloading of source files and conversion to compatible data files needs to be performed only once and can together be automated with the `initialSetup` function which calls all the functions described above sequentially.

```
initialSetupEDFx(download_dir)
```

III. Using the Data

The downloaded data files can be easily opened as a normal file in Matlab. However, recordings have different start and end times that do not necessarily correspond to the hypnogram scoring times. Further, the *lights off* time may also have a positive or negative offset from the recording start time. This section explains how data and hypnogram files are loaded using the multiple time annotation files and by following the recommendations in [16].

1) Start and end times: It was recommended in [16] to use the annotated *lights off* and *lights on* times as start and end times respectively. Where these are not available then the time from 15 minutes before the first scored sleep epoch and 15 minutes after the last scored sleep epoch should be used as start and end times.

```
    Time        Date          Sample #   Type   Sub Chan  Num       Aux
[16:13:00.000 24/04/1989]         0       "      0    0    0   ## time resolution: 100
[16:13:00.000 24/04/1989]         0       "      0    0    0   Sleep_stage_W duration: 30630
[00:43:30.000 25/04/1989]   3063000       "      0    0    0   Sleep_stage_1 duration: 120
[00:45:30.000 25/04/1989]   3075000       "      0    0    0   Sleep_stage_2 duration: 390
```

Fig. 1: *A example section of hypnogram annotations from the recording SC4001E0*

For all the subjects in the EDF Expanded database, the *lights off* times are recorded in two different spreadsheets available on the PhysioNet website. From the hypnogram annotations, the *recording start time* is also known, which may be different from the *lights off* time. The annotations file also has the sample number from which hypnograms have been scored so the *hypnogram start time* can be deduced from this. For example in case of recording *ST7212J0*, the data recording starts at `24:43:00`, hypnogram marking starts at `23:43:30` while *lights off time* is `23:44:00`. As a result there are three different times: *recording start time*, *hypnogram start time* and *lights off time*. But the question is which of these should serve as the time from which data is to be read. The *hypnogram start time* will be either the same as *recording start time* or it will be after this time (since there can be no hypnogram without data). So if the *lights off* time comes after the *hypnogram start time*, then it is used as the start time otherwise the *hypnogram starting time* is used as the start time from which data is to be read.

The end time for reading each data file would ideally be the *lights on* time but these are not provided in the EDF Expanded database and hence need to be deduced. The method will be different for *SC* and *ST* subjects since the former consists of full day recordings of nearly 24 hours while the latter includes overnight sleep recording. For *SC* subjects, end time is taken as the time 15 minutes after the last scored sleep epoch (these have already been saved in a text file are are downloaded with the hypnogram). For *ST* subjects, end time is considered the same as the *recording stop time* in most cases unless the hypnogram scoring ends before the data recording end time. For the same example of *ST7212J0*, data recording stops at `08:30:00` but the hypnogram scoring ends at `08:08:30`. For such cases, the hypnogram end time is taken as the time up to which data is to be extracted. The hypnogram end times for *ST* subjects, extracted from the main annotation files, and the time 15-minute after the last scored sleep epoch for *SC* subjects are stored in the *lights on time* text file.

The time offsets are then calculated such that data and hypnogram files are both read starting from this new start time and finish at the new end time. The number of epochs between the new start and end times is also calculated.

All these steps of determining the time offsets and fully mapped data and hypnogram files are performed within the `loadEDFx` function using which any particular recording from the database can be easily loaded by providing in the path of the data directory and the `classification_mode` (AASM or RK) to use for hypnogram.

An example of loading data for case *SC4001E0* is shown in the code listing below.

```
loadEDFx('SC4001E0',classification_mode)
```

The result is a container with key-value pairs for each channel of data loaded, an array with a list of channels available to use, hypnogram, total number of 30-second epochs, sampling frequency of data and the start and end times between which data is read.

The hypnogram supplied with the PhysioNet EDF Expanded database uses R&K rules of sleep stage classification. In our earlier work, we proposed a method of approximate conversion from R&K to AASM scoring (shown in Table I). The `classification_mode` switch can be set to `AASM` if this conversion is required.

TABLE I: *Conversion from R&K to AASM classification*

R&K	S1	S2	S3	S4	REM	Wake	MT
AASM	N1	N2	N3		REM	Wake	

2) Viewing the data with hypnogram: After the files for a particular recording have been loaded, any signal from it can be viewed as a plot together with its hypnogram as a function of clock time. This is achieved using the `viewEDFxSignals()` function by providing the `signal` to be plotted and its corresponding `hypnogram` as arguments. Additionally, the function also requires `ref_times`, which is the reference start and end time of the recording and `f_samp`, which is the sampling frequency of the test. Both these are obtained from the `loadEDFx()` function explained earlier. A subsection of a signal may also be plotted between an arbitrary `start_time` and `end_time` provided these are between the *lights off* and *lights on* times of the recording.

```
viewEDFxSignals( signal, ref_times, ...
   start_time, end_time, f_samp, hypnogram )
```

A. Performance/Results

The performance of any new automatic sleep staging method can be evaluated by comparing the hypnogram generated by the new method against the reference hypnogram. Often, the performance metrics computed vary between different research groups making direct comparisons difficult. To have a uniform method to evaluate performances of different algorithms, we proposed the use of overall accuracy, sensitivity and selectivity of each sleep stage along with the confusion matrix of the epochs correctly and incorrectly classified. These metrics are shown in the equations below [16].

$$Accuracy = \frac{no.\ of\ true\ detections}{total\ no.\ of\ epochs} \quad (1)$$

$$Sensitivity = \frac{no.\ of\ true\ detections\ in\ stage\ X}{no.\ of\ reference\ epochs\ in\ stage\ X} \quad (2)$$

$$Selectivity = \frac{no.\ of\ true\ detections\ in\ stage\ X}{no.\ of\ all\ detections\ in\ stage\ X} \quad (3)$$

The toolbox includes a function to facilitate the computation of these metrics. It requires the hypnogram to be saved in the same format as the reference (i.e. as a vector of characters for each sleep stage) and is used as follows.

```
computeEDFxPerformance( test_hypnogram, ...
    ref_hypnogram, classification_mode )
```

The function takes in the new `test_hypnogram`, original `ref_hypnogram` and the `classification_mode` as its three arguments and prints out the overall accuracy and confusion matrix of the new algorithm as well as the sensitivity and selectivity obtained in each sleep stage separately. The use of this function can be helpful to compare different algorithms by employing a consistent method for performance analysis.

IV. Discussion & Conclusion

The Sleep EDF Expanded Database is still relatively new which provides an opportunity for sleep researchers to set some consistent rules for getting, using and analyzing the results from this database. The toolbox described in this paper aims to make it easy for researchers to adhere to such rules. The functions listed here are by no means exhaustive and there are many aspects in sleep research that require standardization [20]. However, the toolbox does provide a good starting point for, at least, using the signals from a popular database in a uniform manner. By making the code open source, and available on GitHub [21], it is hoped that other researchers could contribute to it by adding more functionality to the toolbox.

The use of this toolbox will ensure that the same segment of data is being analyzed by different researchers. It also helps in standardizing the way this data is sliced and the performance metrics are reported. This makes the comparison of various algorithms fair and prevents the common problems associated with the usage of previous PhysioNet Sleep EDF database. Further, the toolbox also saves time by providing quick and easy access to the PhysioNet Sleep EDFx database. This, in turn, means more time spent on data analysis rather than getting and preparing the data for use. It is important to adopt standard guidelines while the EDFx database is still in its infancy to prevent a repeat of earlier problems where different sections of data get used by different researchers together with varying performance metrics making their comparison almost impossible. We believe this toolbox provides a platform for much-needed standardization of sleep data usage, which in turn will also help advance the development of algorithms for automatic sleep staging.

References

[1] A. Rechtschaffen and A. Kales, Eds., *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. Washington D.C.: Public Health Service, U.S. Government Printing Office, 1968.

[2] C. Iber, S. Ancoli-Israel, A. Chesson, and S. Quan, Eds., *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications*. Westchester, IL: American Academy of Sleep Medicine, 2007.

[3] S.-F. Liang, C.-E. Kuo, Y.-H. Hu, and Y.-S. Cheng, “A rule-based automatic sleep staging method,” *J. Neurosci. Methods*, vol. 205, no. 1, pp. 169–76, 2012.

[4] M. Hanaoka, M. Kobayashi, and H. Yamazaki, “Automated sleep stage scoring by decision tree learning,” in *IEEE EMBC*, Chicago, July 2000.

[5] M. Ronzhina, O. Janousek, J. Kolarova, M. Novakova, P. Honzik, and I. Provaznik, “Sleep scoring using artificial neural networks,” *Sleep Med. Rev.*, vol. 16, no. 3, pp. 251–63, 2012.

[6] S. Charbonnier, L. Zoubek, S. Lesecq, and F. Chapotot, “Self-evaluated automatic classifier as a decision-support tool for sleep/wake staging,” *Comput. Biol. Med.*, vol. 41, no. 6, pp. 380–9, 2011.

[7] S. Khalighi, T. Sousa, D. Oliveira, G. Pires, and U. Nunes, “Efficient feature selection for sleep staging based on maximal overlap discrete wavelet transform and svm,” in *IEEE EMBC*, Boston, September 2011.

[8] PhysioNet. (2013) Sleep-edf database. [Online]. Available: http://www.physionet.org/physiobank/database/sleep-edf/.

[9] PhysioNet. (2014) Sleep-edf database [expanded]. [Online]. Available: http://www.physionet.org/physiobank/database/sleep-edfx/.

[10] C. O'Reilly, N. Gosselin, J. Carrier, and T. Nielsen, “Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research,” *J. Sleep Res.*, vol. 23, no. 6, pp. 628–35, 2014.

[11] University of MONS - TCTS Laboratory. (2013) The DREAMS Subjects Database. [Online]. Available: http://www.tcts.fpms.ac.be/~devuyst/Databases/DatabaseSubjects/.

[12] F. Yaghouby, P. Modur, and S. Sunderam, “Naive scoring of human sleep based on a hidden markov model of the electroencephalogram,” in *IEEE EMBC*, Chicago, August 2014.

[13] K. Aboalayon, H. Ocbagabir, and M. Faezipour, “Efficient sleep stage classification based on eeg signals,” in *IEEE LISAT*, New York, May 2014.

[14] T. Sanders, M. McCurry, and M. Clements, “Sleep stage classification with cross frequency coupling,” in *IEEE EMBC*, Chicago, August 2014.

[15] J. L. Rodriguez-Sotelo, A. Osorio-Forero, A. Jimenez-Rodriguez, D. Cuesta-Frau, E. Cirugeda-Roldan, and D. Peluffo, “Automatic sleep stages classification using eeg entropy features and unsupervised pattern analysis techniques,” *Entropy*, vol. 16, no. 12, pp. 6573–6589, 2014.

[16] S. Imtiaz and E. Rodriguez-Villegas, “Recommendations for performance assessment of automatic sleep staging algorithms,” in *IEEE EMBC*, Chicago, August 2014.

[17] MathWorks. (2015) Matlab and simulink for technical computing. [Online]. Available: http://www.mathworks.com/

[18] European Data Format. (2013) Home page. [Online]. Available: http://www.edfplus.info/.

[19] A. Delorme and S. Makeig, “Eeglab: an open source toolbox for analysis of single-trial eeg dynamics,” *J. Neurosci. Methods*, vol. 134, no. 1, pp. 9–21, 2004.

[20] V. Mihajlovic, B. Grundlehner, R. Vullers, and J. Penders, “Wearable, wireless eeg solutions in daily life applications: What are we missing?” *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 6–21, 2014.

[21] Anas Imtiaz. (2015) Sleep EDFx Toolbox - GitHub. [Online]. Available: https://github.com/anas-imtiaz/sleep-edfx-toolbox