Spindler: a framework for parametric analysis and detection of spindles in EEG with application to sleep spindles

J LaRocco1,2, P J Franaszczuk2,3, S Kerick2 and K Robbins1

1 University of Texas, Department of Computer Science, San Antonio, Texas 78249, United States of America
2 US Army Research Laboratory, Human Research and Engineering Directorate, Aberdeen Proving Ground, Maryland 21287, United States of America
3 John Hopkins University School of Medicine, Department of Neurology, Baltimore Maryland 21287, United States of America

E-mail: Kay.Robbins@utsa.edu

Received 15 August 2017, revised 20 August 2018
Accepted for publication 22 August 2018
Published 28 September 2018

Abstract

Objective. EEG spindles, narrow-band oscillatory signal bursts, are widely-studied biomarkers of subject state and neurological function. Most existing methods for spindle detection select algorithm parameters by optimizing agreement with expert labels. We propose a new framework for selecting algorithm parameters based on stability of spindle properties and elucidate the dependence of these properties on parameter selection for several algorithms. Approach. To demonstrate this approach we developed a new algorithm (Spindler) that decomposes the signal using matching pursuit with Gabor atoms and computes the spindles for each point in a fine grid of parameter values. After computing characteristic surfaces as a function of parameters, Spindler selects algorithm parameters based on the stability of characteristic surface geometry. Main results. Spindler performs well relative to several common supervised and unsupervised EEG sleep spindle detection methods. Spindler is available as an open-source MATLAB toolbox (https://github.com/VisLab/EEG-Spindles). In addition to Spindler, the toolbox provides implementations of several other spindle detection algorithms as well as standardized methods for matching ground truth to predictions and a framework for understanding algorithm parameter surfaces. Significance. This work demonstrates that parameter selection based on physical constraints rather than labelled data can provide effective, fully-automated, unsupervised spindle detection. This work also exposes the dangers of applying cross-validation without considering the dependence of spindle properties on parameters. Parameters selected to optimize one performance metric or matching method are not optimized for others. Furthermore, elucidation of the stability of predicted indicators with respect to algorithm parameter selection is critical to practical application of these algorithms.

Keywords: EEG, sleep spindles, automated detection

Supplementary material for this article is available online
(Some figures may appear in colour only in the online journal)
1. Introduction

Spindles are narrow-band bursts of activity that dominate an EEG (electroencephalography) signal in one or more scalp areas. Sleep spindles, which are bursts of oscillatory activity in the 11 to 17 Hz frequency range, are a primary indicator of NREM (non-rapid eye movement) stage 2 sleep [1, 2]. Sleep spindles have a well-defined structure often preceded by a signal feature called a K-complex, which consists of a short negative deviation followed by a large positive deviation and then a large negative deviation. The number of spindles per minute (rate or density), spindle amplitude, and spindle duration are stable indicators within individuals [3], but vary considerably across individuals [4]. Spindle indicators generally show specific patterns of variation within different sleep cycles and over the course of a night’s sleep. Sleep spindles are thought to play a role in memory consolidation, and several studies have demonstrated a positive correlation between sleep spindle density and IQ scores [5]. Abnormalities in sleep spindle patterns are biomarkers of several neurodegenerative disorders such as Alzheimer’s disease [6], and observations of sleep spindles are used in the diagnosis of many sleep disorders [7, 8].

Techniques for manually evaluating sleep stages, sleep spindles, and other sleep features have been standardized by the American Academy of Sleep Medicine [9]. Dozens of automated methods for sleep spindle detection have been proposed in the literature, many of which involve spectral decomposition using band-pass filtering [10], Fourier transforms [11, 12], wavelet decomposition [13–15], or matching pursuit (MP) [16, 17]. Some recently proposed algorithms apply transformations to separate oscillatory and transient signals to more effectively approximate spindles [18–21].

Several notable benchmarking efforts [22] have compared sleep spindle ratings from different approaches for standardized datasets and metrics. Warby et al [23] describe an extensive effort to benchmark performance of experts, non-experts, and automated methods for sleep spindle evaluation. They developed a gold standard rating using consensus ratings of 24 experts on 2000 twenty-five second epochs extracted from the sleep recordings of 110 subjects. They found a mean $F_1$ of 0.75 for agreement between the individual expert ratings and the gold standard. The study quantifies the variability among raters and algorithms, highlighting difficulties of creating consistent standards. They also evaluated several automated algorithms and found $F_1$ agreement with the gold standard ranging from 0.21 to 0.52.

Unfortunately, the data from the Warby study is not available to other researchers for use in benchmarking. Many researchers use two publicly available archives: the Dreams Sleep Spindle Database [24] and the montreal archive of sleep studies (MASS) SS2 Cohort [25] to compare algorithms. Each of these data collections have spindle ratings by two experts. The inter-rater agreement between the two experts for each of these data collections is relatively low, and many researchers use the union of expert ratings as ground truth for evaluation.

Current spindle detection algorithms fall into two general categories: unsupervised algorithms that preselect algorithm parameters [26] and supervised algorithms that set parameters by optimizing performance relative to expert ratings [17, 27]. In contrast, this paper proposes a hybrid approach that explicitly calculates the dependence of spindle properties on algorithm parameters and uses the geometry of the resulting spindle property surfaces to select a ‘stable’ set of algorithm parameters. The resulting method (Spindler) and its evaluation framework have been implemented in a freely-available MATLAB toolbox (https://github.com/VisLab/EEG-Spindles.git). If provided with ground-truth information, Spindler produces performance surfaces for many common metrics. In order to facilitate comparison, the toolbox also includes implementations of several other spindle detection algorithms. Where possible, the toolbox uses code provided by the authors with wrappers to incorporate common processing methods.

The paper is organized as follows. Section 2 develops the Spindler algorithm, explains the automated heuristics, defines five common methods for matching detected spindles to ground truth, and introduces the performance metrics used for evaluation. Section 2 also describes the test data used for evaluation and the other algorithms implemented for comparison. Section 3 compares algorithm performance and shows how spindle properties depend on algorithm parameters for various algorithms. Section 4 discusses the strengths and weaknesses of the Spindler approach and the relationship of the proposed method to other approaches for calculating spindles. Section 5 offers concluding remarks.

2. Methods

2.1. The Spindler algorithm

The Spindler algorithm consists of three steps: signal decomposition and reconstruction on a parameter grid, evaluation of spindle properties on the parameter grid, and selection of ‘best’ parameters based on the geometry of these spindle property surfaces. As explained more fully below, spindle properties depend strongly on two parameters: $N_s$ (the number of Gabor atoms/second used to represent the signal) and $T_P$ (a threshold applied to the reconstructed signal to detect the presence of a spindle). Spindler’s other parameters can be varied, but do not strongly influence the results. All of the results of this paper use the default Spindler parameter settings as summarized in table 1.

2.1.1. Representation of the signal using matching pursuit (MP) with Gabor atoms. Spindler takes a single channel of raw EEG data as input. The results reported in this paper use channel Cz. Spindler bandpass filters the signal in the spindle frequency range (11 to 17 Hz for this paper) using EEGLAB’s pop_eegfiltnew prior to signal decomposition. Motivated by spindle geometry, Spindler approximates the signal, $s(t)$, as a weighted sum of Gabor atoms [28]:

$$s(t) \approx \sum_{n=1}^{N_s} a_n g(f_n, \sigma_n, \tau_n, \varphi_n).$$

The Gabor atoms, $g$, are characterized by four parameters:
Table 1. Spindler parameters.

| Parameter            | Defaults | Description                                |
|----------------------|----------|--------------------------------------------|
| Resolution           | 0.5      | Gabor atom resolution (Hz)                 |
| Scales \((2\sigma)\) | [0.125, 0.25, 0.5] | Gabor atom scale (s)                      |
| Phase \((\phi)\)     | 0        | Gabor atom phase                           |
| Atom rate\(^a\) \((N_s)\) | [0.01:0.01:0.4] | Atoms per second grid                     |
| Threshold\(^b\) \((T_b)\) | [0, 10\(^{-3}\), 10\(^{-3}\), 10\(^{-2}\), 0.02:0.01:0.1, 0.2:0.1:1] | Amplitude threshold grid                  |
| Frequency range      | [11, 17] (sleep) | Spindle frequency range (Hz)               |
| Min length           | 0.5      | Minimum spindle length (s)                 |
| Min separation       | 0.25     | Minimum spindle separation (s)             |
| Max length           | 3.0      | Maximum spindle length (s)                 |
| Intersect tolerance  | 0.2      | Tolerance for intersect match (s)          |
| Onset tolerance      | 0.3      | Tolerance for onset match (s)              |
| Timing tolerance     | 0.2      | Tolerance for time match (s)               |

\(^a\)A parameter used in Spindler parameter selection. Items below the double line are common to all spindle detection algorithms.

\[ g(f, \sigma, \tau, \phi) = \frac{1}{\sigma \sqrt{2\pi}} e^{-2\left(\frac{t - \tau}{\sigma^2}\right)^2} \cos(2\pi f(t - \tau + \phi)). \]

Spindler uses an MP dictionary consisting only of atoms with the parameters specified in table 1. \(N_s\), the total number of atoms used to represent the signal, is \(N_s\) times the length of the signal in seconds. \(f\) is an allowed atom oscillation frequency in Hz (in the range 11 to 17 Hz in steps of 0.5 Hz for sleep spindles), \(\sigma\) is half of the width of the atom’s Gaussian envelope (0.0625, 0.125, or 0.25), and \(\phi\) is the phase of the sinusoidal oscillation relative to the peak of the Gaussian (assumed to be 0). Temporal shifts, \(\tau\), corresponding to placement of atoms centered at every time point are allowed.

Spindler uses an efficient greedy strategy [29, 30] to project candidate atoms on the signal and to successively remove the contribution of the atom whose projection has the largest absolute value. The algorithm stops after selecting a specified maximum number of atoms/second \((N_s)\), not when the representation reaches a specified reconstruction error. This greedy strategy assures that a representation with a larger number of atoms is a superset of representations with smaller numbers of atoms. Thus, the MP representation only has to be computed once to generate the signal on a parameter grid of \(N_s\). An efficient representation merely has to retain the values of \((a, f, \sigma, \tau, \phi)\) for each atom added to the representation between the current value of \(N_s\) on the parameter grid and the next. The reconstruction of the signal for each selected \(N_s\) can be quickly recomputed. By default, Spindler uses thirty-nine evenly spaced \(N_s\) values in \([0, 0.4]\). (The spindle rate diverges for \(N_s > 0.3\) for all of the examples we have examined.)

It should be noted that a more traditional implementation of MP uses an unrestricted dictionary of Gabor atoms over the full range of frequencies and then applies post-processing to isolate the desired signal. The difficulty with this approach is that when a signal has significant artificial content, it is difficult to use post-processing to eliminate the effects of these artifacts. By first band-pass filtering the signal and then restricting the representation to only atoms that are likely to represent spindles, Spindler can handle very noisy signals.

2.12. Generation of parameter surfaces. To detect the presence of a spindle in a reconstructed signal at \((N_s, T_b)\), Spindler first takes the absolute value of the reconstructed signal and then divides by the 95th percentile of these values to reduce the effect of signal outliers. Spindler sets all scaled values above the 95th percentile to 1. This scaling restricts \(T_b\) to the interval \([0, 1]\), allowing a common grid of parameter values to be used for all datasets. Spindler forms a mask marking data points in the reconstructed signal with absolute value that exceeds the specified threshold, \(T_b\). Spindler then applies the minimum spindle separation criteria to the mask to join adjacent pieces and the minimum and maximum length criteria to eliminate spurious pieces. Spindlers are detected as regions of consecutive ones in the mask, leading to a list of spindle events for each \((N_s, T_b)\) on the parameter grid.

2.1.3. Heuristics for parameter selection. Given a list of spindle events for each point on the \((N_s, T_b)\) grid, Spindler forms three property surfaces: spindle rate, fraction of time spindling, and spindle length. The geometry of these surfaces, as shown in figure 1 for two representative sleep datasets, is similar across datasets and is used as the basis for parameter selection. \(N_s\) is plotted on the horizontal axis and each thin line corresponds to a fixed value of \(T_b\) from table 1 ranging from 0 (dark blue) to 1 (dark red) with the extreme values \(T_b = 0\) and \(T_b = 1\) emphasized using dashed and dotted teal lines, respectively. The thick solid teal line corresponds to the centered value of the respective parameter, which is computed by averaging the respective parameter values at \(T_b = 1\) and \(T_b = 0\) for each value of \(N_s\).

Spindler heuristics attempt to select algorithm parameters so that small changes in parameter values do not result in large changes in spindle property values. Observe that for small values of \(N_s\), all of the spindle rate cross-sections overlap (first row of figure 1) indicating that the spindle rate is independent of \(T_b\) in this range and changes approximately linearly with \(N_s\). Geometrically this is because the greedy MP representation adds atoms sequentially, so the representation captures only the most energetic (relatively long) spindles regardless of threshold when there are a limited number of atoms to
represent the signal. However, as $N_s$ increases, the spindle rate becomes quite sensitive to the threshold.

Spindle rate saturation occurs when all of the spindles that could be detected for a given threshold have been detected and adding more atoms to the representation does not result in more spindles. We observe a downturn and then levelling off of the spindle rate curves (top row of figure 1) as this happens. Spindle rate saturation first occurs for the $T_b = 1$ (dotted teal curve) and eventually for other $T_b$ thresholds. We restrict $N_s$ to be less than the first saturation point, which we detect as the lowest value of $N_s$ for which the spindle rate curve for $T_b = 1$ has zero slope. This condition implies that adding additional atoms to the representation produces no more spindles. We also require that $N_s$ be large enough that the standard deviation of the spindle rate across thresholds is greater than 0. When it is zero, the spindle rate is completely independent of threshold, indicating that there are not enough atoms to represent even the most prominent spindles. The resulting eligible range of $N_s$ is marked by the thick gray horizontal line on the graphs in figure 1. The darker gray curve represents the standard deviation of the spindle rate as a function of $N_s$. The gold vertical line marks the value that Spindler selected for $N_s^*$; and the thick black line highlights the curve corresponding to the value Spindler selected for $\bar{T}_b$.  

Figure 1. Spindle property surfaces in cross section for two representative datasets. The first row gives the spindle rate, the middle row shows the fraction of time spindling, and the last row shows the spindle length as functions of ($N_s, T_b$) for the first subjects of the MASS-SS2 (left column) and DREAMS (right column) data collections. The thin lines represent cross sections of the property surfaces plotted versus $N_s$ for fixed values of $T_b$ ranging from smallest (dark blue) to largest (red). Thick dashed and dotted teal lines highlight the curves for $T_b = 0$ and $T_b = 1$, respectively. The solid teal lines represent the average of the respective property value at $T_b = 0$ and $T_b = 1$. The thick gray horizontal line indicates the range of $N_s$ values that give a threshold-insensitive estimate of spindle rate. The darker gray curve represents the standard deviation of the spindle rate as a function of $N_s$. The gold vertical line marks the value that Spindler selected for $N_s^*$; and the thick black line highlights the curve corresponding to the value Spindler selected for $\bar{T}_b$. To obtain a stable representation with respect to threshold, Spindler calculates the central spindle fraction curve by averaging the curves for $T_b = 0$ and $T_b = 1$. Spindler then finds $\bar{T}_b^*$, the threshold whose spindle fraction curve remains closest to the central fraction curve in the eligible range of $N_s$. Finally, Spindler selects the value of $N_s^*$ as the value that minimizes
the spindle length for $T_s$ in the eligible range of $N_s$. As we shall see in the results, no single point in the $(N_s, T_s)$ plane optimizes performance for all methods of matching detected spindles to ground truth. The overall strategy of the Spindler method is not to try for the optimal value in the sense of agreement with expert ratings, but rather to pick a point on the parameter grid that is central so that small changes in parameter values do not result in large changes in property values. The range of eligible points is specifically selected to ‘stay away from the edges’.

2.2. Matching ground truth to detected spindles

Most standard performance metrics rely on the generation of a confusion matrix ($TP$, $FN$, $FP$, and $TN$) to match detected spindles with ground truth. True positives ($TP$) occur when detection matches ground truth. False negatives ($FN$) occur when ground truth indicates a spindle not found by detection, and false positives ($FP$) occur when detection indicates a spindle not in ground truth. True negatives ($TN$) occur when neither detection nor ground truth indicate a spindle. Unfortunately, there is no standard way to calculate the confusion matrix for spindles, and many different methods appear in the literature [14], each with advantages and disadvantages. The method for determining agreement between two sets of spindle ratings can significantly influence the performance results.

We have implemented five methods for determining agreement based on: epoch counts [18, 19], simple hits, intersections [23], onsets [14, 24], and times [31]. The toolbox reports the results of all five agreement methods (Count, Hit, Intersect, Onset, and Time) for an extensive list of performance metrics derived from the confusion matrix.

Calculation of true negatives ($TN$) is problematic [23], and Time is the only matching method that provides a direct way of calculating $TN$. We emphasize metrics in the analysis that do not rely on true negatives, but for completeness, we use a default $TN$ method proposed by Devyust et al [24], which assumes that all spindles are of fixed length (1 s for sleep spindles) and removes the time represented by true positives, false positives, and false negatives from the length of the signal. The default method then calculates true negatives as the time remaining divided by the assumed spindle length.

2.2.2. The Count method. The Count method [19] tabulates the number of spindles detected in segments of a specified length (30 s for this paper). A spindle that overlaps multiple segments only counts in the first interval of overlap. The number of true positives ($TP$) in each segment is the minimum of the number of detected spindles and the number of true spindles in that interval. If there are more true spindles than detected spindles, the difference is the number of false negatives ($FN$) in the segment. Conversely, if there are more detected spindles than true spindles in the segment, the difference is the number of false positives ($FP$) in the segment. $TN$ is calculated using the default method. The overall confusion matrix is computed as the sum of the confusion matrices computed in each segment.

2.2.2. The Hit method. The Hit method detects a $TP$ if there is any overlap between a true spindle and any detected spindle. Hit detects a $FP$ if a given detected spindle does not overlap any true spindle and a $FN$ if a true spindle does not overlap any detected spindle. Spindler finds the complementary intervals (start and ends) for the true spindles and the detected spindles and counts overlaps in the complementary intervals as true negatives. However, we caution users against using performance measures based on $TN$, particularly for the Hit method. Hit matching tends to estimate performance more optimistically than other methods for non-$TN$ metrics and less optimistically for $TN$ metrics. One detected spindle can overlap multiple true spindles without penalty (and vice versa). Further, a long period with no spindles counts as a single $TN$, rather than weighted by the interval length. Another difficulty with the Hit method is that labelling the entire dataset as a single spindle ($TP$) gives precision and recall values of 1. Hit performance is relatively insensitive to parameter selection but should be used with caution if there are too many very long events.

The Hit method has some similarities to the method used by Nonclercq et al [12]. These authors decomposed the signal into 0.5 s overlapping windows spaced 125 ms apart and classified each window as a spindle or a non-spindle based on frequency-amplitude properties of the windows. They categorized windows based on whether a particular window had matching labels for true or detected spindles. The Nonclercq algorithm categorizes windows with no label as true negatives. Because the Nonclercq algorithm relies on knowing the distribution of frequency-amplitude values for true positives, it is not suited for a general implementation.

2.2.3. The Intersect method. The Intersect method uses the union/intersection rule similar to the algorithm proposed by Warby et al [23] for sleep spindle evaluation. The Intersect method calculates the ratio between the intersection and the union of the time intervals representing two compared events (true and detected). If this ratio is greater than the intersect tolerance (0.2 s by default), the corresponding spindles are potential matches. The Intersect method chooses the event pair with the greatest eligible ratio as a match, categorizes the match as true positive, and removes the event pair from further consideration. Intersect categorizes unmatched true events as false negatives and unmatched detected events as false positives. Warby et al do not provide pseudo code for calculating true negatives. Our implementation uses the default method for calculating true negatives for the Intersect method.

2.2.4. The Onset method. The Onset method determines whether the starting times of true and detected spindles agree to within a specified onset tolerance. A true positive occurs when a detected spindle starts within onset tolerance of the start of a true spindle. A false positive occurs when a detected spindle does not start within onset tolerance of any true spindle. A false negative occurs when a true spindle does not have a detected spindle within onset tolerance of it. Our implementation of Onset uses the default method for true negatives. As with the Intersect method, Onset removes any matched events.
from further consideration. This method is frequently used for the assessment of sleep spindles with an onset tolerance of 0.5 s [14, 24]. We have found that the performance results are extremely sensitive to the onset tolerance and that 0.5 s gives results that appear to be far too optimistic. Spindler uses 0.3 s as the onset tolerance by default.

2.2.5. The Time method. The Time method, introduced by Lawhern et al [31] to evaluate alpha spindles, uses the actual length of time spent in each state rather than treating matches as events. To account for the difficulty in determining the exact starting time of a spindle, Time allows the extension of the events in either direction by a *timing tolerance* to get a better match. By default, Time uses 0.2 s as the timing tolerance. *Time* categorizes the periods during which a detected spindle does not overlap any true spindle as false positive time, and periods during which a true spindle does not overlap any detected spindle as false negative time. Any periods not labelled as TN, FP, or FN accumulate as *null agreement* or true negative times. The *Time* method does not distinguish between single overlaps and multiple overlaps, but rather calculates true positives based on the amount of time that true spindles and detected spindles overlap.

### 2.3. Performance metrics

Most common performance metrics are derived from the confusion matrix (*TP*, *FP*, *FN*, *TN*). As pointed out by Warby et al [23], true negative calculations are unreliable in this context, so metrics based on *TN* should not be used. This paper emphasizes the metrics of table 2, but Spindler calculates a comprehensive set of metrics based on the confusion matrix and has facilities for graphing any of these metrics as a function of the *N*$_t$ and *T*_b (figure 2 below).

1. **Precision:**

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

Fraction of detected spindles that are true spindles. (Also called PPV or positive predictive value.)

2. **Recall:**

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

Fraction of true spindles that are detected spindles. (Also called sensitivity, TPR, or true positive rate.)

3. **F-measure (F*$_\beta$):**

\[
F_{\beta} = \frac{(1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}
\]

Weighted harmonic mean of precision and recall. \(\beta = 2\) weighs recall more than precision. \(\beta = 1\) weighs recall and precision equally.

4. **FDR:**

\[
\text{FDR} = 1 - \text{Precision}
\]

False discovery rate.

### 2.4. Algorithms implemented for comparison

To facilitate comparisons, the Spindler toolbox also includes implementations of three unsupervised sleep-spindle detection algorithms (CWT-a7, CWT-a8, and SEM-a6) and two supervised sleep-spindle detection algorithms (Spink and McSleep). The Spindler toolbox also includes supervised and unsupervised alpha spindle algorithms, which are not discussed here. Supervised algorithms match predictions to expert ratings on part of the data set to select parameters and then calculate performance on the rest of the data. For the supervised predictions in this paper, we alternately use the first half and second half of the data as respective training sets and then average the results. Spindler can also be used as a supervised algorithm (denoted Spindler-S) using ground truth on a training set to determine the optimal (*N*$_t$, *T*_b).

2.4.1. Continuous wavelet probabilistic estimate (CWT-a7 and CWT-a8)—unsupervised. The continuous wavelet probabilistic estimate methods proposed by Tsanas and Clifford [14] decompose the signal using a continuous Morlet wavelet transform based on 131 Morlet scales and pseudo-frequencies in the range 5.4 to 40.5 Hz. Twenty-four (24) of the basis wavelets fall in the sleep spindle frequency range of 11–17 Hz. At each time sample, the algorithm estimates the probability of a spindle based on how many of the top ten wavelet coefficients at that instant have Morlet scales in the spindle frequency range. The algorithm smooths the instantaneous probability estimate using 10 nearest neighbors and combines adjacent points into spindles by applying several rules governing spindle probabilities and durations. An example of a rule is: if the two intervals have average spindle probability >0.7, are each at least 0.1 s long and less than 0.3 s apart, the intervals are merged into a single spindle. Tsanas and Clifford have provided two versions of their algorithm (denoted by CWT-a7 and CWT-a8) using different methods of smoothing the instantaneous probability function. The implementations take the frequency range as a parameter, but hardcode all thresholds, spindle duration criteria and spindle separation criteria. These thresholds and the Morlet wavelet parameters heavily rely on a signal-sampling rate of 100 Hz. The Spindler toolbox wrapper for the implementation resamples the signal to 100 Hz prior to execution. On return from the call, the Spindler toolbox wrapper combines adjacent events, matches to ground truth, and calculates performance measures.

2.4.2. Signal envelop modelling (SEM-a6)—unsupervised. Wendt et al [10] proposed a sleep spindle detection algorithm that takes input from two EEG channels: a central channel such as C3 and an occipital channel such as O1. The algorithm detects candidate sleep spindles using local extrema of the signal envelop from the central channel after filtering in the 11 to 16 Hz frequency band. The algorithm further corrects for ‘alpha effects’ by removing candidate spindles if the spindle frequency is <13 Hz and the power of the band-passed central channel in the interval defining the candidate spindle is less than the power in the occipital channel. The algorithm assumes
that the input signal corresponds to NREM sleep. We use the implementation provided by Warby et al [23]. The algorithm intrinsically relies on specific frequency filters and cannot be easily adapted for detection of other types of spindles. The Spindler toolbox provides a wrapper to the author’s implementation of this algorithm for combining adjacent events, matching to ground truth, and calculating performance measures.

2.4.3. Spinky Q-tunable wavelet estimate—supervised. Spinky [18, 19] provides a supervised method of sleep spindle detection that decomposes a signal using the tunable $Q$-factor wavelet transform. $Q$ is a hard-coded constant chosen so that the wavelets have a minimum number of cycles corresponding to the number of oscillations in the desired frequency band 11 to 17 Hz for an interval of 0.5 s. After wavelet decomposition, Spinky determines a threshold as the maximum of Sensitivity–FDR using the Count matching method on 30 s intervals. Spinky calculates minimum and maximum allowed values of this threshold based on the signal. Spindler uses a grid of values (by default spaced 10 units apart) between these maximum and minimum values for parametric analysis. The Spindler toolbox provides a wrapper for the original Spinky MATLAB implementation of spindle detection. Note that the original Spinky toolbox also provides K-complex detection and a python user interface to facilitate manual ratings, which are not included in the Spindler toolbox.

Figure 2. Performance surfaces on the $(N_s, T_b)$ plane for $F_1$ (left column) and FDR (right column) for 5 different matching strategies (C: Count, H: Hit, I: Intersect, O: Onset, T: Time) for a representative dataset (the longest stage 2 segment in MASS SS2 subject 1). Each row corresponds to a different choice of ground truth: union of two experts (first row), expert 1 (second row), and expert 2 (third row). The gold vertical line marks Spindler’s chosen $N_s^*$, and the gold circles mark the performance at Spindler’s chosen value of $(N_s^*, T_b^*)$. The thick gray horizontal line marks the candidate range of $N_s$. 

\[ MASS \ 1 \ (T_b^* = 0.3 \ N_s^* = 0.13) \]
2.4.4. McSleep—supervised. McSleep [21] and its single-channel predecessor DETOKS [20] are non-linear methods that model EEG as the sum of a sparse transient, a low frequency signal, and a sparse oscillatory signal. The methods require four regularizer parameters \((\lambda_0, \lambda_1, \lambda_2, \mu)\), a threshold \((c)\), and some block size and iteration size parameters. The authors indicate that \(\lambda_0\) and \(\lambda_1\) can be fixed based on the signal sampling frequency. They optimized the remaining parameters using the \(F_1\) metric with \textit{Count} matching. The McSleep toolbox does not provide code for parameter optimization. Our analysis uses the parameters suggested in [21] for \(\lambda_0, \lambda_1, \) and \(\mu\) for the Dreams and the MASS-SS2 datasets, respectively. Spindler uses a grid for \(\lambda_2\) and \(c\) to generate parameter curves for McSleep. By default, the grid for \(\lambda_2\) is \([20, 50]\) with spacing of 1, and the grid for \(c\) is \([0.5, 3]\) with a spacing of 0.1. This paper uses 3 channels (central frontal, central, central occipital) for the analysis.

2.5. Data used for evaluation

We evaluated the Spindler algorithm using two publicly available data collections that were acquired with informed consent and do not contain personally identifiable information.

2.5.1. Dreams sleep spindle database. We used the 30min excerpts from the publicly available Dreams Sleep Spindles Database [24]. The second expert’s ratings only included the spindle starting times. Following Devuyst et al, we assumed a spindle length of 1s for the second expert. The excerpts did not account for a particular sleep stage and were not preprocessed. In each case, experts annotated the central channel corresponding to position CZ.

2.5.2. Montreal sleep archive database (MASS). The Montreal Sleep Archive [22, 25] Cohort 1 subset SS2 consists of whole night recordings of 19 healthy subjects sampled at 256 Hz. We used all 19 recordings, 15 of which were rated by two experts and 4 of which were only rated by Expert 1 (subjects 4, 8, 15, and 16). The whole night recordings had also had ratings of sleep stage. We extracted the longest segment of stage 2 sleep in each recording to use in the analysis.

3. Results

The results are organized as follows. The first subsection analyzes Spindler in the context of unsupervised spindle detection. After presenting performance curves for a representative dataset and discussing the results in relation to matching metrics, the paper compares Spindler performance with three unsupervised spindle detection algorithms (CWT-a7, CWT-a8, and SEM-a6). The second subsection examines Spindler in the context of supervised spindle detection. After introducing performance curves for Spinky and McSleep, the paper compares Spindler and Spindler-S performance using different expert ratings as ground truth.

3.1. Spindler as an unsupervised algorithm

Figure 2 shows the Spindler performance curves for a representative sleep spindle dataset (the longest stage 2 segment in MASS SS2 subject 1) for two different metrics and several matching methods. Several features are apparent from figure 2. The performance curves have the same general shape for all matching methods, but the \textit{Onset} method performs considerably worse than the other methods and is very sensitive to the exact choice of threshold (as evidence by the spread in performance values for \textit{Onset} between \(T_b = 0\) and \(T_b = 1\)). Note that Spindler does not necessarily choose the global optimum of the designated performance metric, but rather tries to select a relatively stable value. The performance results show a clear trade-off—parameters corresponding to the highest \(F_1\) value do not result in low \(FDR\) values. The \textit{Count} matching method is usually more optimistic than the other methods. The \textit{Hit}, \textit{Intersect} and \textit{Time} methods give more central values that remain relatively close together and relatively independent of \(T_b\) in the candidate range of \(N_s\) (marked by the thick gray horizontal line in figure 2).

Another interesting feature is the dependence of performance on the choice of ground truth. Expert 1 rated spindles more conservatively than Expert 2, so Expert 2’s spindles mostly included Expert 1’s (though not exactly) for this data segment. Thus, the performance for the union of expert ratings was similar to Expert 2’s, and the \(FDR\) for Expert 1 was higher.

We used both graphical and statistical techniques to compare Spindler performance to other unsupervised algorithms. The supplementary material contains tables with statistical details (stacks.iop.org/JNE/15/066015/mmedia). Figure 3 shows a comparison of \(F_1\) (left column), and \(FDR\) (right column) performance of Spindler against three common unsupervised algorithms (CWT-a7, CWT-a8, and SEM-a6) using the five different matching methods (\textit{Count}, \textit{Hit}, \textit{Intersect}, \textit{Onset}, and \textit{Time}). Overall, Spindler has higher \(F_1\) and lower \(FDR\) values than the other three unsupervised algorithms. A paired t-test of Spindler against the other algorithms confirms that Spindler performance is significantly better than the other three algorithms for MASS-SS2 for both metrics, with \(p\) values less than \(10^{-5}\) for \(F_1\) and less than \(10^{-14}\) for \(FDR\). For the Dreams collection, Spindler performance is not significantly different from CWT-a7 performance, but significantly better than the other two algorithms. Results using other matching methods are similar. (See also table 4 below and the tables in the supplemental material.)

To better understand the practical implications for using these algorithms, we examined three spindle properties for the MASS-SS2 collection: average spindle rate (spindles min\(^{-1}\)), fraction of time spindling, and average spindle length (s). These properties are not independent since the fraction of time spindling is proportional to the product of the average spindle length and the average spindle rate. Table 3 shows the average properties for MASS-SS2 predicted by the four unsupervised algorithms as well as those from the expert raters.

Expert 1 has significantly lower spindle rates and spindle fraction than Expert 2. However, Expert 1’s spindle length is
J LaRocco et al

9

not statistically lower than Expert 2’s, possibly because of the large standard deviation exhibited in the spindle length predictions from Expert 2 on the chosen segments. Expert 2’s spindle length is not significantly different from those predicted by Spindler, CWT-a7, or SEM. However, CWT-a8 spindle length predictions are significantly higher than all others. Warby et al.’s [19] gold standard dataset expert ratings have a mean subject sleep spindle length of 0.74 s. Their report of SEM-a6’s mean of individual subject’s average spindle length was 0.55 s, which was consistent with the results we obtained for the Dreams data collection (not shown).

Spindler has significantly smaller spindle rate and spindle fraction \(p < 10^{-3}\) than CWT-a7 and SEM-a6, but spindle length predictions are not significantly different between Spindler, CWT-a7, and SEM-a6. CWT-8 has significantly higher estimates of all three properties \(p < 10^{-2}\). For comparison, consensus expert ratings of a gold standard data collection reported by Warby et al [23] have a mean subject sleep spindle rate of 2.3 spindles min\(^{-1}\), with some subjects having a spindle rate of nearly zero, while others having a spindle rate in the 5 to 10 spindle min\(^{-1}\) range. These authors also report the subject sleep-spindle rate for SEM-a6 on the gold standard clustered averages around 7 spindles min\(^{-1}\), which is consistent with the results presented in tables 3 and figure 4. Spindler and Expert 2 have average spindle rates around 5 spindles min\(^{-1}\), CWT-a8 has an average spindle rate clustered around 12 spindles min\(^{-1}\), a value well outside normal ranges. These results are confirmed by figure 4, which compares average spindle properties by subject across the MASS-SS2 cohort.
3.2. Spindler as a supervised algorithm (Spindler-S)

All methods, both unsupervised and supervised, have internal parameters. Unsupervised algorithms either preset these values or calculate them based on heuristics. Some parameter values, such as the allowed spindle frequency range and minimum spindle length, are constrained by known physical characteristics. In other cases, algorithm designers experiment to find a stable combination based on available data.

Supervised algorithms search for an optimal combination of parameter values using ‘ground truth’ labelled training data. This parameter search also provides an opportunity to examine the stability of predicted properties with respect to parameter selection as illustrated in figure 5 for two supervised algorithms recently reported in the literature: Spinky and McSleep. Spinky (left column of figure 5) has a single threshold parameter \( T \). Spinky uses a fixed grid resolution after computing the grid range based on data amplitude and frequency.

Figure 5 shows that Spinky’s average spindle length is relatively flat, but that spindle rate and spindle fraction have steep declines as \( T \) increases beyond 50. The Spinky parameter curves for all of the datasets examined in this study have a similar general shape.

McSleep has a large number of parameters as described in the methods section. Following the suggestions of [21], we fix \( K = 256 \), \( O = 128 \), \( \mu = 0.5 \), \( N\text{it} = 40 \), \( \lambda_0 = 0.6 \), and \( \lambda_1 = 7 \) for the MASS-SS2 collection. These parameters must be determined empirically for a given collection, and we have found the results to be somewhat sensitive to the selection of \( \lambda_0 \) and \( \lambda_1 \), but do not explore this dependence for this work. We compute property/performance surfaces on a two-dimensional grid of \( (c, \lambda_2) \). This paper uses a grid for \( c \) in the range \( [0.5, 3.0] \) in increments of 0.1, and a grid for \( \lambda_2 \) in the range \( [20, 50] \) in increments of 1. McSleep spindle length is relatively independent of parameter selection, but spindle rate and spindle fraction decrease strongly with \( \lambda_2 \), particularly for small values of \( c \).

Figure 6 shows the dependence of the \( F_1 \) metric on parameters for Spinky (left column) and McSleep (right column). Colors indicate matching method. The performance curves have roughly the same shape for all matching methods. Performance measures for figure 6 are computed on the entire selected data segment rather than splitting the segment into training and test sets, so the results are comparable to figure 2.

Both Spinky and McSleep have differently shaped parameter curves for ground truth based on the different experts, resulting in different values of spindle properties predicted by the algorithms. The union of expert ratings results are close to those of Expert 2 when Expert 2’s ratings are available. McSleep results also show a dependence on the parameter \( c \).
which strongly influences its spindle rate and spindle fraction predictions.

Table 4 compares average performance for different supervised and unsupervised methods for the MASS-SS2 collection using two metrics ($F_1$ and $FDR$), two versions of ground truth (Expert 1 and Expert 2), and the Time matching method. As a point of reference, table 4 also shows performance of each expert rate using the other rater as ground truth. Table 4 indicates that the performance results are heavily influenced by which expert rater is taken to be ground truth. Every algorithm (both unsupervised and supervised) performs significantly worse on each metric when Expert 1 is used for ground truth than when Expert 2 is used.

As shown by figure 5, spindle properties (particularly rate and fraction) are strongly influenced by the parameter selection for both Spinky and McSleep. This in turn, depends on which expert is used as ground truth. Table 5 compares average property values for the two choices of expert. McSleep gives significantly smaller average spindle length than either Spindler or Spinky, regardless of the expert used to determine ground truth.

Figure 7 displays these results broken down by subject for the MASS-SS2 data collection. The predictions from Expert 2 are generally higher than those from Expert 1 for all measures and algorithms. The supervised property predictions generally bracket the predictions from the unsupervised version of Spindler. There is quite good agreement on the spindle fraction among algorithms. Supervised Spindler’s length predictions appear to be on the high side.

4. Discussion

It is often difficult to assess how many of the published spindle detection algorithms will perform in practice. Comprehensive studies such as Warby et al [23] are an important step in understanding sources of variability in spindle assessment. As shown in [23], ‘ground truth’ is never absolute, and there is significant disagreement among expert raters. Reporting results on known publicly available data is essential for reproducing results. However, beyond reporting the start and end times of the detected events, researchers should also document how smaller segments were combined to form larger
spindles and how matching with ‘ground truth’ was accomplished, since these choices can dramatically affect results. By providing a common infrastructure for this evaluation, the Spindler toolbox allows a more realistic comparison among approaches. A key result of this work is the demonstration that optimizing a performance metric without considering stability of predicted properties can lead to unrealistic results. Reporting cross validation results without taking into account these dependencies does not give a true picture of performance.

Table 4. Average performance (SD) of unsupervised and supervised spindle detection algorithms for MASS-SS2 based on ratings from two experts using Time matching. For supervised algorithms, $F_1$ is used as the base metric for training.

| Algorithm | Expert 1: $F_1$ | Expert 2: $F_1$ |
|-----------|-----------------|-----------------|
| Spindler-S  | 0.57 (0.17)     | 0.67 (0.07)     |
| CWT-a7     | 0.41 (0.13)     | 0.61 (0.09)     |
| CWT-a8     | 0.26 (0.11)     | 0.51 (0.11)     |
| SEM        | 0.36 (0.16)     | 0.65 (0.08)     |
| Spindler-S  | 0.61 (0.14)     | 0.73 (0.05)     |
| McSleep    | 0.70 (0.11)     | 0.75 (0.07)     |
| Spindler-S  | 0.66 (0.11)     | 0.74 (0.06)     |
| Spinky     | 0.52 (0.15)     | 0.52 (0.15)     |

Table 5. Average properties (SD) of supervised algorithms and experts for MASS-SS2.

| Algorithm | Expert | Spindles (min) | Spindle fraction | Spindle length (s) |
|-----------|--------|----------------|------------------|-------------------|
| Spindler-S  | 1      | 3.37 (1.74)    | 0.06 (0.02)      | 1.19 (0.25)       |
| McSleep    | 1      | 3.37 (1.63)    | 0.05 (0.03)      | 0.85 (0.09)       |
| Spindler-S  | 2      | 6.72 (1.84)    | 0.14 (0.04)      | 1.28 (0.20)       |
| McSleep    | 2      | 7.60 (2.22)    | 0.12 (0.04)      | 0.96 (0.10)       |
| Spindler-S  | 2      | 6.76 (1.65)    | 0.14 (0.04)      | 1.22 (0.13)       |

Sleep spindle detection methods commonly use Intersect or Onset for matching detected spindles to ground truth. Both approaches treat matches as distinct and remove matched events from further consideration. These methods assume fixed spindle length when computing true negatives, an assumption that empirical data does not support. We concur with Warby et al. [23] that performance metrics that rely on true negatives should not be used due to the severe imbalance between the spindles and non-spindles.

We have focused on the metric $F_1$, which balances precision and recall, as well as FDR. The choice of ‘best’ parameters for supervised algorithms depends strongly on the metric chosen as well as other factors. Onset F-measures are generally lower than all other measures for the same dataset and are
very sensitive to choice of onset tolerance. Time, Intersect, and Hit give fairly close estimates of $F_1$ for Spindler in the stable region. Time is slightly more optimistic than Intersect for Spinky and McSleep. The Hit metric can be susceptible to imbalance. If an algorithm declares everything is a spindle, precision and recall and their derived measures can be nearly one, while these metrics are nearly zero for other matching algorithms. The Count method consistently overestimates performance for all metrics and algorithms. Overall, we believe that either Intersect or Time should be used for performance evaluation. We prefer Time because it gives an intrinsic measure of true negatives. Most of the results reported in this paper use the Time method.

All of the algorithms that we have examined have hidden parameters and embedded constant values selected by ad hoc methods. Because researchers generally test their algorithms using a small collection of test data, ad hoc choices may not be applicable in a larger context. We believe that it is important for algorithms to fully expose all hidden parameters for effective evaluation. We also believe that it is important to evaluate the dependence of spindle properties on parameter selection. We manually examined the Spindler parameter curves generated for all of the DREAMS and MASS-SS2 datasets and verified that the parameter curves were similar to those displayed in figure 1.

Algorithms can also be sensitive to artifacts, and some authors report results after artifacts have been manually removed. We believe that either artifacts should be removed using a standardized automated method or that the algorithm should be evaluated on unprocessed data. The results of this paper use raw signal, as might be expected in practice.

Several other MP-based methods using Gabor atoms or a combination of Gabor, Fourier and Dirac atoms have been proposed in the literature [17, 26, 27, 32, 33]. These algorithms decompose the signal into short windows (0.5 s, 16 s, 20 s or 30 s depending on the method), use a full dictionary of atoms to decompose the signal in each window, and then consider atoms in the decomposition that have frequencies and scales appropriate for spindles. CWT-a7 and CWT-a8 use a related strategy, decomposing the signal at each time point using a continuous wavelet transformation with Morlet wavelets from a specified dictionary. These algorithms then calculate the probability of a spindle by examining the portion of the 10 wavelets at each point that are in the spindle frequency and scale range. All of these algorithms are local.

In contrast, Spindler is a global algorithm that decomposes the entire signal using a dictionary consisting only of atoms that capture spindles. Spindler then constructs a global parameter surface to set dataset-specific thresholds. Unlike most other algorithms, Spindler produces a reconstruction.
of the spindle signals in addition to the spindle events. This reconstruction makes it possible to study synchronization of spindles across channels or frequency shifts using these reconstructed signals. Spindler can also handle source signals, such as those produced by independent component analysis, to map spindle features to networks.

While we have only addressed single-channel detection in this paper, the channel label is a settable parameter for all of the algorithms implemented in the toolbox. Lawhern et al. [31] have a voting method for combining channels in a scalp area to get a single measure. Such methods can be easily applied in a second stage by using the information that Spindler saves as input.

5. Conclusion

Our comparisons of unsupervised (Spindler, SEM-a6, CWT-a7, CWT-a8) sleep spindle algorithms show that Spindler provides comparable or better performance in a variety of settings and uses unprocessed EEG data. Spindler can also be applied in a supervised setting (Spindler-S) and gives property predictions that are comparable to Spinky and McSleep. An automated sleep spindle detection method with high performance during sleep in middle aged subjects 2012 Annual Int. Conf. IEEE Eng. Med. Biol. Soc. pp 4250–3.

Acknowledgments

This research was sponsored by the US Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0022. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory or the US Government. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

ORCID IDs

P J Franaszczuk https://orcid.org/0000-0002-5166-4224
K Robbins https://orcid.org/0000-0002-7147-5797

References

[1] De Gennaro L, Ferrara M 2003 Sleep spindles: an overview Sleep Med. Rev. 7 423–40
[2] Coppériers’t Wallant D, Maquet P and Phillips C 2016 Sleep spindles as an electrographic element: description and automatic detection methods Neural Plast. 2016 e6783812
[3] Tan X, Campbell I G and Feinberg I 2001 Internight reliability and benchmark values for computer analyses of non-rapid eye movement (NREMS) and REM EEG in normal young adult and elderly subjects Clin. Neurophysiol. 112 1540–52
[4] De Gennaro L, Ferrara M, Vecchio F, Cucchio G and Bertini M 2005 An electroencephalographic fingerprint of human sleep NeuroImage 26 114–22
[5] Fogel S M and Smith C T 2011 The function of the sleep spindle: a physiological index of intelligence and a mechanism for sleep-dependent memory consolidation Neurosci. Biobehav. Rev. 35 1154–65
[6] Petit D, Gagnon J-F, Fantini M L, Ferrini-Strambi L and Montplaisir J 2004 Sleep and quantitative EEG in neurodegenerative disorders J. Psychosom. Res. 56 487–96
[7] Espa F, Ondze B, Deglise P, Billiard M and Besset A 2000 Sleep architecture, slow wave activity, and sleep spindles in adult patients with sleepwalking and sleep terrors Clin. Neurophysiol. 111 929–39
[8] Himanen S-L, Virkkala J, Huupponen E and Hasan J 2003 Spindle frequency remains slow in sleep apnea patients throughout the night Sleep Med. 4 361–6
[9] Berry R, Brooks R, Gamaldo C, Harding S, Lloyd R, Marcus C and Vaughn B 2017 AASM scoring manual updates for 2017 (Version 2.4) J. Clin. Sleep Med. 13 665–6
[10] Wendt S L, Christensen J A E, Kempfner J, Leonthin H L, Jennum P and Sorensen H B D 2012 Validation of a novel automatic sleep spindle detector with high performance during sleep in middle aged subjects 2012 Annual Int. Conf. IEEE Eng. Med. Biol. Soc. pp 4250–3
[11] Huupponen E, Gómez-Herrero G, Saastamoinen A, Värrí A, Hasan J and Himanen S-L 2007 Development and comparison of four sleep spindle detection methods Artif. Intell. Med. 40 157–70
[12] Nonclercq A, Urbain C, Verheulpen D, Decaestecker C, Van Bogaert P and Peigneux P 2013 Sleep spindle detection through amplitude-frequency normal modelling J. Neurosci. Methods 214 192–203
[13] Adamczyk M, Genzel L, Dresler M, Steiger A and Friess E 2015 Automatic sleep spindle detection and genetic influence estimation using continuous wavelet transform Frontiers Hum. Neurosci. 9 624
[14] Tsanas A and Clifford G D 2015 Stage-independent, single lead EEG sleep spindle detection using the continuous wavelet transform and local weighted smoothing Frontiers Hum. Neurosci. 9 181
[15] O’Reilly C, Godbout J, Carrier J and Lima J-M 2015 Combining time-frequency and spatial information for the detection of sleep spindles Frontiers Hum. Neurosci. 9 70
[16] Durka P J and Blinowska K J 1996 Matching pursuit of sleep spindles Proc. 18th Annual Int. Conf. IEEE Eng. Med. Biol. Soc. vol 3 pp 1011–2
[17] Schönwald S V, de Santa-Helena E L, Rossatto R, Chavez M L F and Gerhardt G J L 2006 Benchmarking matching pursuit to find sleep spindles J. Neurosci. Methods 156 314–21
[18] Lajnef T, Chaibi S, Eichenlaub J-B, Ruby P M, Aguera P-E, Samet M, Kachouri A and Jerbi K 2015 Sleep spindle and K-complex detection using tunable Q-factor wavelet transform and morphological component analysis Frontiers Hum. Neurosci. 9 414
[19] Lajnef T et al 2017 Meet spinky: an open-source spindle and K-complex detection toolbox validated on the open-access Montreal archive of sleep studies (MASS) Frontiers Neuroinform. 11 15
[20] Parekh A, Selesnick I W, Rapoport D M and Ayappa I 2015 Detection of K-complexes and sleep spindles (DETKOS) using sparse optimization J. Neurosci. Methods 251 37–46
[21] Parekh A, Selesnick I W, Osorio R S, Varga A W, Rapoport D M and Ayappa I 2017 Multichannel sleep spindle detection using sparse low-rank optimization J. Neurosci. Methods 288 1–16
[22] O’Reilly C and Nielsen T 2015 Automatic sleep spindle detection: benchmarking with fine temporal resolution using open science tools *Frontiers Hum. Neurosci.* **9** 353

[23] Warby S C, Wendt S L, Welinder P, Munk E G S, Carrillo O, Sorensen H B D, Jenum P, Peppard P E, Perona P and Mignot E 2014 Sleep-spindle detection: crowdsourcing and evaluating performance of experts, non-experts and automated methods *Nat. Methods* **11** 385–92

[24] Devuyst S, Dutoit T, Stenuit P and Kerkhofs M 2011 Automatic sleep spindles detection—overview and development of a standard proposal assessment method 2011 Annual Int. Conf. IEEE Eng. Med. Biol. Soc. pp 1713–6

[25] O’Reilly C, Gosselin N, Carrier J and Nielsen T 2014 Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research *J. Sleep Res.* **23** 628–35

[26] de Souza R T F, Gerhardt G J L, Schönwald S V, Rybarczyk-Filho J L and Lemke N 2016 Synchronization and propagation of global sleep spindles *PLOS One* **11** e0151369

[27] Durka P J, Malinowska U, Zieleniewska M, O’Reilly C, Różański P T and Żygierewicz J 2015 Spindles in Svarog: framework and software for parametrization of EEG transients *Frontiers Hum. Neurosci.* **9** 258

[28] Mallat S G and Zhang Z 1993 Matching pursuits with time-frequency dictionaries *IEEE Trans. Signal Process.* **41** 3397–415

[29] Krstulovic S and Gribonval R 2006 MPTK: Matching pursuit made tractable 2006 IEEE Int. Conf. on Acoustics Speech and Signal Processing Proc. vol 3 pp 496–9

[30] Mineault P 2011 Matching pursuit for 1D signals—file exchange—MATLAB Central (www.mathworks.com/matlabcentral/fileexchange/32426-matching-pursuit-for-1d-signals)

[31] Lawhern V, Kerick S and Robbins K A 2013 Detecting alpha spindle events in EEG time series using adaptive autoregressive models *BMC Neurosci.* **14** 101

[32] Huupponen E et al 2006 Determination of dominant simulated spindle frequency with different methods *J. Neurosci. Methods* **156** 275–83

[33] Schönwald S V, Carvalho D Z, Dellagustin G, de Santa-Helena E L and Gerhardt G J L 2011 Quantifying chirp in sleep spindles *J. Neurosci. Methods* **197** 158–64