



# Information-Theoretical Analysis of EEG Microstate Sequences in Python

Frederic von Wegner<sup>1,2\*</sup> and Helmut Laufs<sup>2,3</sup>

<sup>1</sup> Epilepsy Center Rhein-Main, Goethe University Frankfurt, Frankfurt am Main, Germany, <sup>2</sup> Department of Neurology and Brain Imaging Center, Goethe University Frankfurt, Frankfurt am Main, Germany, <sup>3</sup> Department of Neurology, University Hospital Kiel, Kiel, Germany

We present an open-source Python package to compute information-theoretical quantities for electroencephalographic data. Electroencephalography (EEG) measures the electrical potential generated by the cerebral cortex and the set of spatial patterns projected by the brain's electrical potential on the scalp surface can be clustered into a set of representative maps called EEG microstates. Microstate time series are obtained by competitively fitting the microstate maps back into the EEG data set, i.e., by substituting the EEG data at a given time with the label of the microstate that has the highest similarity with the actual EEG topography. As microstate sequences consist of non-metric random variables, e.g., the letters A–D, we recently introduced information-theoretical measures to quantify these time series. In wakeful resting state EEG recordings, we found new characteristics of microstate sequences such as periodicities related to EEG frequency bands. The algorithms used are here provided as an open-source package and their use is explained in a tutorial style. The package is self-contained and the programming style is procedural, focusing on code intelligibility and easy portability. Using a sample EEG file, we demonstrate how to perform EEG microstate segmentation using the modified K-means approach, and how to compute and visualize the recently introduced information-theoretical tests and quantities. The time-lagged mutual information function is derived as a discrete symbolic alternative to the autocorrelation function for metric time series and confidence intervals are computed from Markov chain surrogate data. The software package provides an open-source extension to the existing implementations of the microstate transform and is specifically designed to analyze resting state EEG recordings.

## OPEN ACCESS

### Edited by:

Pedro Antonio Valdes-Sosa,  
Clinical Hospital of Chengdu Brain  
Science Institute, China

### Reviewed by:

Emilia Iannilli,  
National Center for Adaptive  
Neurotechnologies, United States  
Vasil Kolev,  
Institute of Neurobiology (BAS),  
Bulgaria

### \*Correspondence:

Frederic von Wegner  
vonWegner@med.uni-frankfurt.de

**Received:** 05 March 2018

**Accepted:** 11 May 2018

**Published:** 01 June 2018

### Citation:

von Wegner F and Laufs H (2018)  
Information-Theoretical Analysis of  
EEG Microstate Sequences in Python.  
*Front. Neuroinform.* 12:30.  
doi: 10.3389/fninf.2018.00030

**Keywords:** EEG microstates, information theory, entropy, mutual information, Markovianity, open-source

## 1. INTRODUCTION AND BACKGROUND

Electroencephalography (EEG) is a routine technique in neuroscientific research and the clinical sciences, used to measure electrical potentials generated by the cerebral cortex. It is a relatively low-cost and widely distributed diagnostic tool. The measured EEG signal records a superposition of excitatory and inhibitory postsynaptic potentials via electrodes located on the skull surface (Niedermeyer and da Silva, 2005). Among the data reduction techniques that have been employed to compress EEG recordings, the microstate algorithm is of special importance as it has been evaluated in a variety of experimental conditions (Lehmann et al., 1987; Wackermann et al., 1993; Pascual-Marqui et al., 1995). The microstate algorithm can be summarized as follows. Consider an

EEG data set that consists of  $n_t$  time samples from  $n_{ch}$  channels, or electrode locations. Then, each sample is an array of  $n_{ch}$  real numbers, each number representing the electrical potential at a specific location, and the whole array provides a discrete sampling of the continuous electrical field. An EEG data set can therefore be visualized as a time series of changing spatial patterns, often called maps. The microstate algorithm searches for a small set of spatial patterns that explain the maximum amount of the data's variance. Often, only four representative maps are needed to explain ca. 70% of the data variance and to capture neurobiologically relevant data features (Koenig et al., 2002; Brodbeck et al., 2012; Khanna et al., 2015; Kuhn et al., 2015). It could be shown that the computed microstates convey information about functional brain states during cognition (Milz et al., 2015), different vigilance states (Brodbeck et al., 2012; Kuhn et al., 2015), and disease (Koenig et al., 1999; Nishida et al., 2013). We here implement the commonly employed modified K-means algorithm, as introduced in Pascual-Marqui et al. (1995) and which has been used in many published studies, for instance in vision research (Antonova et al., 2015), studies on olfaction (Iannilli et al., 2013), and taste (Iannilli et al., 2017), and in a multi-center schizophrenia study (Lehmann et al., 2005), to name but a few.

The canonical K-means algorithm yields a cluster assignment that minimizes the sum of squared distances of all data points to their respective cluster centroids, i.e., to the arithmetic mean of all points currently assigned to that cluster. The algorithm proceeds stochastically, using a fixed number of clusters and initializing the cluster centroids with randomly selected data samples. In the case of EEG records, a data sample consists of an array of electrical potential values at a given time point, and the size of the array represents the number of EEG channels. In each iteration, the algorithm assigns each data sample to its closest cluster centroid, and then updates all clusters and their centroids taking into account the newly assigned samples. Modified K-means clustering for EEG microstates, as introduced in Pascual-Marqui et al. (1995), does not use the arithmetic mean of samples to represent the cluster, but the first principal component of the samples. Thus, the polarity of the EEG topography is ignored, leaving the overall symmetry of the potential topography as the feature to be clustered (Wackermann et al., 1993; Pascual-Marqui et al., 1995). The convergence criterion for the modified K-means algorithm is the relative error in the explained variance, as detailed further below.

In this context, two particular types of EEG experimental designs should be mentioned, resting state recordings on the one hand, and event-related potentials (ERP) on the other. We here focus on resting state recordings, in which the ongoing EEG in a task-free ("resting") condition is recorded in order to follow spontaneous changes in cortical activity. Resting state recordings have received considerable attention as they provide insight into functional brain networks that spontaneously activate and de-activate (Tagliazucchi et al., 2012). In ERP experiments, which we will not study further in this article, certain stimuli (acoustic, visual, cognitive tasks) are presented repetitively and the synchronously recorded EEG signal is analyzed in blocks. The start of each EEG data block is aligned with the stimulus

presentation times, and thus, EEG features time-locked to stimulus onset are extracted. Due to often low signal-to-noise ratios in single stimulus responses, the evoked EEG changes are usually averaged. We mention ERP experiments since the microstate approach has been applied to ERP data for a long time (Murray et al., 2008). The algorithms presented here can readily be applied to ERP data sets, however, we do not provide the functionalities for the necessary pre-processing, such as epoch splitting, averaging, and ERP component identification.

The microstate algorithm transforms an EEG data set into a sequence of microstate labels, according to maximum similarity between the candidate microstates and the actual EEG topography. Commonly, the microstate maps are labeled with the symbols A–D. The fact that the resulting time series consist of categorical variables severely limits the list of applicable time series methods. Frequently used linear characteristics, such as the autocorrelation function or the power spectral density for instance, cannot be computed as sum and product terms are not defined on the discrete set of states.

The most frequently used approach is the transition matrix method that summarizes microstate dynamics by a square matrix of first-order transition statistics, i.e., the conditional probabilities of transitioning from one microstate to the next (Wackermann et al., 1993). The main limitation of this approach is the fact that only the  $t \rightarrow t + 1$  transition, i.e., a single time lag is considered. On a conceptual level, the transition matrix can only fully represent a first-order Markov process, for which the complete information about the future state  $X_{t+1}$  is contained in the random variable  $X_t$ . We have shown that resting state EEG microstate sequences do not follow the Markov property, when testing Markovianity of the time series statistically (von Wegner et al., 2017). EEG microstate sequences rather show memory effects extending up to time scales of several hundred milliseconds (von Wegner et al., 2017). The statistical tests for Markovianity of order 0, 1, and 2 are contained in the software package introduced here.

As an alternative, a random walk analysis of microstate sequences has been proposed (Van de Ville et al., 2010). To use the method, however, the microstate labels (e.g., A–D) have to be mapped to real numbers, e.g.,  $\pm 1$ , in order to use Hurst exponent estimators. Furthermore, the method aggregates several microstates into one class which is mapped to a single real number (Van de Ville et al., 2010; von Wegner et al., 2016). In the case of four microstates, an arbitrary pair of microstates is mapped to the value  $-1$ , and the other two microstates are mapped to the value  $+1$ . This procedure has several disadvantages. First, there are no biologically inspired reasons which microstate maps should be grouped into one class. If all possible class assignments are tested independently, their number diverges exponentially for larger numbers of microstates. In the case of an odd number of microstates, the partition into two classes is even more difficult to justify. Second, the arithmetic operations performed on the assigned real numbers (sums, products, square roots) do not have a clearly defined meaning on the level of the EEG potential topographies they represent.

Finally, the transition matrix approach and the random walk embedding contradict each other on a theoretical level, as the first uses a memory-less Markov model and the latter uses an infinite memory, scale-free approach. To overcome these limitations, in a recent publication we introduced information-theoretical methods in order to (i) work with an arbitrary number of microstate labels directly, and (ii) to assess the memory structure of microstate sequences for all time lags, i.e., beyond  $t \rightarrow t + 1$  transitions, as captured by the transition matrix method (von Wegner et al., 2017). We also added further statistical tests for stationarity and symmetry of the transition matrix, and finally detected previously unrecognized periodicities in microstate sequences by means of the time-lagged mutual information function. Our previous publication provides an evaluation of these methods on a set of healthy subject resting state EEG recordings. In the present work, we make the methods developed and analyzed in von Wegner et al. (2017) available to other researchers. The code provided along with this manuscript allows to reproduce our previous results, and to perform new studies using the same methodology.

## 2. SOFTWARE DESIGN

The philosophy of this project is to provide a free and open-source stand-alone package. We chose to implement the algorithm in Python (Rossum, 1995) in order to provide a freely distributable, open-source, cross-platform implementation without restrictions with respect to licensed or commercial software, or the operating system used. Moreover, Python offers easily accessible source code and a reasonable trade-off between performance and code comprehensibility.

The programming style is procedural, providing a set of functions to import, pre-process, visualize, and analyze EEG data sets with the microstate algorithm and the information-theoretical metrics described in von Wegner et al. (2017). Individual functions can easily be imported into other Python projects, and the procedural approach facilitates portability, compared to an object-oriented approach introducing specific class structures for EEG data that may interfere with data structures defined in other software packages.

From our experience, the scientific process often starts with a visual exploration phase, followed by more extensive and often automated data analyses, where the code may run on headless servers or remote computation facilities using the command line. Therefore, we have chosen to implement the software as a command line tool, to be used in scripts aimed at high productivity, i.e., to process a hierarchy of directories containing the EEG files to be analyzed. Rather than providing a graphical user interface (GUI), we use an interactive IPython notebook tutorial for visual analysis during the data exploration phase. The notebook format can easily be extended and modified. By this means, preliminary and intermediate results can directly be used in a tutorial or presentation setting. Finally, as another argument to use Python and as a perspective into the processing of larger EEG databases, Python code allows straightforward extensions to database management and web applications while staying within

the same programming framework. The code is provided as a single source code file in Python 2.7 syntax and using standard code documentation following the PEP 8 style guide.

### 2.1. Requirements and Dependencies

We provide stand-alone code using minimal dependencies. Dependencies consist in standard Python packages for scientific computing and visualization and have no link to licensed or commercial software. In order to run the tutorials provided in the source code and in the IPython notebook, the user needs to have installed:

- NumPy for array handling and numerical computing.
- matplotlib for data visualization.
- SciPy for filtering, interpolation, and  $\chi^2$  statistics.
- StatsModels for multiple comparison statistics.
- scikit-learn for principal component analysis based EEG data visualization.

When only using the core functions implementing the microstate algorithm, statistical tests and information-theoretic quantities, the packages StatsModels (for multiple comparisons) and scikit-learn (for principal component analysis based visualization) can be omitted.

### 2.2. Command Line Options

From the command line, the following options are available:

- “-i” or “-input” defines the full path to a single “.edf” file to be processed.
- “-f” or “-filelist” defines the full path to a text file containing the full paths to all “.edf” files to be processed, row-wise.
- “-d” or “-directory” defines the path to a directory, from which all “.edf” files will be processed.
- “-m” or “-markovsurrogates” sets the number of Markov surrogates to be used for the computation of the mutual information confidence interval.

On the command line, all options can be viewed running the source file with the `-help` option.

### 2.3. Other Implementations of the Microstate Algorithm

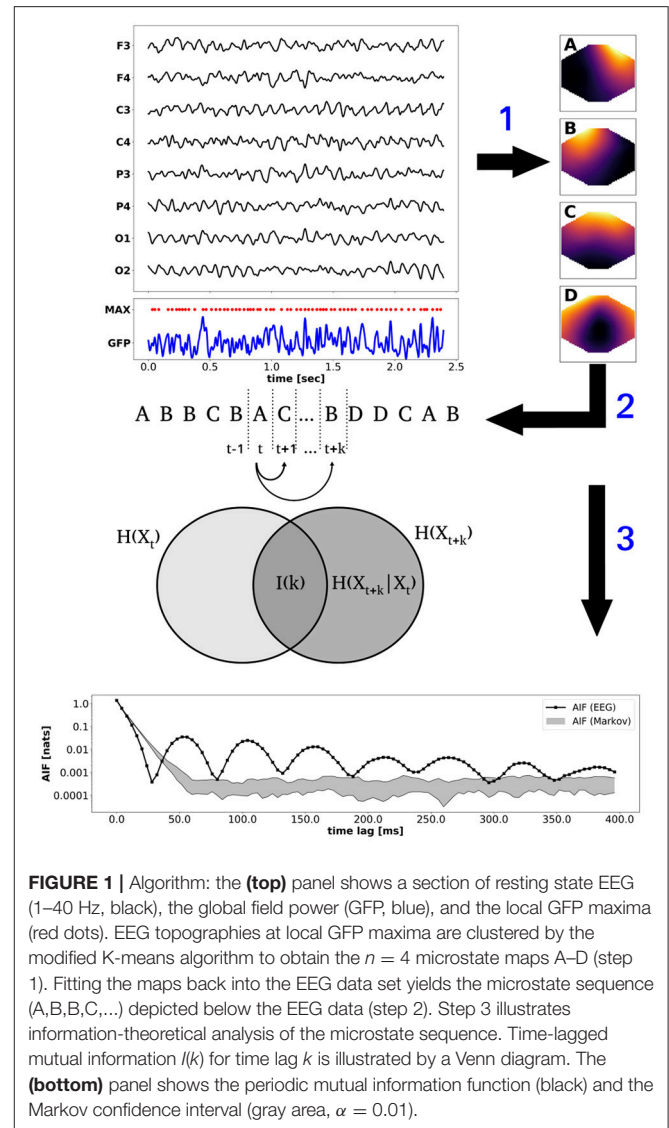
A detailed description of the original microstate algorithm is given in Pascual-Marqui et al. (1995) and Murray et al. (2008). Computational implementations exist as the Windows executable Cartool which is described in Brunet et al. (2011). The program is freely available, but ships without source code. For the commercial Matlab software, a free and open-source implementation called microstates has been published, and the package depends on the Matlab EEGLAB toolbox. Another EEGLAB-based implementation is the Microstate-EEGLab-toolbox. The large Chicago Electrical Neuroimaging Analytics software package contains the microstate algorithm as a Matlab plugin for the Brainstorm software. Recently, the probabilistic microstate analysis approach was published (Dinov and Leech, 2017), where microstates were obtained by the classical K-means algorithm rather than the modified K-means algorithm as given in Murray et al. (2008).

The Key institute Python implementation for ERP analysis contains the microstate algorithm as presented in Milz (2016).

We would like to highlight that none of the implementations listed above includes the information-theoretical analyses contained in our package. However, as the other packages contain alternative implementations of the microstate algorithm and further analytic approaches, the user may choose to use and benefit from interfacing our code with some of these software packages. This is especially easy in the case of communication with external Python code. Our code allows for easy portability to other Python programs by simply importing the corresponding functions, e.g., the Markov tests, entropy calculations, surrogate data synthesis, and the implementation of the mutual information function.

### 3. THE PROCESSING PIPELINE

In the following, we illustrate a typical processing pipeline that can be implemented with the provided functions. During presentation, it should become clear that not all presented computations have to be performed, or necessarily in the order presented in the example. In particular, the number of clusters to be computed is provided by the user, and any integer ( $\geq 2$ ) is allowed. All subsequent analysis steps are not affected by the choice and are computed for the given number of clusters. The **Figure 1** summarizes the procedure. Selected EEG channel data is illustrated on the top left, showing the channel abbreviations on the y-axis (e.g., O1 is the left occipital electrode). Below, the global field power (GFP, blue) and its local maxima (MAX, red dots) are shown. The EEG topographies at the local GFP maxima provide the input for modified K-means clustering (arrow 1). The clustering procedure yields the four microstate maps A–D shown on the right. Step 2 refers to the competitive back-fitting of the microstate maps into the EEG data set, based on a maximum squared correlation metric. The microstate time series is illustrated by the label sequence A, B, B, C... depicted below the EEG data. Step 3 corresponds to information-theoretical analysis, in particular to time-lagged mutual information. The Venn diagram visualizes mutual (or shared) information between the microstates at time point  $t$  and the  $k$ -step future time point  $t + k$  as the intersection between two sets representing the entropies  $H(X_t)$  and  $H(X_{t+k})$ . Equivalently, mutual information is defined as  $I(k) = H(X_{t+k}) - H(X_{t+k} | X_t)$ , i.e., as the difference between the uncertainty about state  $X_{t+k}$ , and the uncertainty about  $X_{t+k}$  given exact knowledge about  $X_t$ . To put it differently,  $I(k)$  measures the information about  $X_{t+k}$  that is contained in  $X_t$ . Step 3 points to the bottom panel of **Figure 1**, which shows the time-lagged mutual information function for time lags  $k$  up to 400 ms. To point out non-Markovian memory effects in experimental EEG data, we show the mutual information function for a microstate sequence from experimental EEG data, as well as a confidence interval computed from 10 Markov surrogate sequences (significance level  $\alpha = 0.01$ ). Time-lagged mutual information of EEG data is shown in black (solid line with black squares) and the Markov confidence interval is shown as a gray-shaded area. The experimental mutual information



**FIGURE 1** | Algorithm: the (top) panel shows a section of resting state EEG (1–40 Hz, black), the global field power (GFP, blue), and the local GFP maxima (red dots). EEG topographies at local GFP maxima are clustered by the modified K-means algorithm to obtain the  $n = 4$  microstate maps A–D (step 1). Fitting the maps back into the EEG data set yields the microstate sequence (A, B, B, C, ...) depicted below the EEG data (step 2). Step 3 illustrates information-theoretical analysis of the microstate sequence. Time-lagged mutual information  $I(k)$  for time lag  $k$  is illustrated by a Venn diagram. The (bottom) panel shows the periodic mutual information function (black) and the Markov confidence interval (gray area,  $\alpha = 0.01$ ).

function shows distinct oscillatory peaks not explained by the Markov model. Further details are given in the subsequent sections.

#### 3.1. EEG Data and Pre-processing

To run this tutorial, we provide a test EEG file (test.edf) which must be located in the same folder as the source code file (eeg\_microstates.py). The record contains 192 s. from an eyes-closed resting state experiment of a healthy male subject recorded with a 30 channel EEG cap in the standard 10–10 electrode configuration. The experiment was approved by the local ethics committee of the Goethe University, Frankfurt, Germany. The EEG sampling rate is 250 Hz and the data is band-pass filtered to the 1–40 Hz range. Electrode locations are given as cartesian coordinates in the cap.xyz file which is imported for visualization of EEG topographies. All files are contained in our GitHub repository. To import EEG data, the package contains a basic edf



file reader, using the publicly available specifications for the .edf file format. Data is loaded in a single line of code:

```
# (1) Load EEG data from '.edf' file
chs, fs, data_raw = read_edf("test.edf")
```

The first two return variables are a list of channel name strings and the sampling frequency in Hz. EEG data is contained in a NumPy array of shape  $(n_t, n_{ch})$ , with time samples along the rows (first index) and electrodes or channels along columns (second index). In case the provided functions are called from another Python program, EEG data must be formatted into that shape to be processed by our functions.

EEG data is usually pre-processed by a band-pass filter. As an example, we give the code for a pass band of 1–35 Hz, where *fs* denotes the sampling frequency in Hz and the data array has the time axis running along the first dimension (*axis* = 0):

```
# (2) Band-pass filtering
data = bp_filter(data_raw, (1, 35), fs)
```

To get a general impression of the data in one dimension, an option to plot the time-course of the first principal component of the multi-channel data set is included. **Figure 2** shows that the time series contains strongly amplitude modulated, irregular oscillations in the alpha frequency band. The inset in the upper right corner shows the first 8 s. of the data to illustrate alpha oscillations on a shorter time scale, as often found in EEG visualization software.

### 3.2. Modified K-Means Clustering and Competitive Fitting

Microstates are computed by the modified K-means clustering algorithm introduced in Pascual-Marqui et al. (1995) and as reviewed in Murray et al. (2008). The algorithm receives the EEG data array  $(n_t \times n_{ch})$  and the desired number of microstate maps  $n_{maps}$  as minimum inputs. Note that the number of microstates is an optional argument that can be set to any integer  $n_{maps} \geq 2$ . If not provided, the default value  $n_{maps} = 4$  is used. In the worked example, four microstates are computed. The remaining parameters (*n\_runs*, *maxerr*, *maxiter*) can be provided as additional parameters to the *kmeans* function call, otherwise the default values (*n\_runs* = 10, *maxerr* =  $10^{-6}$ , *maxiter* = 500) are used. From all K-means runs, the optimum run is selected according to the cross-validation (CV) criterion detailed in Murray et al. (2008). The CV criterion to be minimized measures the residual variance while correcting for the number of electrodes. Note that Murray recommends to (spatially) downsample EEG data with more than 64 electrodes (Murray et al., 2008). The function call looks like:

```
# (4) Modified K-means clustering
maps, x, gfp_peaks, gev, cv = kmeans(data,
    n_maps=4, doplot=True)
```

The microstate maps are returned in a  $(n_{maps}, n_{ch})$  formatted array, and the variable *x* contains the sequence of microstate map labels for the given EEG data series, i.e., its length is equal to the number of EEG time samples. The remaining return values contain the indices of the GFP peaks used for clustering, the

global explained variance (GEV) of the microstate maps, and the absolute value of the cross-validation criterion, i.e., the minimum value across all K-means runs.

The microstate ordering returned by the K-means algorithm is random, whereas a standard ordering has been established in the literature (Koenig et al., 2002). Our K-means implementation contains an option to re-label the microstates interactively before proceeding. In the case of four microstates, the standard microstate labeling is based on the map geometry. For map A, the border between positive and negative potential values runs approximately along the diagonal from the frontal left to the occipital right corners. Map B is diagonal in the opposite direction, map C has a horizontal orientation, and map D is often circular. Sometimes, slightly different maps are generated. We mostly observed the occurrence of a map D with a vertical axis instead of a circular pattern. In this case, you can either accept the results and proceed, or re-cluster the data set as the results of the K-means algorithm differ between runs.

Note that the sequence returned by the K-means algorithm contains the microstate labels as numbers,  $\{A, B, C, D\} \rightarrow \{0, 1, 2, 3\}$ , in order to accelerate information-theoretical computations. Using numbers, the microstate label can directly be used as an index in arrays and matrices. If microstate sequences generated by external software are analyzed, the labels have to be converted to the numerical values  $0, \dots, n_{maps} - 1$  before using the functions of this package.

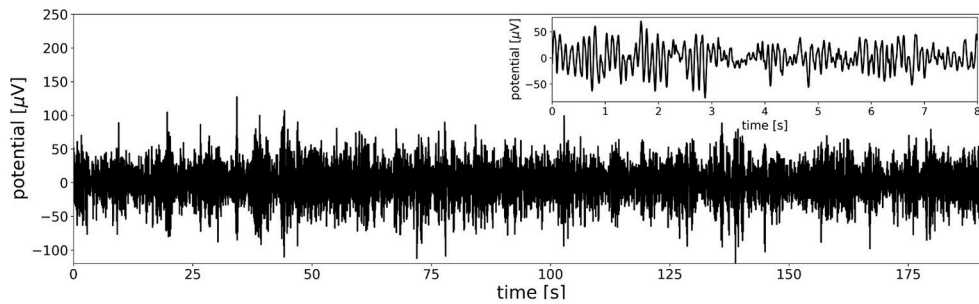
### 3.3. Information-Theoretical Analysis-Motivation and Basics

To analyze the microstate sequence with information-theoretical methods, we first need to compute the distribution of microstate labels  $P(X_t = S_i)$ , i.e., the probability to find the label  $S_i \in \{A, B, C, D\}$  at time point *t*. The distribution of  $X_t$  can be characterized by its Shannon entropy *H* (Kullback et al., 1962):

$$H = - \sum_i P(X_t = S_i) \log P(X_t = S_i). \quad (1)$$

If the sequence always repeated the same microstate label, uncertainty would be minimal and its Shannon entropy would attain its minimum value  $H = 0$ , corresponding to a delta distribution  $P(X_t)$ . Maximum entropy is obtained for a uniform distribution of microstate labels, resulting in  $H = \log(4)$  in the case of four microstates. Logarithms are taken with respect to the base *e* (Euler's constant), leading to the unit "nats."

The subsequent tests will take into account dependencies between microstate labels at different times. For one-step transitions  $X_t \rightarrow X_{t+1}$ , dependencies on the values  $X_t, X_{t-1}$ , and  $X_{t-2}$  are tested by the Markovianity tests, as detailed below. For further time lags,  $X_{t+k}$  with  $k > 1$ , temporal dependencies are assessed by mutual information between  $X_t$  and  $X_{t+k}$ . Moreover, we test time stationarity and the symmetry of the transition matrix. Each test leads to specific distributions, an empirical distribution derived from the actual data, and a reference distribution derived from the independence assumption under the null hypothesis. The distance between the empirical



**FIGURE 2 |** The overall structure of the EEG time series is visualized by the first principal component of the data set. The first principal component of the test file yields a time series similar to the signals observed at single electrodes, especially at parietal and occipital locations. The main feature of the data set are amplitude modulated oscillations in the alpha frequency band (8–12 Hz). The inset in the upper right corner shows the first 8 s of data to better visualize the characteristic alpha oscillations at a shorter time scale.

distribution ( $p_i$ ) and the null distribution ( $q_i$ ) is measured by the Kullback-Leibler divergence  $D(p, q) = -\sum_i p_i \log \frac{p_i}{q_i}$  (Kullback, 1959; Kullback et al., 1962). Statistical significance is tested with  $\chi^2$ -statistics using classical convergence theorems (Anderson and Goodman, 1957; Kullback, 1959; Billingsley, 1961; Kullback et al., 1962).

The specific test statistics along with their mathematical expressions are given in **Table 1**. In notation, we follow (Kullback et al., 1962) and denote observed frequencies by  $f$ . The estimated probability of microstate label  $S_i$ , denoted as  $p_i$ , is the ratio of  $f_i$ , the number of observations of label  $S_i$ , and the sample size  $n$ , or  $p_i = \frac{f_i}{n}$ . In **Table 1**, indices run over microstate labels and multiple sums are abbreviated by a single summation sign and the indices over which the sum is calculated.

### 3.4. Symbol Distribution and the Transition Matrix

Basic statistics commonly used to characterize microstate sequences can be obtained by calling:

```
# (5) Basic microstate statistics

# Empirical label distribution and
# transition matrix
p_hat = p_empirical(x, n_maps)
T_hat = T_empirical(x, n_maps)

# PPS (peaks per second)
pps = len(gfp_peaks) / (len(x)/fs)
print("GFP peaks per sec.: {:.2f}".format(
    pps))

# GEV (global explained variance A-D)
print("Global explained variance (GEV) per
map:" +
    keywordsstr(gev))
print("Total GEV: {:.2f}".format(
    gev.sum()))
```

The tutorial code outputs the basic statistics on the console. The distribution of the four microstate maps is  $p = [0.279, 0.222, 0.235, 0.264]$ , identical to the sometimes used term

“ratio of time covered” (RTT) (Brodbeck et al., 2012). The first order transition matrix  $T$  evaluates to:

$$T_{\text{sym}} = \begin{pmatrix} 0.799 & 0.071 & 0.069 & 0.061 \\ 0.079 & 0.771 & 0.094 & 0.057 \\ 0.053 & 0.064 & 0.767 & 0.116 \\ 0.099 & 0.061 & 0.056 & 0.784 \end{pmatrix}$$

The EEG data set at hand contains 21.51 GFP peaks per second, and the global explained variance (GEV) per microstate map in our case was 0.13 (map A), 0.10 (map B), 0.31 (map C), and 0.19 (map D), giving the total GEV of this run equal to 0.73. Due to the stochastic initialization of the K-means algorithm, other runs of the algorithm may give slightly different GEV values.

The Shannon entropy of the sequence and the maximum entropy possible for the sequence are computed as:

```
# (6) Shannon entropy
h_hat = H_1(x, n_clusters)
h_max = max_entropy(n_clusters)
```

The empirical Shannon entropy of the microstate sequence is  $H = 1.38$ , while the maximum possible Shannon entropy for any series of four symbols is  $\log(4) = 1.39$ . We see that the EEG derived sequence almost achieves maximum entropy, suggesting a process with high randomness. In the following sections, however, we show how distinct memory features such as periodicities linked to the cortical alpha rhythm can be extracted and used to characterize the sequence.

### 3.5. Markov Properties and Markovianity Tests

First, we test if the sequence follows a simple Markov process of order 0, 1, or 2. To this end, the following tests will be computed:

```
# (7) Markovianity tests
alpha = 0.01
p0 = testMarkov0(x, n_maps, alpha)
p1 = testMarkov1(x, n_maps, alpha)
p2 = testMarkov2(x, n_maps, alpha)
p_geo_vals = geoTest(x, n_maps, 1000./fs,
    alpha)
```

**TABLE 1** | Test statistics for microstate sequences.

Property	Expression	G-test statistic	d.o.f.
Markov-0	$P(X_{t+1}   X_t) = P(X_{t+1})$	$G_0 = 2 \sum_{ij} f_{ij} \log \frac{nf_{ij}}{f_i f_j}$	$(n_S - 1)(n_S - 1)$
Markov-1	$P(X_{t+1}   X_t, X_{t-1}) = P(X_{t+1}   X_t)$	$G_{1a} = 2 \sum_{ijk} f_{ijk} \log \frac{f_{ijk} f_j}{f_i f_j k}$	$n_S(n_S - 1)(n_S - 1)$
Geometric	$q_i(k) = (1 - T_{ii}) \times T_{ii}^{k-1}$	$G_{1b} = 2 \sum_{i=1}^m p_i \log \frac{p_i}{q_i}$	$m - 1$
Markov-2	$P(X_{t+1}   X_t, X_{t-1}, X_{t-2}) = P(X_{t+1}   X_t, X_{t-1})$	$G_2 = 2 \sum_{ijkl} f_{ijkl} \log \frac{f_{ijkl} f_{jk}}{f_i f_j k l}$	$n_S n_S (n_S - 1)(n_S - 1)$
Stationarity	$P(X_{t+1}^{(k)}   X_t^{(k)}) = P(X_{t+1}   X_t)$	$G_3 = 2 \sum_{ijk} f_{ijk} \log \frac{f_{ijk} f_j}{f_i f_j k}$	$(r - 1)(n_S - 1)n_S$
Symmetry	$P(X_{t+1} = S_j   X_t = S_j) = P(X_{t+1} = S_j   X_t = S_i)$	$G_4 = 2 \sum_{i \neq j} f_{ij} \log \frac{2f_{ij}}{f_{ij} + f_{ji}}$	$n_S(n_S - 1)/2$

The console output for all tests also gives the value of the test statistic and the degrees of freedom of the corresponding  $\chi^2$  distribution, calculated according to **Table 1**.

### 3.5.1. Zero-Order Markov Property

The null hypothesis is that the transition probability from the current state  $X_t$  to the next state  $X_{t+1}$  is independent on  $X_t$ . Therefore,  $P(X_{t+1} | X_t) = P(X_{t+1})$  under the null hypothesis. We obtain the corresponding test statistic  $G_0$  in **Table 1**, if the observed number of transitions  $X_t = S_i \rightarrow X_{t+1} = S_j$  is denoted  $f_{ij}$ , the number of observations  $X_t = S_i$  is  $f_i$ , and the number of observations  $X_{t+1} = S_j$  is  $f_j$ . The length of the microstate sequence  $X_t$  is  $n$ . The degrees of freedom (d.o.f.) of the asymptotic  $\chi^2$  distribution is given in the right column of **Table 1** (Kullback, 1959; Kullback et al., 1962). The console output shows that the zero-order Markovianity test yields a p-value of almost zero, within double floating point precision, under the assumption of total independence between subsequent symbols. The independence assumption is therefore clearly rejected.

### 3.5.2. First-Order Markov Property

The null hypothesis is that the transition probability  $P(X_{t+1} | X_t)$  only depends on  $X_t$ , and not on any states further in the past, implying  $P(X_{t+1} | X_t, X_{t-1}) = P(X_{t+1} | X_t)$ . Calculating the test statistic  $G_{1a}$  as given in **Table 1**, we find  $p = 4.29 \times 10^{-145}$ , indicating that a first-order Markov process is also rejected as a data model. Alternatively, first-order Markovianity can be assessed based on the equivalence of the first-order Markov property (the memoryless property) with a geometric distribution of state durations (Feller, 1971). Each microstate label has an associated lifetime distribution that contains the lengths of contiguous segments of the given label. For a first-order Markov process, the probability that label  $i$  appears in a contiguous segment of length  $k$  follows the geometric distribution  $q_i(k) = (1 - T_{ii}) \times T_{ii}^{k-1}$ . The term  $T_{ii}^{k-1}$  is the  $(k - 1)$ -th potency of the  $i$ -th diagonal element of the transition matrix  $T$ . In the G-test statistic  $G_{1b}$  in **Table 1**,  $m$  is the maximum lifetime and  $p_i$  is the empirical lifetime distribution. Testing our EEG data set for geometric lifetime distributions gives analogous results to the test statistic  $G_{1a}$ , rejecting the first-order Markov hypothesis for all four microstate maps ( $p_A = 1.33 \times 10^{-19}$ ,  $p_B = 2.68 \times 10^{-17}$ ,  $p_C = 1.57 \times 10^{-75}$ ,  $p_D = 3.24 \times 10^{-14}$ ).

### 3.5.3. Second-Order Markov Property

Second-order Markovianity is tested based on the null hypothesis that the transition probability  $P(X_{t+1} | X_t, X_{t-1})$  does not change if one step further in the past is taken into consideration. The resulting statistical expression for the null hypothesis is  $P(X_{t+1} | X_t, X_{t-1}, X_{t-2}) = P(X_{t+1} | X_t, X_{t-1})$ . Note that a true first-order Markov process, as later used in surrogate data tests, should also fulfill the second-order Markov property, as neither of the states  $X_{t-1}$  or  $X_{t-2}$  contribute information about the transition probability  $P(X_{t+1} | X_t)$ . For the test data, second-order Markovianity is clearly rejected ( $p = 3.32 \times 10^{-86}$ ).

### 3.6. Stationarity of the Transition Matrix

Stationarity of the transition matrix over time depends on the length  $L$  of the time window. For a given  $L$ , the data set is partitioned into  $r$  non-overlapping blocks of length  $L$  and the transition matrix is computed for each data block  $k = 0, \dots, r-1$ . In case of stationarity, the number of transitions  $X_t = S_i \rightarrow X_{t+1} = S_j$  within block  $k$ , denoted  $f_{ijk}$ , is independent of the block index  $k$ . The expression for the null hypothesis and the test statistic can be found in **Table 1**. The stationarity test is computed as:

```
# (8) Stationarity test for the transition matrix
p3 = conditionalHomogeneityTest(x, n_maps, L, alpha)
```

For a block size of 5,000, we obtain 9 data blocks, and the  $p$ -value of  $p = 2.21 \times 10^{-5}$  indicates that the transition matrix of the test data set is not stationary. Other block sizes can be defined in the function call, or interactively in the console and IPython tutorials provided.

### 3.7. Symmetry

If each state transition occurs with the same probability as the reverse transition, the transition matrix  $T$  will be symmetric. The expression for the null hypothesis and the test statistic are given in **Table 1**, and the symmetry test is computed as:

```
# (9) Symmetry test for the transition matrix
p4 = symmetryTest(x, n_maps, alpha)
```

The test result  $p = 4.88 \times 10^{-89}$  leads to rejection of the null hypothesis, and to the conclusion that the EEG data set has an asymmetric transition matrix. Asymmetry of the transition

matrix is important when considering non-equilibrium processes possibly underlying microstate dynamics (von Wegner et al., 2017).

### 3.8. Markov Surrogate Data

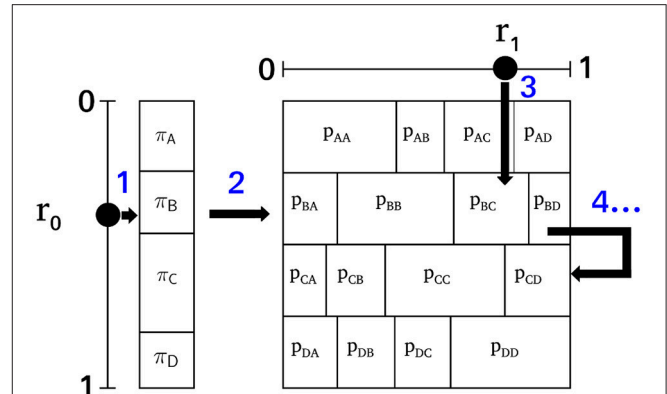
As a first-order Markov process is uniquely defined by an initial state distribution  $\pi$  and a first-order transition matrix  $T$ , an equivalent Markov process with  $\pi$  and  $T$  identical to the empirical microstate sequence can be synthesized (Häggström, 2002). The iterative construction is visualized in **Figure 3** where the individual steps are labeled with (blue) numbers. The procedure starts with an initialization function and then iterates an updating function for the desired length of the surrogate sequence. The initial state, one of the microstate labels  $A, B, C, D$ , is selected in accordance with the equilibrium distribution  $\pi$ . A pseudo-random number  $r_0 \sim U[0, 1]$ , uniformly and independently distributed on the unit interval, defines the index of the initial state by the condition  $\sum_{i=0}^{j-1} \pi_i \leq r_0 < \sum_{i=0}^j \pi_i$ . Step 1 in **Figure 3** illustrates this step showing that the equilibrium distribution  $\pi$  partitions the unit interval  $[0, 1]$ . Given the initial state, all subsequent states of the surrogate sequence are generated by the transition matrix  $T$ . The same principle as in the initialization step is used. The current state at time  $t$  determines the row of the transition matrix to be used for the next transition  $t \rightarrow t + 1$ . In **Figure 3**, the random initial state is  $B$ , so the next state is calculated from the second row of  $T$  (step 2). As the conditional probabilities in each row fulfill  $\sum_j T_{ij} = 1$ , each row of  $T$  is a partition of the unit interval. Choosing another random variable  $r_1 \sim U[0, 1]$  (step 3), the index of the next state is determined by  $\sum_{i=0}^{j-1} T_{ij} \leq r_1 < \sum_{i=0}^j T_{ij}$ . In **Figure 3**,  $r_1$  points to element  $p_{BC}$ , and thus, we record the state transition  $B \rightarrow C$  for  $t \rightarrow t + 1$ . The next state is generated in the same manner (step 4), this time using the third row of  $T$ , because the current state is now  $C$ . Using NumPy, the two computational steps used by the algorithm can be written as a single line of code each.

First-order Markov surrogates are computed as:

```
# (10) Synthesize a surrogate first-order
      Markov process
x_mc = surrogate_mc(p_hat, T_hat, n_maps,
                  len(x))
p_surr = p_empirical(x_mc, n_maps)
T_surr = T_empirical(x_mc, n_maps)
```

By construction, the synthetic Markov chain has a symbol distribution almost identical, within stochastic boundaries, to the one of the experimental data. Here, we get  $p = [0.281, 0.219, 0.239, 0.261]$  for the surrogate Markov chain. The surrogate process also has a transition matrix almost identical to the experimental data:

$$T_{\text{sym}} = \begin{pmatrix} 0.797 & 0.070 & 0.072 & 0.061 \\ 0.081 & 0.769 & 0.091 & 0.059 \\ 0.050 & 0.064 & 0.768 & 0.119 \\ 0.105 & 0.060 & 0.059 & 0.777 \end{pmatrix}$$



**FIGURE 3** | The surrogate Markov process algorithm. The initial state is selected according to the equilibrium distribution  $\pi$  using a pseudo-random number  $r_0 \sim U[0, 1]$  (step 1). In this example, state  $B$  is selected and therefore, the next state is selected using the partition of unity defined by row 2 of the transition matrix  $T$  (step 2) and the pseudo-random number  $r_1 \sim U[0, 1]$ . Here,  $r_1$  points to  $p_{BC}$  (step 3), so the next state of the process is  $C$ . Starting from state  $C$ , the successor state will be selected from the third row of  $T$ , and a new random number  $r_1$  (step 4). The algorithm can be iterated for any desired length of the surrogate sequence.

All tests calculated for the experimental data set can now be applied to the surrogate sequence, which by construction is a first-order Markov sequence:

```
# (11) Markov tests for surrogate data
p0_ = testMarkov0(x_mc, n_maps, alpha)
p1_ = testMarkov1(x_mc, n_maps, alpha)
p2_ = testMarkov2(x_mc, n_maps, alpha)
p_geo_vals_ = geoTest(x_mc, n_maps, 1000./
                    fs, alpha)
p3_ = conditionalHomogeneityTest(x_mc,
                                n_maps, L, alpha)
p4_ = symmetryTest(x_mc, n_maps, alpha)
```

The test results reveal that all desired properties are fulfilled by the surrogate Markov sequence. The sequence is not zero-order Markov ( $p = 0.00$ ), but is first-order Markov ( $p = 0.553$ ) and also second-order Markov ( $p = 0.886$ ), as expected. The alternative test for first-order Markovianity based on geometric lifetime distributions confirms the above results, as the null hypothesis is accepted for all four microstate maps ( $p_A = 0.545$ ,  $p_B = 0.302$ ,  $p_C = 0.207$ ,  $p_D = 0.797$ ). As the surrogate sequence is synthesized from a constant transition matrix, we find the sequence to be stationary ( $p = 0.180$ , block size 5,000) while reproducing the asymmetry of the experimental transition matrix ( $p = 3.84 \times 10^{-95}$ ).

### 3.9. Mutual Information

Time-lagged mutual information for discrete time lag  $k$  can be defined in entropy terms as:

$$I(k) = H(X_{t+k}) - H(X_{t+k} | X_t). \tag{2}$$



In words,  $I(k)$  measures dependencies between time points  $t$  and  $t + k$  as the difference between two entropies. The term  $H(X_{t+k})$  is the uncertainty about  $X_{t+k}$  without further knowledge about the past, and  $H(X_{t+k} | X_t)$  is the conditional uncertainty about  $X_{t+k}$ , knowing the state  $X_t$ . The time-lagged mutual information of a first-order Markov process can be written in terms of its equilibrium distribution  $\pi$  and its transition matrix  $T$  (von Wegner et al., 2017):

$$\hat{I}(k) = - \sum_i \pi_i \log \pi_i + \sum_i \pi_i \sum_j T_{ij}^k \log T_{ij}^k. \quad (3)$$

The equation uses the matrix potency  $T^k$ , computed using a diagonalization of  $T$  (von Wegner et al., 2017). The time-lagged mutual information function for the EEG microstate sequence and the Markov surrogate are calculated as:

```
# (12) Time-lagged mutual information with
# confidence interval
l_max = 100
aif = mutinf(x, n_maps, l_max)
aif_mc = mutinf(x_mc, n_maps, l_max)
```

In the tutorials, a confidence interval for the mutual information function, or autoinformation function (AIF) is computed from 10 Markov surrogates, in order to limit the computation time. The results are visualized at the bottom of **Figure 1**. The EEG-derived autoinformation function [label AIF (EEG), solid black line and squares] lies within the  $\alpha = 0.01$  confidence interval defined by  $n = 10$  Markov surrogate processes [label AIF (Markov), gray area]. Time lags up to 400 ms are shown on a y-semi-logarithmic scale.

The demonstration of non-Markovianity, non-stationarity and periodic information in resting state EEG recordings were the main results presented in von Wegner et al. (2017). This tutorial should enable the reader to reproduce these results with their own data, and to design new studies to further elucidate the functional role of EEG microstates.

## 4. IPYTHON TUTORIAL

We provide an interactive IPython notebook to illustrate a typical analysis pipeline starting from raw EEG and leading to test statistics and graphical presentations of the results. The analysis proceeds in the same order as the console tutorial, and using identical numbering of all steps for easier comparison. The notebook is part of the repository and can be visualized on the GitHub page directly, or using the nbviewer web application, pasting the `eeg_microstates` package url. The only exception is the `iframe` element calling the PubMed site for reviewing recent publications on the topic, which is not rendered on these pages for security reasons. In a running notebook, however, the link is rendered interactively.

## 4.1. Acceleration

The published code can be further optimized with respect to computational speed. As several functions involve nested loops over simple numerical values, well known Python accelerators can be applied. We tested Numba just-in-time compilation, compiled C functions using Cython, pure C functions invoked by a system call from Python and external Julia code. All methods gave considerable speedups which we do not further quantify here. We anticipate users to choose their favorite method. Though we did not include these code variants in the published package, in order to minimize dependencies and to maximize portability of the code across platforms, additional code can be obtained from the authors.

## 5. DISCUSSION AND OUTLOOK

In the present article, we introduce an open-source Python package to perform the microstate algorithm on EEG data sets, and to analyze the resulting symbolic time series using information-theoretic measures and statistical tests. We presented the application of the procedures included in the package in a recent publication (von Wegner et al., 2017). As the methods we used in the paper are not available as open-source code, to the best of our knowledge, the code is presented alongside with the theoretical basis and a tutorial. We focused on code portability in order to provide easy access of the algorithms presented here. Useful applications of the package include the comparison of information-theoretical quantities under varying experimental conditions. For instance, in the past we have used the transition matrix approach to quantify microstate sequences calculated from EEG recordings during wakefulness and non-REM sleep in healthy subjects and in synaesthesia patients (Brodbeck et al., 2012; Kuhn et al., 2015). Using the new algorithms, we can extend these analyses and add spectral information, in particular the peaks of the time-lagged mutual information function, to search for subtle differences in the temporal structure of microstate sequences. Also, the (non-)stationarity of microstate sequences can be compared under different conditions, and prior to using other algorithms requiring stationarity. The same principle can be followed to study EEG recordings (resting state or ERPs) in neuropsychiatric diseases or during cognition.

## AUTHOR CONTRIBUTIONS

FvW implemented the code, performed software tests and wrote the manuscript and tutorials. HL provided EEG data and co-designed the analysis pipeline and software structure.

## FUNDING

This work was funded by the Bundesministerium für Bildung und Forschung (grant 01 EV 0703) and LOEWE Neuronale Koordination Forschungsschwerpunkt Frankfurt (NeFF).

## REFERENCES

- Anderson, T. W., and Goodman, L. A. (1957). Statistical inference about Markov chains. *Ann. Math. Stat.* 28, 89–110.
- Antonova, I., Bänninger, A., Dierks, T., Griskova-Bulanova, I., Koenig, T., and Kohler, A. (2015). Differential recruitment of brain networks during visuospatial and color processing: evidence from ERP microstates. *Neuroscience* 305, 128–138. doi: 10.1016/j.neuroscience.2015.07.078
- Billingsley, P. (1961). Statistical methods in Markov chains. *Annals Math. Stat.* 32, 12–40.
- Brodbeck, V., Kuhn, A., von Wegner, F., Morzelewski, A., Tagliazucchi, E., Borisov, S., et al. (2012). EEG microstates of wakefulness and NREM sleep. *Neuroimage* 62, 2129–2139. doi: 10.1016/j.neuroimage.2012.05.060
- Brunet, D., Murray, M. M., and Michel, C. M. (2011). Spatiotemporal analysis of multichannel EEG: CARTOOL. *Comput. Intell. Neurosci.* 2011:813870. doi: 10.1155/2011/813870
- Dinov, M., and Leech, R. (2017). Modeling uncertainties in eeg microstates: analysis of real and imagined motor movements using probabilistic clustering-driven training of probabilistic neural networks. *Front. Hum. Neurosci.* 11:534. doi: 10.3389/fnhum.2017.00534
- Feller, W. (1971). *An Introduction to Probability Theory and Its Applications, Vol. II, 2nd Edn.* New York, NY: John Wiley & Sons Inc.
- Hägström, O. (2002). *Finite Markov Chains and Algorithmic Applications.* Cambridge: Cambridge University Press.
- Iannilli, E., Broy, F., Kunz, S., and Hummel, T. (2017). Age-related changes of gustatory function depend on alteration of neuronal circuits. *J. Neurosci. Res.* 95, 1927–1936. doi: 10.1002/jnr.24071
- Iannilli, E., Wiens, S., Arshamian, A., and Seo, H.-S. (2013). A spatiotemporal comparison between olfactory and trigeminal event-related potentials. *Neuroimage* 77, 254–261. doi: 10.1016/j.neuroimage.2012.12.057
- Khanna, A., Pascual-Leone, A., Michel, C. M., and Farzan, F. (2015). Microstates in resting-state EEG: current status and future directions. *Neurosci. Biobehav. Rev.* 49, 105–113. doi: 10.1016/j.neubiorev.2014.12.010
- Koenig, T., Lehmann, D., Merlo, M. C., Kochi, K., Hell, D., and Koukkou, M. (1999). A deviant EEG brain microstate in acute, neuroleptic-naive schizophrenics at rest. *Eur. Arch. Psychiatry Clin. Neurosci.* 249, 205–211.
- Koenig, T., Prichep, L., Lehmann, D., Sosa, P. V., Braeker, E., Kleinlogel, H., et al. (2002). Millisecond by millisecond, year by year: normative EEG microstates and developmental stages. *Neuroimage* 16, 41–48. doi: 10.1006/nimg.2002.1070
- Kuhn, A., Brodbeck, V., Tagliazucchi, E., Morzelewski, A., von Wegner, F., and Laufs, H. (2015). Narcoleptic patients show fragmented EEG-microstructure during early NREM sleep. *Brain Topogr.* 28, 619–635. doi: 10.1007/s10548-014-0387-1
- Kullback, S. (1959). *Information Theory and Statistics.* Mineola, NY: Dover Publications, Inc.
- Kullback, S., Kupperman, M., and Ku, H. H. (1962). Tests for contingency tables and Markov chains. *Technometrics* 4, 573–608.
- Lehmann, D., Faber, P. L., Galderisi, S., Herrmann, W. M., Kinoshita, T., Koukkou, M., et al. (2005). EEG microstate duration and syntax in acute, medication-naive, first-episode schizophrenia: a multi-center study. *Psychiatry Res.* 138, 141–156. doi: 10.1016/j.psychres.2004.05.007
- Lehmann, D., Ozaki, H., and Pal, I. (1987). EEG alpha map series: brain microstates by space-oriented adaptive segmentation. *Electroencephalogr. Clin. Neurophysiol.* 67, 271–288.
- Milz, P. (2016). Keppy—an open source library for EEG microstate analysis. *Eur. Psychiatr.* 33, S290–S643. doi: 10.1016/j.eurpsy.2016.01.1812
- Milz, P., Faber, P. L., Lehmann, D., Koenig, T., Kochi, K., and Pascual-Marqui, R. D. (2015). The functional significance of EEG microstates-associations with modalities of thinking. *Neuroimage* 125, 643–656. doi: 10.1016/j.neuroimage.2015.08.023
- Murray, M. M., Brunet, D., and Michel, C. M. (2008). Topographic ERP analyses: a step-by-step tutorial review. *Brain Topogr.* 20, 249–264. doi: 10.1007/s10548-008-0054-5
- Niedermeyer, E., and da Silva, F. L. (2005). *Electroencephalography. Basic Principles, Clinical Applications, and Related Fields, 5th Edn.* Philadelphia, PA: Lippincott Williams & Wilkins.
- Nishida, K., Morishima, Y., Yoshimura, M., Isotani, T., Irisawa, S., Jann, K., et al. (2013). EEG microstates associated with salience and frontoparietal networks in frontotemporal dementia, schizophrenia and Alzheimer's disease. *Clin. Neurophysiol.* 124, 1106–1114. doi: 10.1016/j.clinph.2013.01.005
- Pascual-Marqui, R. D., Michel, C. M., and Lehmann, D. (1995). Segmentation of brain electrical activity into microstates: model estimation and validation. *IEEE Trans. Biomed. Eng.* 42, 658–665.
- Rossum, G. (1995). *Python Reference Manual.* Amsterdam: Centre for Mathematics and Computer Science.
- Tagliazucchi, E., von Wegner, F., Morzelewski, A., Brodbeck, V., and Laufs, H. (2012). Dynamic BOLD functional connectivity in humans and its electrophysiological correlates. *Front. Hum. Neurosci.* 6:339. doi: 10.3389/fnhum.2012.00339
- Van de Ville, D., Britz, J., and Michel, C. M. (2010). EEG microstate sequences in healthy humans at rest reveal scale-free dynamics. *Proc. Natl. Acad. Sci. U.S.A.* 107, 18179–18184. doi: 10.1073/pnas.1007841107
- von Wegner, F., Tagliazucchi, E., Brodbeck, V., and Laufs, H. (2016). Analytical and empirical fluctuation functions of the EEG microstate random walk - short-range vs. long-range correlations. *Neuroimage* 141, 442–451. doi: 10.1016/j.neuroimage.2016.07.050
- von Wegner, F., Tagliazucchi, E., and Laufs, H. (2017). Information-theoretical analysis of resting state EEG microstate sequences - non-Markovianity, non-stationarity and periodicities. *Neuroimage* 158, 99–111. doi: 10.1016/j.neuroimage.2017.06.062
- Wackermann, J., Lehmann, D., Michel, C. M., and Strik, W. K. (1993). Adaptive segmentation of spontaneous EEG map series into spatially defined microstates. *Int. J. Psychophysiol.* 14, 269–283.

**Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 von Wegner and Laufs. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.