



ELSEVIER

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/mex

Method Article

Optimization of real-time analysis of sleep-wake cycle in mice



Stephen Thankachan^a, Andrei Gerashchenko^b, Ksenia V Kastanenka^c,
Brian J Bacsikai^c, Dmitry Gerashchenko^{a,*}

^aHarvard Medical School / Veterans Affairs Boston Healthcare System, West Roxbury, MA 02132, USA

^bNeurotargeting Systems, Inc., Chestnut Hill, MA 02467, USA

^cDepartment of Neurology, MassGeneral Institute of Neurodegenerative Diseases, Massachusetts General Hospital and Harvard Medical School, Charlestown, MA 02129, USA

ABSTRACT

Studying the biology of sleep requires accurate and efficient assessment of the sleep stages. However, analysis of sleep-wake cycles in mice and other laboratory animals remains a time-consuming and laborious process. In this study, we developed a Python script and a process for the streamlined analysis of sleep data that includes real-time processing of electroencephalogram (EEG) and electromyogram (EMG) signals that is compatible with commercial sleep-recording software that supports user datagram protocol (UDP) communication. The process consists of EEG/EMG data acquisition, automated threshold calculation for real-time determination of sleep stages, sleep staging and EEG power spectrum analysis. It also allows data storage in the format that facilitates further analysis of the sleep pattern in mice. The described method is aimed at increasing efficiency of sleep stage scoring and analysis in mice thus facilitating sleep research.

- A process of EEG/EMG recording and streamline analysis of sleep-wake cycle in real time in mice.
- The compatibility with commercial sleep-recording software that can generate a UDP stream.
- The capability of further analysis of recorded data by an open-source software.

© 2022 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license
(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

ARTICLE INFO

Method name: Real-time sleep recording and analysis

Keywords: Sleep scoring, Vigilance states, EEG, EMG, Real-time analysis

Article history: Received 25 March 2022; Accepted 13 July 2022; Available online 8 August 2022

* Corresponding author.

E-mail address: dmitry_gerashchenko@hms.harvard.edu (D. Gerashchenko).

Specifications table

Subject Area:	
More specific subject area:	<i>Sleep research</i>
Method name:	<i>Real-time sleep recording and analysis</i>
Name and reference of original method:	<i>N/A</i>
Resource availability:	<i>N/A</i>

*Method details

Introduction

Accurate classification of behavioral states in mice is important to determine sleep structure in preclinical research. During recent years, many diverse algorithmic approaches have been developed to automate the sleep stage scoring process. For example, the deep learning algorithm for assessing sleep stages by the multimodal with fine-tuning approach achieved excellent performance in mice with an accuracy rate of 95.07% (% of epochs that were scored identically by automated and manual approaches) [1]. Alternatively, MC-SleepNet combines two types of deep neural networks that can automatically score sleep stages with an accuracy of 96.6% [2], whereas mixture z-scoring, a machine learning algorithm used in AccuSleep, demonstrated an accuracy of 96.8% [3]. A very high accuracy of determining sleep stages was also achieved with automated sleep stage scoring using a k-nearest neighbors classifier [4], a supervised deep convolutional neural network called WaveSleepNet [5], a generative adversarial network-assisted image-based sleep staging program named GI-SleepNet [6], and a neural network architecture that is able to distinguish between the three main stages Wake, REM and NREM as well as the infrequent stages pre-REM and artifact [7]. Although the accuracy of automated sleep staging is excellent, analysis of sleep-wake cycles in mice remains a time-consuming and laborious process. One of the challenges is that various data acquisition software save EEG and EMG traces in different formats that subsequently require conversion for the analysis. Another challenge arises from the need to analyze the EEG power spectrum, which requires extensive computer processing and considerable time commitment. To overcome these problems, we have developed a process for streamlining the sleep data analysis that includes processing of EEG and EMG signals. The process consists of EEG/EMG data acquisition via a UDP stream, automated calculation of thresholds for real-time determination of sleep stages, online sleep staging and EEG power spectrum analysis, as well as saving the data in the format that facilitates the further analysis of the sleep pattern in mice. Additionally, Transistor-Transistor Logic (TTL) pulses can be generated during particular stages of sleep to allow closed-loop stimulation, sleep deprivation, and other behavioral interventions in real time.

Materials and methods

Animals

Adult C57BL/6J mice (n=8, stock #000664), 3 to 6 months old, were obtained from the Jackson Laboratory. The mice were housed at 70°F - 74°F with a 12-h light/dark cycle (7:00 AM–7:00 PM) and unlimited food and water. All procedures were conducted in accordance with Veterans Affairs (VA) & National Institute of Health (NIH) guidelines and were approved by the VA Boston Healthcare System Institutional Animal Care and Use Committee. Adherence to transparent reporting was followed through the updated Animal Research: Reporting of In Vivo Experiments (ARRIVE 2.0) guidelines.

Surgical techniques

Mice were deeply anesthetized with isoflurane (1–3%) and fixed to a stereotaxic frame. Skin over the skull was incised and retracted to expose the skull. An electroencephalogram (EEG) electrode was screwed into the skull above frontal cortex (AP = 1.7; ML = ±1), and a reference electrode was screwed

into the skull above the cerebellum. Electromyogram (EMG) electrodes were placed in the nuchal muscle. The electrodes were then connected to a plug (PlasticsOne) and fixed to the skull using dental cement. Following the mounting of the electrodes on the skull with dental cement, the scalp incision was sutured closed. The mice were injected with meloxicam and allowed to recover.

In vivo EEG/EMG recordings

Following at least one week of postsurgical recovery and two days of adaptation to the recording chamber, mice were tethered to either the F20-EET wireless transmitter (DSI) or 3-channel wireless EEG/EMG system (Pinnacle Technology) on a swivel mounted system (Neurotargeting Systems) allowing them to move freely in their cage as previously described [8]. The data were transferred to either a receiver plate (DSI) or a USB dongle receiver (Pinnacle Technology). EEG and EMG signals were recorded via telemetry using Dataquest ART 4.1 software (DSI) at a frequency of 500 Hz or with Sirenia® acquisition software (Pinnacle Technology) at a frequency of 512 Hz. The data was either acquired via UDP communication and processed by the Python script or saved for subsequent analysis.

Sleep data analysis

The data files were imported to AccuSleep software for further manual scoring of sleep states in 5 second epochs during the 24-hour recording. The data transfer could occur after the recordings were completed or while the data acquisition was still going on. Vigilance states—including NREM sleep, REM sleep, and waking—were determined in 1 hour time blocks as previously described [8] as one of the following: (1) Wake; active behavior accompanied by desynchronized EEG (low in amplitude), and tonic/phasic motor activity evident in the EMG signal; (2) NREM sleep; more synchronized EEG, higher in amplitude, with particularly notable power in the delta (0.5–4.0 Hz) band, and low motor activity (EMG); and (3) REM sleep; small amplitude EEG, particularly notable power in the theta (4–9 Hz) band, and phasic motor activity (EMG). Compared to wake, EEG power during REM sleep was significantly reduced in delta frequencies (0.5–4.0 Hz) and increased in the range of theta activity (4–9 Hz). One hour of the EEG/EMG data was used to calculate thresholds for the Python script and generate a calibration file for the AccuSleep script. The thresholds and calibration files were then used to perform the automated sleep scoring. Accuracy, sensitivity, specificity and F1 score were calculated as previously described [9;10].

Statistical analysis

Comparisons of the accuracy, sensitivity, specificity, and F1 score between different time intervals was done using one-way repeated-measures analysis of variance (ANOVA). Results are reported as mean \pm SEM.

Results

Description of the system for the real-time streamline analysis of sleep data in mice

Acquisition of EEG and EMG

Commercial data acquisition software often require saving the EEG/EMG data in specialized formats. Most of the software allow data conversion into alternative formats, such as TXT or EDF, but the conversion requires significant amount of time, especially for recordings of long durations. To allow data storage in the desired format during the recording, we chose to acquire the data via the UDP stream. Some commercial programs (Sirenia by Pinnacle Technology, the Stellar system by TSE) allow generation of such data streams, and it is anticipated that other companies will include similar options in the near future. Our code allows for continuous data collection via the UDP and real-time sleep staging (Fig. 1 and Video 1). It also allows saving the data as .mat file for offline sleep staging (Video 2) and further analysis using MATLAB scripts.

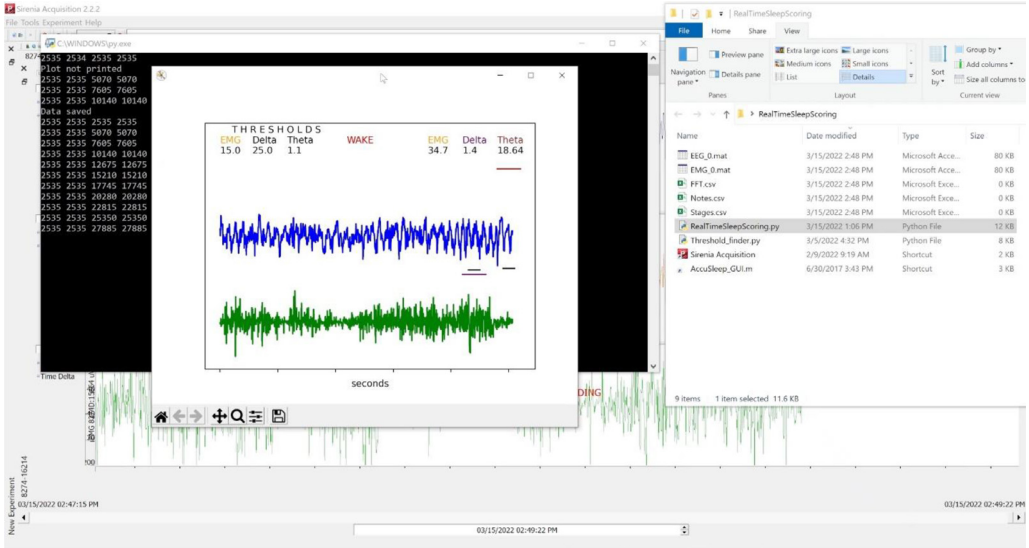


Fig. 1. Graphic interface of EEG and EMG recordings. EEG and EMG signals are sampled at 512 Hz by Sirenia Acquisition software (Pinnacle Technology) and transmitted using user datagram protocol (UDP). Five-second intervals of the EEG and EMG data are processed by the Python script and drawn in a separate window. Selected thresholds are shown in the upper left corner of the window. The EEG, Delta and Theta values recorded during the 5-second interval are displayed in the upper right corner and also visualized as bars on the right side of the window. The scored epoch (WAKE, NREM or REM) is shown at the top of the window.

Sleep-scoring algorithm

Sleep-scoring algorithms for determining sleep-wake state using a computer-based analysis of EEG and EMG data are typically based on user defined logic rules/thresholds [11–15] or a machine learning approach [1-5;7;9;16]. The machine learning approaches generally result in a better accuracy of classifying the sleep stages, but the threshold-based approaches offer a greater ease of use and simplify making adjustments for each individual mouse. We decided to classify the sleep stages in real time based on 3 thresholds, making the program convenient to use. Because the data is saved in a .mat file, it can be analyzed directly by AccuSleep, a MATLAB script that allows determining sleep-wake states with high accuracy [3].

A sleep-scoring algorithm that allows the real-time scoring of sleep-wake state in mice is based on the classification procedure previously used in other research laboratories. [12–14]. As a first step, we assigned the time interval (epoch) for determining behavioral states. We selected the epoch duration of 5 seconds, but this could be adjusted according to the need of the research study. To process the EMG signal, we then (1) removed the mean value from the signal, (2) band pass-filtered (40–90 Hz) the signal, (3) applied 4th order Butterworth filters, and (4) calculated the root-mean-square (RMS). The power spectral density (PSD) of the EEG was calculated using the Welch method in the following frequency ranges: (δ : 0.5–5 Hz; θ : 6–9 Hz; α : 10–15 Hz; η : 16–22.75 Hz and β : 23–31.75 Hz). The delta band and theta band were specified according to the following formulas: $(\delta \times \alpha)/(\eta \times \beta)$ and $(\theta \times \theta)/(\delta \times \alpha)$ [12–14]. Thus, we obtained three values (EMG RMS, EEG delta band and EEG theta band) that were subsequently used to discriminate between wakefulness, NREM sleep and REM sleep. The determination of the wake or sleep stage was done according to the following logic procedure:

STEP 1: EMG RMS value is greater than EMG threshold. → WAKEFULNESS

STEP 2: EEG delta band $(\delta \times \alpha)/(\eta \times \beta)$ is lower than delta threshold and three preceding epochs were scored as WAKEFULNESS. → WAKEFULNESS

STEP 3: EEG delta band $(\delta \times \alpha)/(\eta \times \beta)$ is greater than delta threshold. → NREM sleep

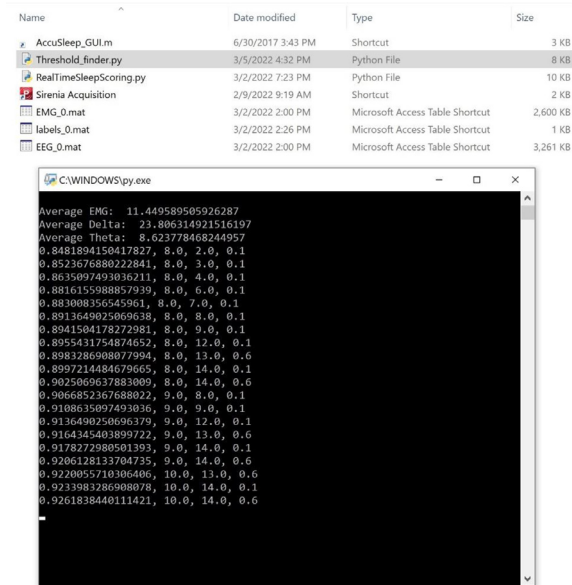


Fig. 2. Automated selection of thresholds. An hour of recorded EEG and EMG data (EEG_0.mat and EEG_0.mat) and a file containing epochs scored in AccuSleep (labels_0.mat) are processed by the Python script to select the thresholds allowing the highest correspondence between the manual and automated scoring. The accuracy value and corresponding EMG, Delta and Theta thresholds are displayed on the screen.

STEP 4: EEG delta band $(\delta \times \alpha)/(\eta \times \beta)$ is greater than 80% of delta threshold and one preceding epoch was scored as NREM sleep. → NREM sleep

STEP 5: EEG delta band $(\delta \times \alpha)/(\eta \times \beta)$ is greater than 60% of delta threshold and two preceding epochs were scored as NREM sleep. → NREM sleep

STEP 5: EEG delta band $(\delta \times \alpha)/(\eta \times \beta)$ is greater than 40% of delta threshold and three preceding epochs were scored as NREM sleep. → NREM sleep

STEP 6: EEG theta band $(\theta \times \theta)/(\delta \times \alpha)$ is greater than theta threshold and any of two preceding epochs were not scored as WAKEFULNESS. → REM sleep

STEP 7: EEG theta band $(\theta \times \theta)/(\delta \times \alpha)$ is greater than 80% of theta threshold and the preceding epoch was scored as REM sleep. → REM sleep

STEP 8: EEG theta band $(\theta \times \theta)/(\delta \times \alpha)$ is greater than 60% of theta threshold and two preceding epochs were scored as REM sleep. → REM sleep

STEP 9: EEG theta band $(\theta \times \theta)/(\delta \times \alpha)$ is greater than 40% of theta threshold and three preceding epochs were scored as REM sleep. → REM sleep

STEP 10: None of the above conditions were true. → WAKEFULNESS

Threshold selection

When thresholds are adjusted manually, it is difficult to reach a high accuracy of determining sleep-wake states unless a large number of threshold combinations is tested. To solve this problem, we wrote a Python script that allows us to test 10,000 or even 100,000 combinations automatically in under a minute. According to our procedure, we select an hour of EEG and EMG recordings saved as .mat files and score sleep in 5 second epochs manually using the AccuSleep script. Then, the files containing the EEG and EMG data as well as the scored file containing assigned sleep epochs are processed by the Python script (Fig. 2). The outcome of this procedure is 3 thresholds (EMG RMS, EEG delta band and EEG theta) selected based on the highest correspondence between epochs that were assigned identically by the Python script and manual scoring.

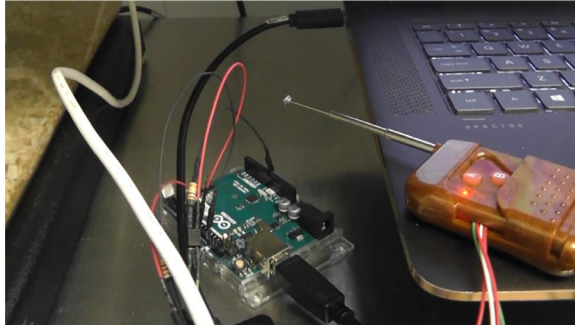


Fig. 3. Interaction with external devices via Transistor-Transistor Logic (TTL). TTL interface allows turning on and off a remote control during chosen behavioral states.

Determination of sleep-wake states in real time

As soon as the thresholds are determined as described in the prior paragraph, they are entered into the Python script, after which the sleep recordings are restarted (Fig. 1). The algorithm contains a graphical user interface for visualization of the EEG/EMG traces. The acquired data can be also visualized by the program through which the data stream is acquired. In this study, we used the Sirenia Acquisition program for this purpose. The assigned epochs are saved in a file named "Stages.csv", and the power spectrum calculated during each 5 second epoch at the 0.5 Hz resolution is saved in a file named "FFT.csv".

Closed-loop feedback capability

The Python script can output classification results in real-time to an Arduino device, which allows implementing a closed-loop feedback system of choice. The delay of initiating the feedback is 5 seconds due to the epoch window size. Additionally, the Arduino device is connected to a wireless remote controller that allows wireless control of various devices during specific states of sleep-wake cycle (Fig. 3).

Evaluation of the classification performance of the threshold-based method

The role of circadian time in threshold calculation

In our method, threshold values are calculated based on a 1-hour sample of EEG/EMG data. Because the thresholds are used to determine sleep stages of the subsequent recordings, it is important to know whether selecting the sample data at a different circadian time has an effect on the accuracy of the sleep scoring procedure. To test the effect of the circadian time, we selected 4 intervals during which we calculated thresholds (ZT3-4, ZT8-9, ZT15-16, and ZT20-21). Two of these 1-hour intervals were selected during the light-on period (3 hours after the light on or 3 hours before the light off), and another two 1-hour intervals were selected during the light off period (3 hours after the light off or 3 hours before the light on). REM sleep was not present in some 1-hour intervals recorded during the light off period. In such cases, we replaced these intervals with the subsequent hour. The thresholds were calculated for each of these hours as described in the paragraph 3.1.3 and then used to perform automated staging during the entire 24 hours of the recording. Then, the accuracy of sleep staging was calculated by comparing the results of the automated and manual scoring during the 24-hour period. Additionally, the entire 24-hour period was used to calculate the thresholds to determine the highest level of accuracy that could be achieved by this method.

As shown in Fig. 4, accuracy of sleep staging exceeded 90% for all four intervals. The highest accuracy was achieved when the data recorded during ZT3-4 were used to determine thresholds (91.2%). However, one-way ANOVA with repeated measures did not identify significant differences between these time intervals ($F(4, 28) = 2.649$, $P = 0.054$). The highest accuracy reached 92.3% when the entire 24-h period was used to calculate the thresholds, which is 1.1% better than using the only

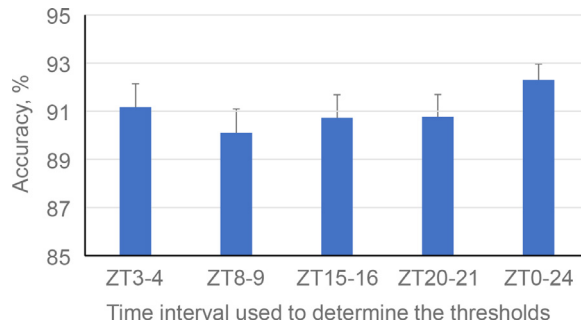


Fig. 4. Effect of the choice of the time interval used to determine thresholds on the accuracy of sleep staging. The accuracy was about 90% or more for all tested intervals.

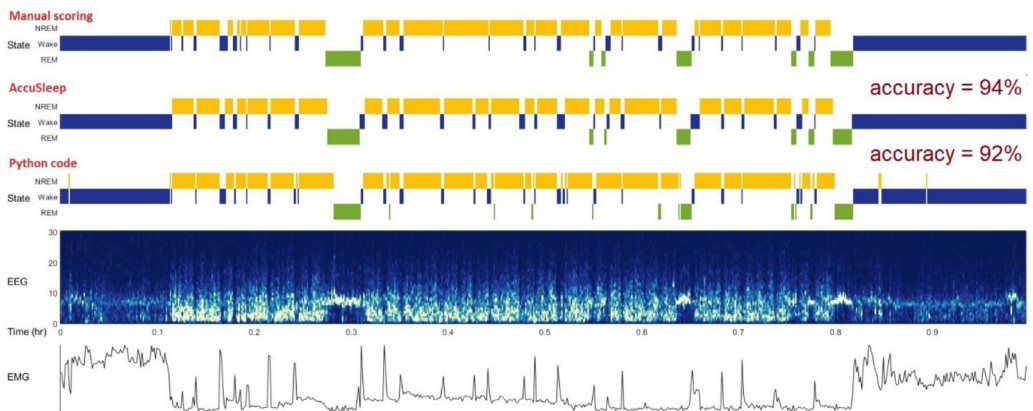


Fig. 5. A representative example of automated sleep scoring by the threshold-based Python script and AccuSleep script. One hour of the EEG/EMG data recorded during the ZT3-4 time interval was used to automatically determine the thresholds for the Python script and produce a calibration file for the AccuSleep script. One hour of EEG/EMG data recorded five hours later (ZT8-9) in the same mouse was then automatically scored by both scripts. A high correspondence between the manual scoring and automated scoring by both scripts was observed. The figure was produced using the AccuSleep script in MATLAB.

ZT3-4 interval to calculate the thresholds. An example of comparison between the manual scoring and automated scoring by AccuSleep and Python script is shown in Fig. 5.

Sensitivity, specificity and F1 score

In addition to assessing the overall accuracy of sleep staging, we used sensitivity, specificity and F1 score as additional metrics of classification performance for specific behavioral stages. In this analysis, we determined thresholds based on the ZT3-4 interval and then used them to assign sleep stages for the entire 24-hour recording. As shown in Table 1, a high level of these measures (90% or more) was obtained for wakefulness and NREM sleep. The level of sensitivity and F1 score were around 70% for REM sleep.

Discussion

A large number of highly accurate behavioral state classifiers for automatic sleep staging have been developed in recent years. Most common automated sleep scoring approaches are based on either

Table 1

Performance evaluation of wakefulness, NREM sleep, and REM sleep staging. The thresholds calculated during ZT3-4 in 8 mice were used to automatically stage sleep for 24 hours in the same mice. The 24-h data were scored manually and compared to the result of automated scoring. The highest sensitivity, specificity and F1 score were observed for wakefulness.

	Sensitivity	Specificity	F1 score
Wake	93.5 ± 1.1	92.4 ± 2.2	93.6 ± 1.0
NREM sleep	91.0 ± 2.4	92.7 ± 1.1	89.6 ± 1.1
REM sleep	68.3 ± 2.3	99.1 ± 0.2	73.3 ± 2.8

thresholds/logic rules [11–15] or machine learning algorithms [1–5;7;9;16]. These approaches typically use the power spectrum of the EEG signal in combination with EMG signal power to determine sleep stages, although they differ in how those characteristics are selected and combined to predict behavioral states. In this study, we also used EEG and EMG for sleep staging and achieved a high accuracy of behavioral stage detection based only on 3 thresholds and a few logic rules. The accuracy depended on the quality of EEG and EMG recordings, as it varied from 86% to 95%. The sleep staging worked particularly well for recognizing wakefulness and NREM sleep, so sensitivity, specificity, and F1 scores were over 90% for wakefulness and NREM sleep (Table 1). Sensitivity and F1 score were lower for REM sleep compared to other behavioral states (about 70%).

Although reaching high accuracy of sleep staging is important, it was not the main goal of our study. Instead, we focused on the major obstacles that sleep researchers face during data collection and analysis in mice. These obstacles include difficulties in efficient collection and processing of large EEG and EMG datasets that are often sampled at a high rate for weeks or even months. We rationalized that the efficiency of sleep analysis would be significantly improved if the collected data are processed in real time while they are collected on the computer. Thus, we wrote a Python script that allows calculating the PSD and sleep stage for each epoch at the time of data collection, which are then saved on the computer in separate files. Other difficulty in processing the data derives from the variability of formats in which the EEG and EMG data are saved. If the format of saved data cannot be read by a sleep-analysis software, a conversion to another format is needed. To avoid such a problem, we collected data via the UDP communications protocol and saved it in .mat files that can be directly read by the AccuSleep script. AccuSleep was chosen because it is an open source MATLAB script that demonstrated one of the highest accuracies of sleep staging in mice [3]. Additionally, it has a user-friendly interface and is simple to use (Fig. 5).

Another advantage of processing the data in real time is that it gives a possibility to perform various sleep-dependent behavioral interventions. Considering that our script demonstrated a high accuracy in detecting wakefulness, we anticipate that it will be particularly useful for automated sleep deprivation protocols. It could be also employed in closed-loop stimulation and other protocols that require performing various research manipulations (optogenetic, auditory or visual stimulation, etc.) specifically during wakefulness or sleep.

The presented method has limitations. One potential limitation derives from the use of UDP communication. Despite its high efficiency in transmitting data, such communication could potentially generate errors because UDP does not use an error control. Although we did not notice errors during this study, they may arise in certain situations. Other likely limitation may be reduced accuracy of sleep staging in case the EEG and EEG signals are noisy or are missing some data. In this study, we did not test the ability of our script to discriminate between sleep stages when the EEG and/or EEG contained many artifacts. If artifacts significantly impact the accuracy of sleep staging, the algorithm can be modified to remove most of them as it has been done in previous studies [14], or an error-correcting connection can be used.

In conclusion, we developed a Python script and a process for a streamline analysis of sleep-wake cycle in mice that is compatible with commercial sleep-recording software that can generate a UDP stream of EEG and EMG signals. The real-time data processing and the capability of further analysis of recorded data by an open-source software is likely to make this script a useful tool for the research community.

Data availability

All data generated or analyzed during this study is included in the published article. Datasets not included are available from the corresponding author upon reasonable request.

Declaration of competing interest

DG has family members employed by Neurotargeting Systems, and AG has affiliation with Neurotargeting Systems.

Acknowledgements

This work was supported by NIH NIA-AG061774. We are thankful to Neurotargeting Systems for providing us an apparatus for connecting with freely moving mice and writing the Python script (AG) for this study.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.mex.2022.101811](https://doi.org/10.1016/j.mex.2022.101811).

References

- [1] T. Yagi, K. Akada, C.T. Beuckmann, Y. Miura, K. Aoshima, N. Koyama, A deep learning algorithm for sleep stage scoring in mice based on a multimodal network with fine-tuning technique, *Neurosci. Res.* 173 (2021) 99–105.
- [2] M. Yamabe, K. Horie, H. Shiokawa, H. Funato, M. Yanagisawa, H. Kitagawa, MC-SleepNet: large-scale sleep stage scoring in mice by deep neural networks, *Sci. Rep* 9 (2019) 15793.
- [3] Z. Barger, C.G. Frye, D. Liu, Y. Dan, K.E. Bouchard, Robust, automated sleep scoring by a compact neural network with distributional shift correction, *PLoS ONE* 14 (2019) e0224642.
- [4] S. Morairty, T. Kiss, T.S. Kilduff, M. Schwartz, D. Volfson, D.L. Buhl, Automated sleep stage scoring using k-nearest neighbors classifier, *J. Open Source Softw.* 5 (2020) 2377.
- [5] D.M. Rapoport, K. Kam, I. Ayappa, A. Parekh, A.W. Varga, WaveSleepNet: an interpretable deep convolutional neural network for the continuous classification of mouse sleep and wake, *J. Neurosci. Methods* 360 (2021) 109224.
- [6] T. Gao, J. Li, Y. Watanabe, C. Hung, A. Yamanaka, K. Horie, M. Yanagisawa, M. Ohsawa, K. Kume, GI-SleepNet: a highly versatile image-based sleep classification using a deep learning algorithm, *Clocks. Sleep* 3 (2021) 581–597.
- [7] J.T.C. Schwabedal, N. Grieger, Y. Ritze, S. Wendel, S. Bialonski, Automated scoring of pre-REM sleep in mice with deep learning, *Sci. Rep* 11 (2021) 12245.
- [8] L. Gerashchenko, M.R. Zielinski, D. Gerashchenko, S.A. Karpova, A novel telemetric system to measure polysomnographic biopotentials in freely moving animals, *J. Neurosci. Methods* 216 (2013) 79–86.
- [9] D. Kumar, T. Tezuka, I. Koyanagi, S. Singh, M. Sakaguchi, T. Naoi, Real-time, automatic, open-source sleep stage classification system using single EEG for mice, *Sci. Rep* 11 (2021) 11151.
- [10] H. Sei, G.A. Sunagawa, Y. Urade, S. Shimba, H.R. Ueda, FASTER: an unsupervised fully automated sleep staging method for mice, *Genes Cells* 18 (2013) 502–518.
- [11] S. Izawa, S. Chowdhury, T. Miyazaki, Y. Mukai, D. Ono, R. Inoue, Y. Ohmura, H. Mizoguchi, K. Kimura, M. Yoshioka, A. Terao, T.S. Kilduff, A. Yamanaka, REM sleep-active MCH neurons are involved in forgetting hippocampus-dependent memories, *Science* 365 (2019) 1308–1313.
- [12] C.P. Romanowski, T. Fenzl, K. Honsberg, C. Flachskamm, A. Hoehne, E. Boll, M. Kimura, Fully automated sleep deprivation in mice as a tool in sleep research, *J. Neurosci. Methods* 166 (2007) 229–235.
- [13] J. Lee, R.P. Louis, R. Stephenson, Design and validation of a computer-based sleep-scoring algorithm, *J. Neurosci. Methods* 133 (2004) 71–80.
- [14] S. Polta, M. Kreuzer, C. Schuler, J. Gapp, T. Fenzl, E.F. Kochs, Sleep scoring made easy-Semi-automated sleep analysis software and manual rescoring tools for basic sleep research in mice, *MethodsX* 2 (2015) 232–240.
- [15] C.M. Walsh, B.A. Gross, V. Booth, A.A. Turakhia, G.R. Poe, G.A. Mashour, Open-source logic-based automated sleep scoring software using electrophysiological recordings in rats, *J. Neurosci. Methods* 184 (2009) 10–18.
- [16] M.B. Dash, J.G. Ellen, An artificial neural network for automated behavioral state classification in rats, *Peer J.* 9 (2021) e12127.