Real-Time Detection of High-Frequency Oscillations and Computations of Coherence from MEG Signals to Facilitate the Prediction of Epileptic Seizures

Jessica Yatvitskiy
The Pingry School
131 Martinsville Road
Basking Ridge, NJ 07920

Acknowledgements

This work would not have been possible without the guidance of my mentor, Dr. Davide Valeriani of the Dystonia and Speech Motor Control Laboratory at the Massachussets Eye and Ear Infirmary (MEEI), who suggested scientific literature for me to review and proposed the idea of investigating coherence and high frequency oscillations (HFOs) as factors for seizure prediction.

I would also like to thank Dr. Farid Sichani, a neurologist at MEEI, who confirmed the start and end times of the seizure in the magnetoencephalography data I had received and affirmed the accuracy of my HFO detection algorithm (by reviewing plots that I developed of the detected HFOs).

Finally, I would like to offer special thanks to Dr. Kristina Simonyan, Harvard Medical School director of laryngology research, who accepted me as a volunteer in her laboratory this summer.

Table of Contents

1. Introduction **4**

2. Methods **5**

2.1 Background, Challenges, and Objective 5

2.2 Computation of coherence 6

2.3 Time-efficient implementation of coherence computation 8

2.5 Training of machine learning model 10

2.6 Simulation of real-time MEG data flow 11

2.7 Processing of MEG data in real-time 11

2.8 Parallelization of computations across CPUs 12

2.9 Aggregation of results across channels and across time windows, with subsequent generation of seizure prediction 13

2.10 Computation of inter-channel coherence for different parameter sets 14

3. Results **15**

3.1 Validation of accuracy 15

3.2 Comparison of performance: new algorithm versus traditional method of computing coherence 16

3.3 Speed of real-time processing, relative to speed of arrival of new data 17

3.4 Dependency pattern between inter-channel coherence spike and epileptic seizure 17

3.5 Dependency of computed coherence on parameters used in its computation. 19

4. Discussion **21**

4.1 Implications 21

4.2 Limitations 21

4.3 Potential Directions 22

5 References **23**

1. Introduction

Closed-loop (i.e, adaptive) systems are becoming increasingly relevant to neuroscience and have been shown to be applicable to the treatment of various movement disorders, such as Parkinson's disease, dystonia, and essential tremor [1]-[3]. Such systems can also be applied to facilitate early epileptic seizure prediction, with subsequent seizure control [4]. It has been shown that seizures are usually preceded by HFOs, which are defined as changes in electrical potentials in EEG and MEG in frequencies between 80 and 500 Hz [5]-[6]. Establishing the presence of HFOs in the interictal state (the period between two seizures) and the preictal state (the period immediately before the seizure) corroborated the idea that HFOs can be used to predict the onset of a seizure. However, HFOs themselves have not shown to be sufficient for accurate predictions. They may need to be used for predictions in combination with other factors that accompany seizures. Changes in inter-channel coherence, which reflect the degree of synchronization of the phases for a given frequency for each pair of channels, were also recently shown to be tied to the onset of seizures in epilepsy [9]-[12].

So far, most of the research in epileptic seizure prediction was performed using intracranial electroencephalography (iEEG) data, because it provides a better signal-to-noise ratio than do non-invasive techniques, such as EEG and MEG [7]. The use of software that is capable of filtering out noise while enhancing the relevant signal is required in order to detect HFOs reliably from EEG and MEG data. MEG is more accurate than EEG, because the skull and the tissue surrounding the brain affect the magnetic fields measured by MEG much less than they affect the electrical impulses measured by EEG, which leads to better spatial resolutions by MEG [8]. Thus, a file containing MEG data was chosen for use in this research.

This project aimed at developing a high-performance computational model for the closed-loop prediction of epileptic seizures from MEG data. The model, while being flexible enough to accommodate any additional factors that will be found useful in the prediction of seizures, focused on the two that are already known to be relevant to seizures: coherence and HFO events.

Once such computational model was built, it became possible to compute values of inter-channel coherence for a wide range of time windows, time shifts between time windows of different MEG channels, and overlaps of time windows within a given channel, for MEG data from a patient, in less than a day. The dependency between computed inter-channel coherence and an occurrence of epileptic seizure was, subsequently, visually analyzed.

2. Methods

2.1 Background, Challenges, and Objective

When I began my research in the Dystonia and Speech Motor Control Laboratory at the Massachusetts Eye and Ear Infirmary / Harvard Medical School, my long-term objective was to develop a model for the prediction of epileptic seizures. Based on recent publications on the topic of the seizures, a rise in the number of HFOs and an increase in the degree of coherence across brain channels stood out as two likely markers of the approaching recurrence of a seizure. For my analysis, I had a 24-minute long MEG recording of neural data from an epileptic patient. 102 MEG channels had been selected for this recording, based on the relevance to epileptic seizures of the area of the brain that corresponded to each channel. The data were sampled at 1000 Hz, thus supporting the extraction of frequencies up to 500 Hz, which is the high-level frequency cutoff for HFOs. For the processing of the data, I used a Dell Computer, with a 3.4GHz i7 processor, 8 logical cores, a 16GB RAM, and a Microsoft Windows 10 Operating System.

While the computation of per-channel HFOs, described below, did not present performance issues, the computation of coherence showed that the use of readily available methods for the calculation of coherence are not suitable for the processing of neurological models that involve MEG data. The fastest available off-the-shelf implementation of coherence calculation comes from the Python package scipy, an open-source package for mathematics, science, and engineering. But even with that relatively fast method, it took over 11 hours to compute the coherence for the 24-minute data file, using all of the cores on my computer. Clearly, such slow processing would impede real-time prediction of epilepsy seizures, as well as slow down further research involving larger files from multiple patients.

That finding set an immediate goal for my research: the creation of a flexible closed-loop model, with a focus on factors deemed critical in seizure prediction, that can perform the necessary computations approximately as fast as the patient's data is recorded.

2.2 Computation of coherence

The objective of the seizure prediction model is to predict, based on N seconds of data starting from time T, the probability that a seizure will be occurring during the N second interval starting from time T+M, where M is the number of seconds into the future we want to make the prediction. Thus, the coherence aggregated over all channels over N seconds must be computed, which should be done as frequently as possible. For this research, the coherence between channels is recomputed for every 500 ms time interval.

For each 500 ms time window, the coherence between each pair of channels is computed 7 times (Figure 1): between a time window of channel 1 starting at time T and time windows of channel 2 starting at T-100 ms, T, T+100 ms, T+200 ms, T+300 ms, T+400 ms, and T+500 ms. In other words, the start time for the window of channel 2 is shifted 100 ms at a time. This is done because different parts of the brain have a natural phase lag relative to each other [16], across all frequencies, and this phase lag must be accounted for when computing coherence between MEG channels, which represent different areas of the brain. For each of the 7 above mentioned combinations, a coherence for each frequency up to 500 Hz is calculated. The mean across all frequencies is then computed. The maximum of these 7 means is considered to represent the coherence between the two channels for the time window of 500 ms starting from time T.

Subsequently, the mean of the coherence values across all pairs of channels for this window is computed and thus represents the overall coherence (across all channels) for the current time window. Finally, this value is stored in a channel by channel by window "coherence matrix".

Based on the above, the low-level computation of coherence over 500 data points (the data contains one point per millisecond), is done num_channels * num_channels * num_shifting_offsets times every 0.5 seconds, which translates into about 140K computations per second.

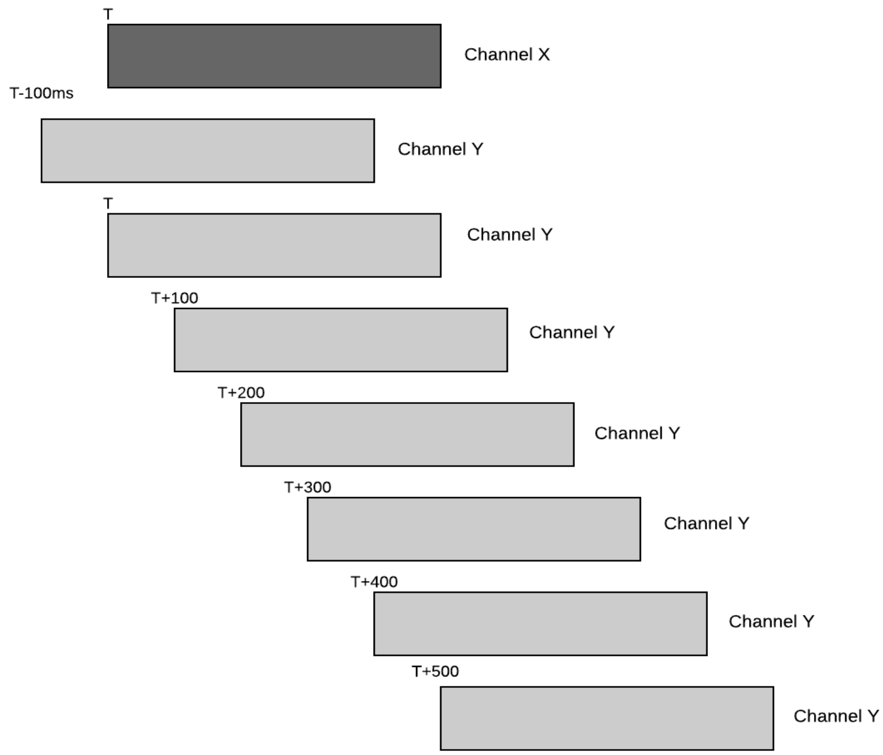

Figure 1: For each 500 ms time window (such as the one above, starting from time T), the coherence between each pair of channels (such as channels X and Y, above) is computed 7 times, i.e., once for each 100 ms shift of the window of the second channel.

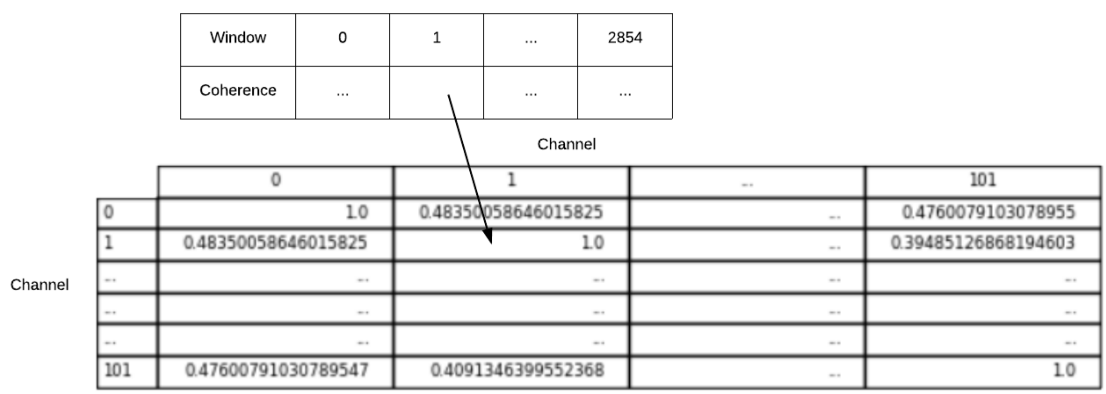

Figure 2: For each 500 ms time window, a channel-by-channel coherence matrix is computed.

2.3 Time-efficient implementation of coherence computation

In attempting to compute the “coherence matrix” discussed above, it was found that 1000 invocations of scipy’s coherence function take 600 ms. Thus, it would take 84 seconds to process 1 second of data using scipy. Even if all 8 cores on my computer were used fully in parallel, it would still take more than 10 seconds to process 1 second of data. Clearly, a different method of computing coherence had to be found.

Upon a closer look into scipy’s source code for computing coherence, it was determined that it uses the following conceptual model:

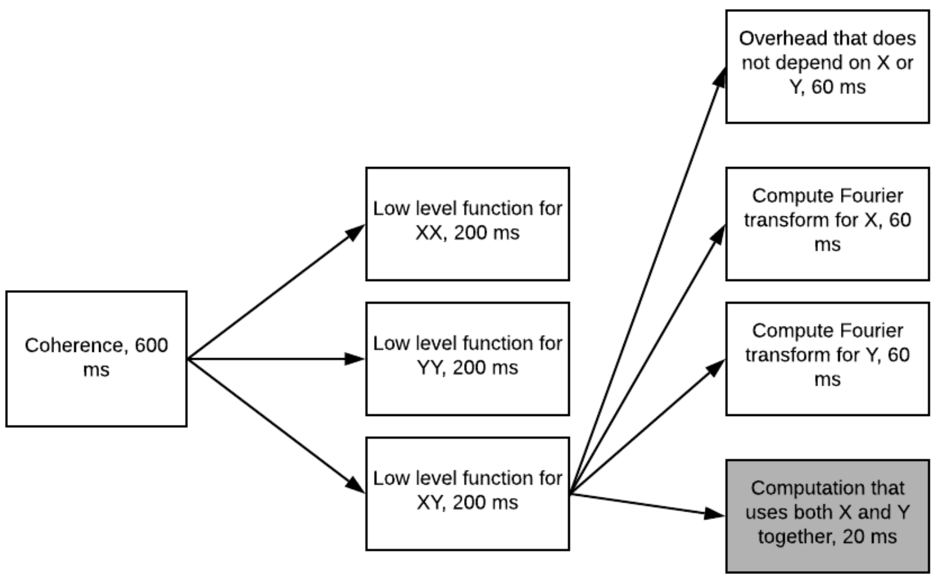

Figure 3: Flowchart of the high-level logic used by scipy’s coherence method. Only the logic in gray box has to be executed for every combination of signals X and Y. All other steps can be executed only once per channel.

As shown in Figure 3, each call to scipy’s coherence method splits into three identical sub-methods: one to compute values that involve only channel X, another to compute values that involve only channel Y, and a third to compute values for a combination of channels X and Y. Clearly, there is no need to make the former two computations 102*102 times. They can be done once per channel, i.e., 102 times. Furthermore, the third computation, for a combination of X and Y, involves relatively costly computation of the Fourier transform for both channels, independently of one another. These two Fourier transform calculations consume about 60% of the time spent on the computation for a combination of channels X and Y. Of course, the transforms also need to be performed only 102 times, i.e., once per channel, rather than 2 * 102 * 102 times. In addition, it was found that any invocation of scipy's coherence method involves a fixed overhead of almost 30% of the entire time taken by this method. That fixed amount of processing can be done once. The amount of processing that has to involve both channel X and channel Y should hence comprise only about 3.3% of the time that the scipy coherence function takes.

Thus, the closed-loop model presented in this paper splits the computation of coherence into three steps:

1. The computation of the initial coherence state - performed once
2. The computation of the coherence state for each MEG channel - done once for each of the 102 channels, every 500 ms
3. The computation of coherence between each pair of MEG channels - done 102 * 102 * 7 (7 is the number of shifting offsets described in 3.2) times every 500 ms

2.4 HFO detection

The number of HFO events per time window, averaged over 10 time windows, was used as another factor for the closed loop prediction model. The number of HFO events per time window was itself computed as the average, across all 102 channels, of the number of HFOs detected for each given channel.

To detect HFOs, an algorithm based on that described in [5] was used. The first step in the detection of HFO events was finding all peaks in the time series of data. Of these peaks, those that have an amplitude of at least 3 standard deviations above the mean and whose distances from the two adjacent peaks do not differ from each other by more than 2 ms and do not exceed 10 ms when added, are marked as “high peak”. The sequences of 6 or more adjacent “high peaks” are identified as HFO events as long as the total duration of the sequence is at least 25 ms.

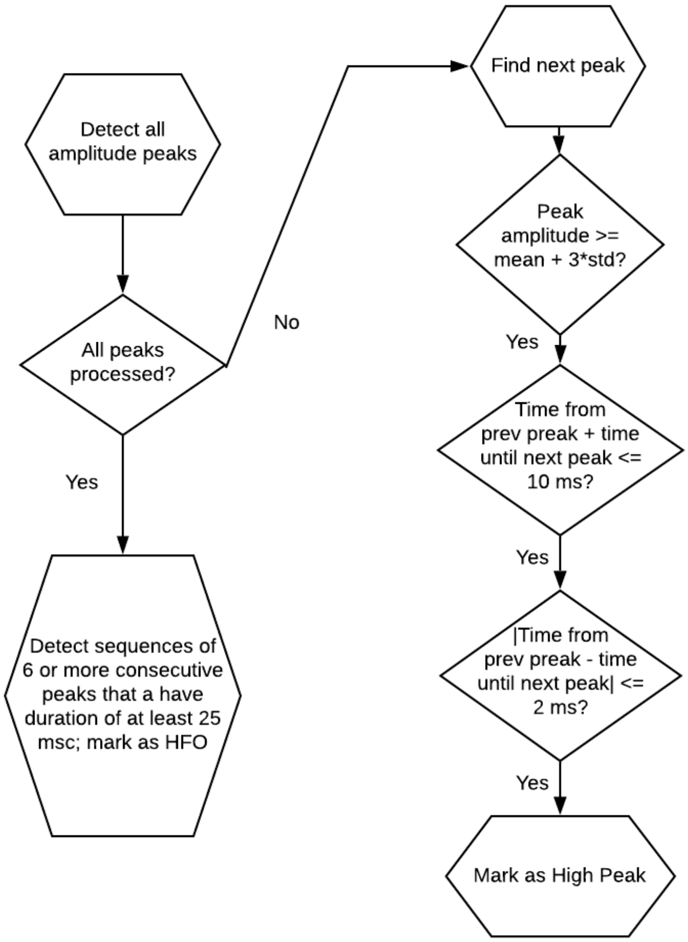

Figure 4: Flowchart of the HFO detection algorithm.

2.5 Training of machine learning model

For the closed-loop seizure prediction model, the LinearDiscriminantAnalysis machine learning module from the Python Scikit-Learn software library was used. This module implements a Linear Discriminant Analysis method for separating statistical events into two groups (i.e., a binary classification method). In this study, these two groups are the occurrence of a seizure at a certain time (group 1) and the absence of a seizure at that time (group 0). While this module's predict method is very fast, the training of this module is very computationally intensive. For this reason, the training was done offline, using previously recorded data from a patient.

For each 500 ms window in the 24 minute patient data file, a coherence value and a number of HFOs value (both aggregated over the last 10 windows) were fed into the

LinearDiscriminantAnalysis module, along with a 0 or 1 “label” to indicate whether a seizure had occurred for this patient within 35 seconds after the start of the current window.

2.6 Simulation of real-time MEG data flow

In order to simulate a real-time stream of MEG data, a Python program was developed to read the same 24-minute MEG recording that was used for the training step described in Section 3.5. Every five seconds, this program generated a new MEG file, containing only the next five seconds of data (i.e., ten 500 ms time windows). Thus, the real-time replay of MEG data took 24 minutes and produced 288 MEG files.

2.7 Processing of MEG data in real-time

In order to process MEG data in real-time, the appearance of every new MEG file was automatically and efficiently detected via the Python watchdog package, which provides a programming interface for monitoring any events that occur in a specified directory. The contents of each discovered file were read into memory. The closed-loop prediction model required at least three time windows to be available in memory in order to process a given time window of data: because of the shifting requirement in the computation of coherence, we needed access to the data that arrived up to 500 ms after the start of a current time window, as well as to the data that arrived 100 ms before the start of a current time window. Thus, the previous, current, and next time windows of data were required in order to compute the coherence, along with the number of HFO events, for the current window. This is depicted in the figure below.

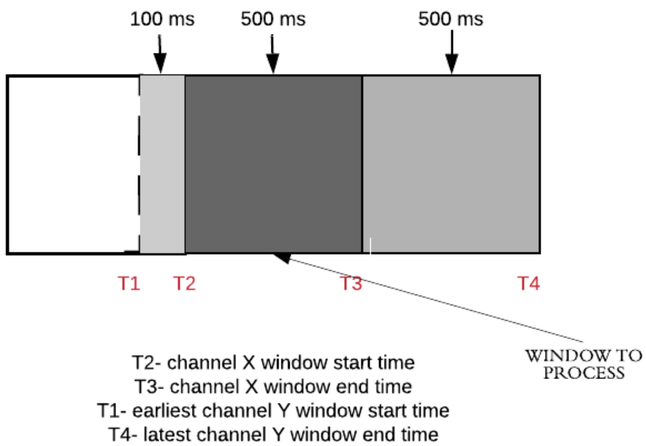

Figure 5: Due to the shifting requirement, the previous and the next window must be available in order to compute the coherence for the window in the middle. The gray areas in the figure represent the time intervals during which data from channel Y must be available in order to compute the coherence between channels X and Y for a 500 ms time window of channel X.

2.8 Parallelization of computations across CPUs

Despite the highly optimized implementation of coherence described in 3.3, a single computer core is not sufficient for the processing of coherence in real-time. The computation must be distributed across cores. The Pool module from the Python multiprocessing package was used for this purpose. Initial attempts to parallelize processing by channel failed because coherence computations require a lot of patient data and per-channel state data, and this data has to be serialized and sent to the worker processes created by the Pool module. For this reason, the Pool module was not able to process more than 30 per-channel requests per second, which is not enough for MEG data with 102 channels. To work around this limitation, the processing was parallelized across time windows. Each worker process handled all of the computations for a given time window.

As a time window requires 70K coherence computations, and as such a computation, after the optimizations described in 3.3, takes about 20 ms, the results for each time window come with a latency between 1.5 and 2 seconds.

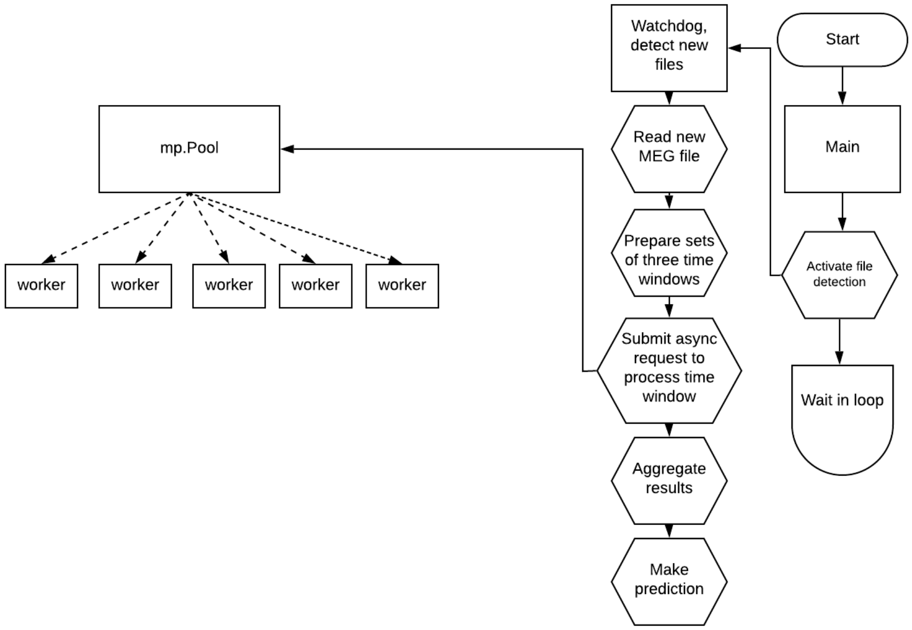

Figure 6. Process diagram and high-level flow chart for the closed loop prediction module.

2.9 Aggregation of results across channels and across time windows, with subsequent generation of seizure prediction

Each worker process mentioned in 3.8 computes a channel-by-channel coherence matrix and a per-channel number of HFOs for each 500 ms window of the 24 minute neural data file. For the purpose of prediction by the machine learning module, both the coherence and number of HFOs are aggregated (averaged, in the current implementation) across channels and then across the latest ten time windows (the current time window and the nine directly preceding ones), producing one number for coherence and one number for the amount of HFOs, for the current time window (500 ms). These numbers are, in turn, fed into the pre-trained LinearDiscriminantAnalysis machine learning module (as described in 3.5), which makes a prediction about whether or not a seizure will occur within 35 seconds after the start time of the current window.

This flow of processing is demonstrated by the figure below:

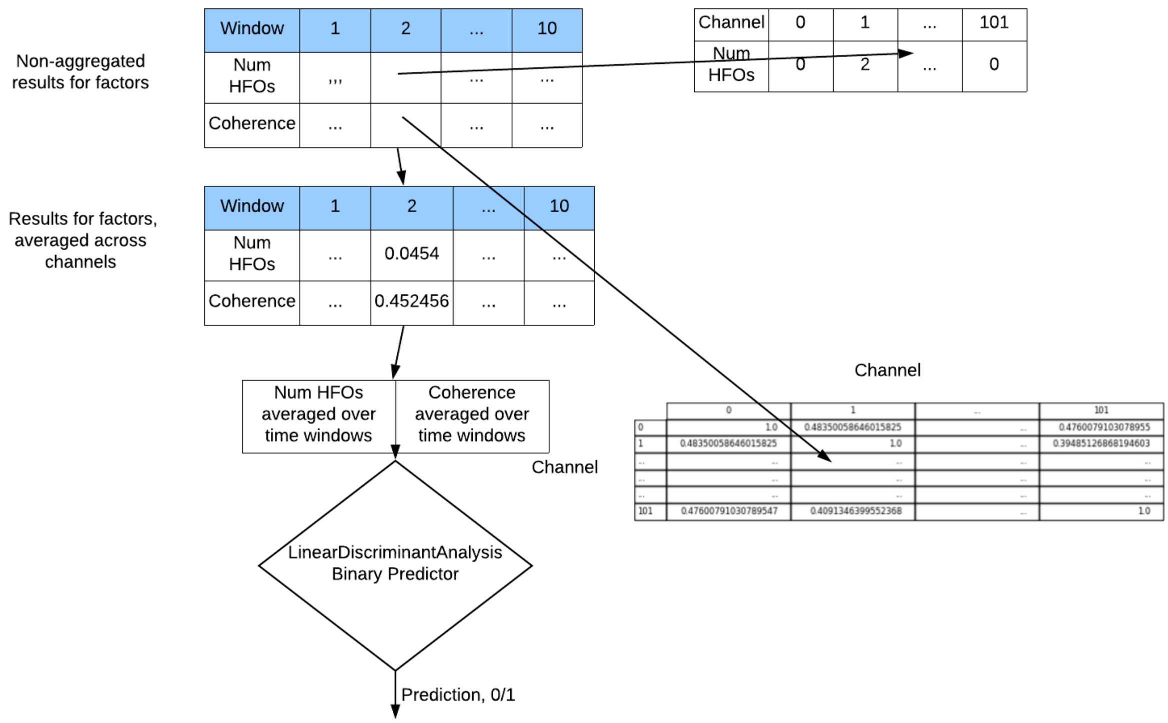

Figure 7: The aggregation of results across channels and across time windows, with the subsequent generation of prediction.

2.10 Computation of inter-channel coherence for different parameter sets

In order to observe the sensitivity of computed coherence to the values of parameters involved in its computation, coherence between MEG channels was computed for all combinations of values of the following three parameters:

a. Time window - values in range 100 msec to 2000 mec, with step of 100 msec, were used
b. Minumum changes of time shift between a pair of MEG channels - values in range 0% to 50% of duraction of time window, with step of 10%, were used. Value of 0% means no time shift is applied to a pair of channels.
c. Time window overlap within the same MEG channel - values 0, 30%, and 50% of duration of time window were used.

3. Results

3.1 Validation of accuracy

To verify the accuracy of the coherence computations, the channel-by-channel coherence matrices that were computed using the new method, both for static data and in real-time, were compared with the coherence matrices computed for the same MEG file earlier, using scipy. The results fully matched.

To verify the quality of the HFO detection, both a neurologist from the Massachusetts Eye and Ear Infirmary and my mentor for this research reviewed the HFOs detected by this algorithm and found the results to be valid. The figures below provide both an aggregated and a per-channel view into detected HFO events.

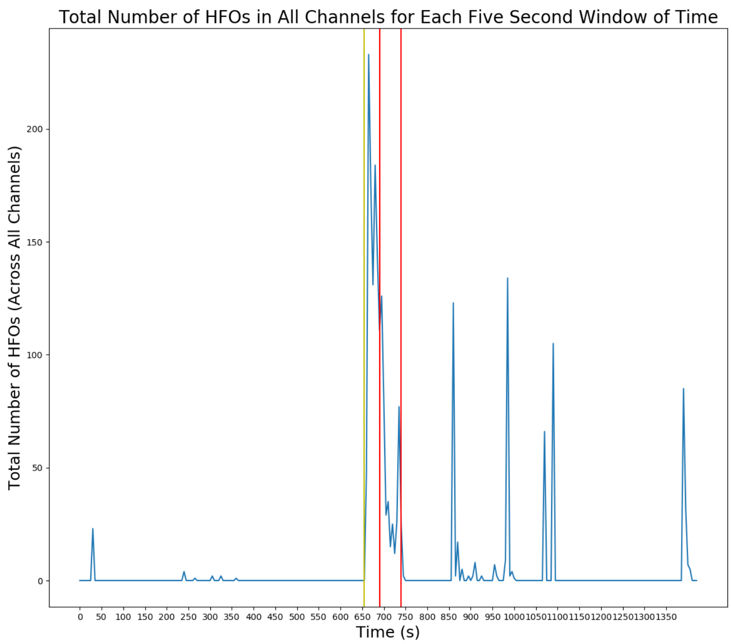

Figure 8: The total number of detected HFOs over time, per five second interval. The peak amount of HFO events per five second interval was reached in the preictal stage (the period just before the seizure), which shows promise in terms of the use of HFOs to predict seizures.

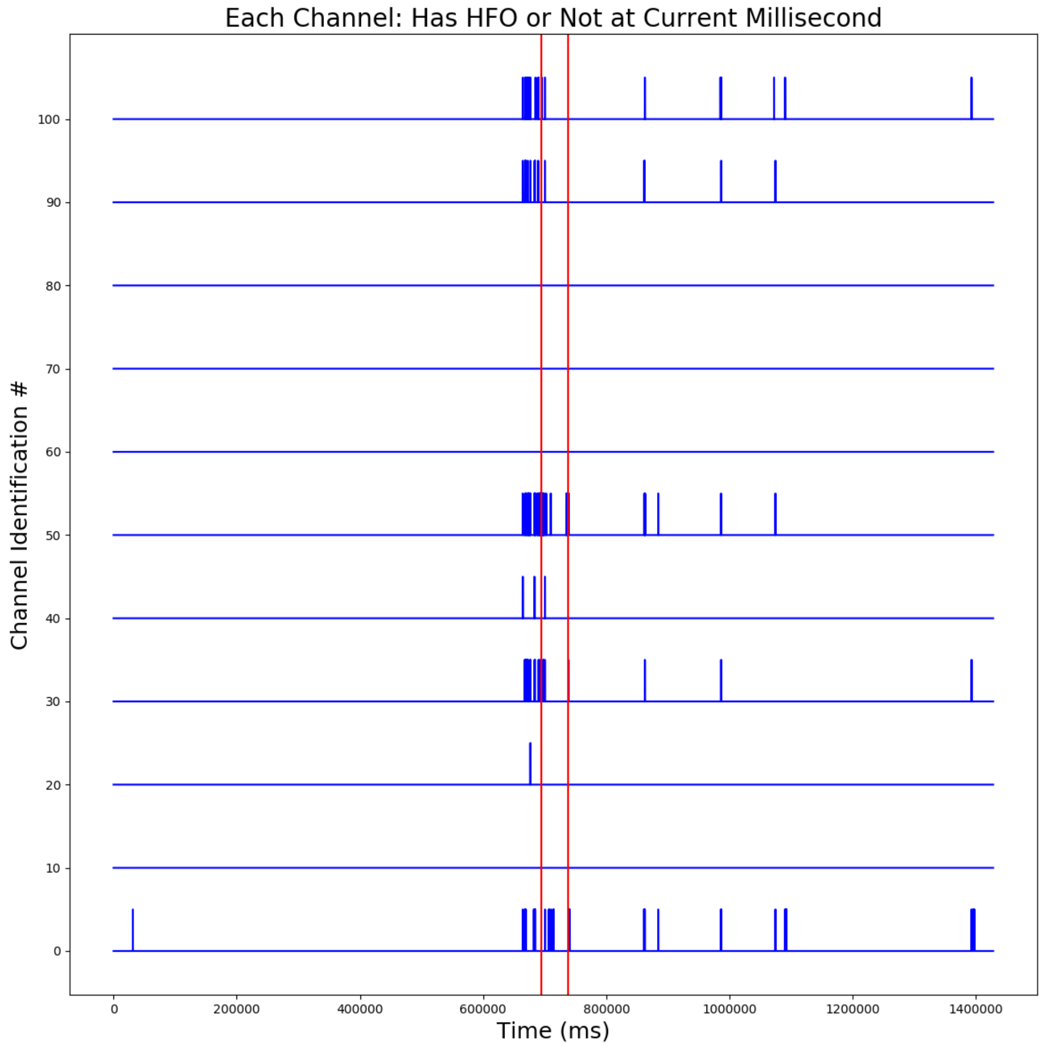

Figure 9: For one of every 10 channels, across time, a vertical blue line is depicted if, at the current milisecond, there is an HFO in progress, and such a line is not depicted if this is not the case. The two red lines represent the start and end times of the seizure.

3.2 Comparison of performance: new algorithm versus traditional method of computing coherence

With the commonly used coherence function from the scipy package, it took 11 hours 4 minutes to process the 24-minute MEG file using all of the cores of the computer on which the measurements were performed. When the new method of computing coherence was used, the same data was processed in 23 minutes 25 seconds, also using all 8 cores of this computer.

This speedup in the processing of coherence makes it much more feasible to experiment with configurable parameters used in the computation of coherence, such as the sizes of time windows and shifting intervals, as well as the minimum and maximum shifting offsets. It also facilitates experimentation with data across multiple patients. Both of these improvements enable research into the optimal use of coherence in seizure prediction.

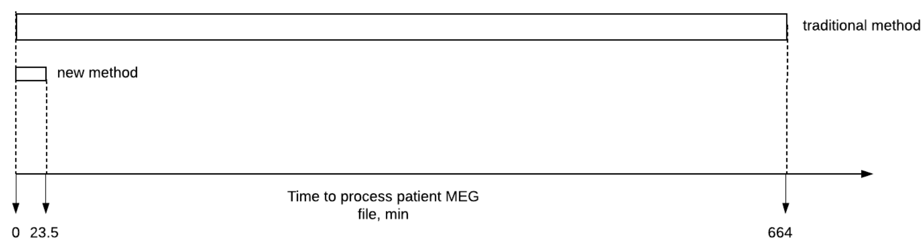

Figure 10: Times taken to compute the coherence for a 24-min MEG patient MEG file with the traditional method versus with the new method proposed in this study.

3.3 Speed of real-time processing, relative to speed of arrival of new data

Making coherence computations about 30 times faster enabled the closed-loop multi-factor prediction model described in this paper to perform with a speed at least as fast as the speed with which the MEG data was recorded across 102 channels. This demonstrates that the new method of computing coherence makes possible real-time seizure predictions that involve the factors that are seen as the most promising today: coherence and HFO events.

3.4 Dependency pattern between inter-channel coherence spike and epileptic seizure

As shown in Figure 11 below, a very large spike in inter-channel coherence, averaged across all MEG channels, was observed 4-7 minutes before the eplileptic seizure, shown as time interval between two vertical red lines. This indicates that the seizure could be predicted in advance, based on computed coherence across MEG channels. However, this finding needs to be verified agaist data for many epileptic seizures, across many patients.

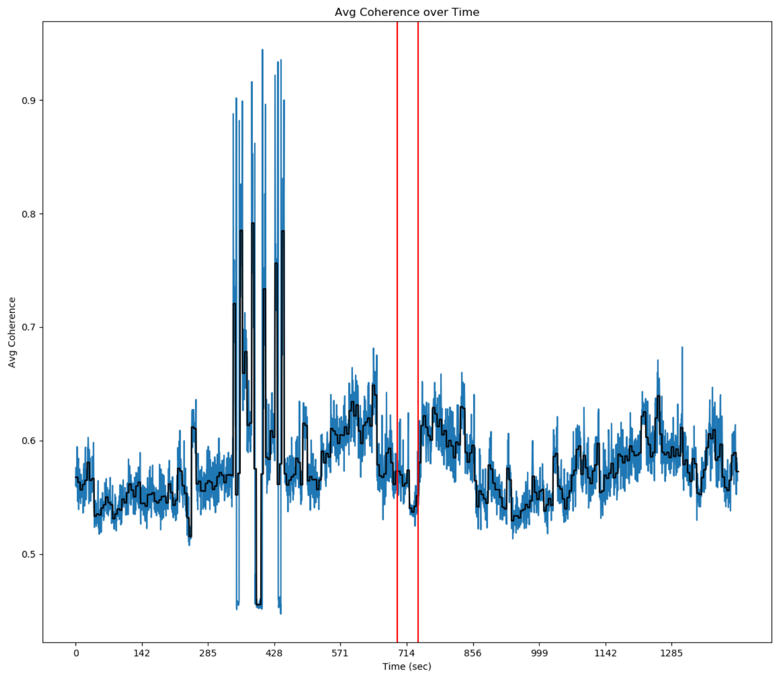

Figure 11. Average inter-channel coherence over time.

Figure 12 below shows number of channel and frequency pairs for which maximum coherence with channel MEG0241 was detected on a particular minute.Two vertical red lines in the figure show time interval during which the seizure occurred. Most channel and frequency pairs had their maximum coherence with channel MEG0241 4-7 minutes before the epileptic seizure. This pattern was observed for all 102 MEG channels that are considered to be relevant to epileptic activity.

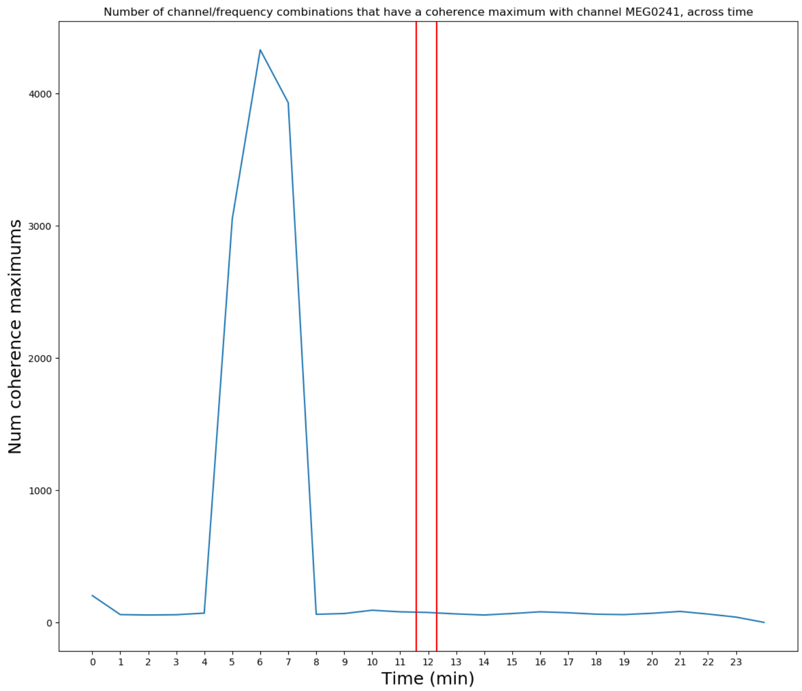

Figure 12. Number of channel/frequency combinations that have a coherence maximum with channel MEG0241, over time.

3.5 Dependency of computed coherence on parameters used in its computation.

In order to see how sensntive computed coherence is to factors used in its computation, the coherence was computed for combinations of time window interval, time window interval overlap, and minimum changes of time shift between a pair of channels, as mentioned in section 3.10.

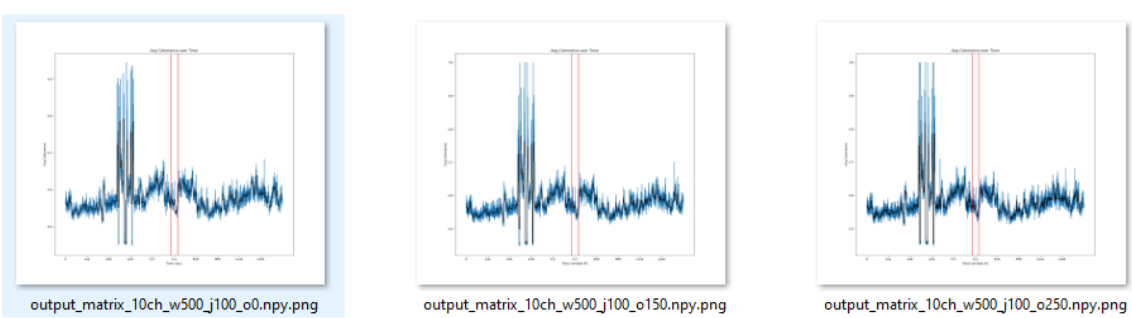

Figure 13. Plots of coherence vs time for time window overlap values 0 msec, 150 msec, and 250 msec

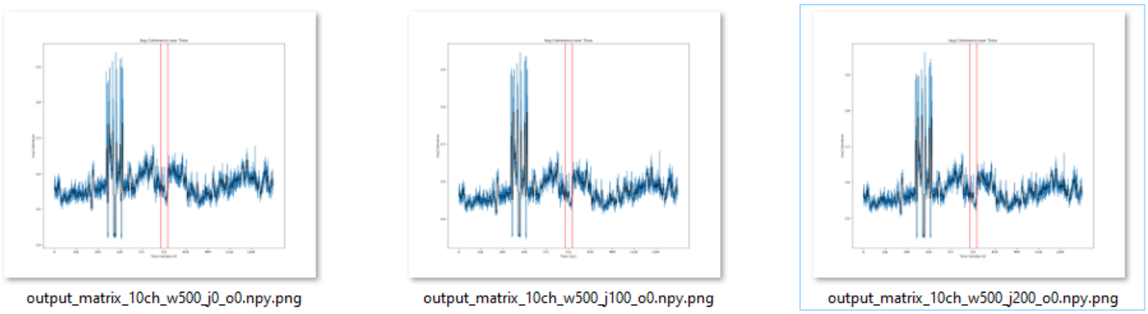

Figure 14. Plots of coherence vs time for minimum changes of time shift between channel pairs 0% (no time shift applied), 20% of time window, and 40% of time window.

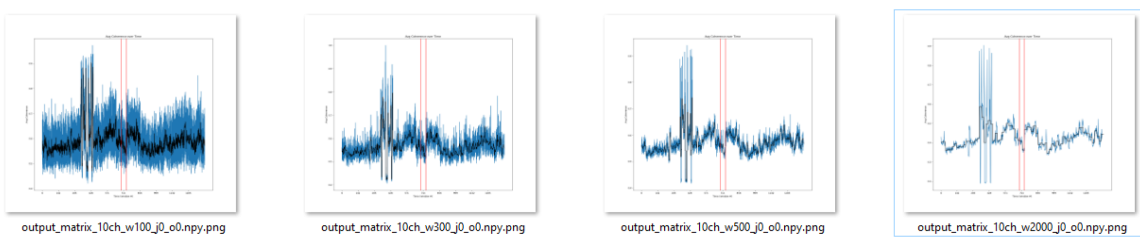

Figure 15. Plots of coherence vs time for time window values 100 msec, 300 msec, 500 msec, and 2000 msec.

Figures 13-15 above show that the graph of average inter-channel coherence vs time was not significantly impacted by those factors. The time window overlap and the minimum changes of time shift between a pair of channels had very small effect on computed coherence. Smaller time window size increased amount of noise in the graph, apparently because Fourier transform, which is the basis of coherence computation, produces more accurate results when more time samples are provided. Since MEG sample rate used in this/ research was fixed at 1000 per second , a smaller time window means fewer samples to analyze per time window.

4. Discussion

4.1 Implications

In this project, I have developed a high-performing computational model for the closed-loop prediction of epileptic seizures from HFO and coherence statistics from MEG data. As this is the first model capable of the live processing of factors necessary for seizure prediction, it provides a basis for later models that accurately predict seizures in real-time.

Additionally, because of its high speed, this model facilitates further experimentation into the optimal use of coherence as a factor for seizure prediction. In this research, a very large surge of inter-channel coherence in the time interval of 4-7 minutes before the epileptic seizure was observed.

Further, alterations in coherence are also tied to the onset of pathological events in movement disorders, such as Parkinson's disease [13] and dystonia [14]. As such, the real-time processing of coherence data can potentially be applied to these diseases to inform their diagnosis and treatment. For example, a machine learning algorithm could be trained to use coherence data to predict the tremor onset in patients with Parkinson's disease, allowing measures to be taken to prevent the manifestation of such symptoms.

4.2 Limitations

The closed-loop multi-factor prediction model described in this paper was only tested on two factors at a time. Future experimentation may show that the algorithm must be altered to adjust for an increased number of factors (in order to continue to maintain a high enough speed).

Also, because of the limitations of the Python multiprocessing package, a given time window of 500 ms has to be processed fully in a single Python process. Due to the need to compute coherence 70K times per time window, this results in a latency between 1.5 and 2 seconds. The replacement of the Python multiprocessing package with a better-performing module should be considered for future work.

Data from only one epileptic patient, with a single epileptic seizure, was used for this research. MEG recordings from more epileptic seizures and from more epileptic patients need to be studied in order to see whether observed surge in coherence several minutes before the epileptic seizure reflects a stable pattern.

4.3 Potential Directions

A natural next step would be to analyze MEG data from more epileptic seizures, from multiple patients, in order to try to identify a common pattern that ties changes in coherence to the seizures.

Since it is likely that coherence alone will be found not sufficient to accurately predict approaching seizure, it will be important to investigate which other factors should be considered in terms of seizure prediction. If necessary, I plan to adjust the model so that it computes a larger number of various factors with adequate speed. The need to compute additional factors for the prediction model may require the model to scale not only across all cores of one computer, but across a cluster of computers. Thus, the model could benefit from the use of Python packages such as ParallelPython, which supports clusters of computers.

For any factors changes in which accompany epileptic seizures, I plan to attempt to discern and analyze patterns that may be used to forecast the seizure, using both statistics and visual analysis.

Yet another task to consider could be an investigation into which machine learning algorithms would be optimal for real-time seizure prediction from MEG data.

5 References

1. Broccard FD, 1 Mullen T, Chi YM, Peterson D, Iversen JR, Arnold M, Kreutz-Delgado K, Jung T-P, Makeig S, Poizner H, Sejnowski T, Cauwenberghs G. Closed-loop Brain-Machine-Body Interfaces for Noninvasive Rehabilitation of Movement Disorders, Ann Biomed Eng. 2014, vol. 42(8) (pg. 1573-1593).

2. Marsden, J, Limousin-Dowsey P, Fraix, V, Pollak, P, Odin, P, Brown, P. Intermuscular coherence in Parkinson's disease: effects of subthalamic nucleus stimulation, NeuroReport 2001, vol. 12 (pg. 1113-1117)

3. Zrenner C, Belardinelli P, Müller-Dahlhaus F, Ziemann U. Closed-Loop Neuroscience and Non-Invasive Brain Stimulation: A Tale of Two Loops, J Clin Neurophysiol. 2016, vol. 10 (pg. 92).

4. Nagaraj V, Lee S, Krook-Magnuson E, Soltesz I, Benquet P, Irazoqui P, Netoff T. The Future of Seizure Prediction and Intervention: Closing the loop, J Clin Neurophysiol. 2015, vol. 32(3) (pg. 194-206).

5. Sergey Burnos, Peter Hilfiker, Oğuzkan Sürücü, Felix Scholkmann, Niklaus Krayenbühl, Thomas Grunwald, Johannes Sarnthein. Human Intracranial High Frequency Oscillations (HFOs) Detected by Automatic Time-Frequency Analysis, PLoS ONE, April 10, 2014

6. Urrestarazu E, Chander R, Dubeau F, Gotman J. Interictal high-frequency oscillations (100–500 Hz) in the intracerebral EEG of epileptic patients, Brain 2007, vol. 130 (pg. 2354–2366).

7. Höller P, Trinka E, Höller Y. High-Frequency Oscillations in the Scalp Electroencephalogram: Mission Impossible without Computational Intelligence, Computational Intelligence and Neuroscience 2018, Article ID 1638097.

8. Abd El-Samie FE, Alotaiby TN, Khalid M, Alshebeili SA. A Review of EEG and MEG Epileptic Spike Detection Algorithms, IEEE Access 2018.

9. Jirsch JD, Urrestarazu E, LeVan P, Olivier A, Dubeau F, Gotman J. High-frequency oscillations during human focal seizures, Brain 2006, vol. 129 (pg. 1593–1608).

10. M. Bandarabadi M, Rasekhi J, Teixeira CA, Dourado A. Sub-band Mean Phase Coherence for Automated Epileptic Seizure Detection, The International Conference on Health Informatics 2014, vol.42 (pg. 319-322).

11. D.K. Ravish DK, Shenbaga Devi S, Krishnamoorthy SG. Wavelet analysis of EEG for seizure detection: Coherence and phase synchrony estimation, Biomedical Research 2015, vol. 26, issue 3.

12. Aggarwal G, Gandhi TK, Prediction of Epileptic Seizures based on Mean Phase Coherence, bioRxiv 2017.

13. Airaksinen K, Jyrki M, Nurminen J, Luoma J, Taulu S, Ahonen A, Pekkonen E . Cortico-muscular coherence in advanced Parkinson’s disease with deep brain stimulation, Clinical Neurophysiology 2015, vol. 126 (pg. 748-755)

14. McClelland V, Cvetkovic Z, Lin JP, Mills K. Corticomuscular coherence in childhood dystonia, Clinical Neurophysiology 2018, vol. 129 (pg. 167)

15. Miguel C. Soriano MC, Niso G, Clements J, Ortín S, Carrasco S, Gudín M, Mirasso CR, Pereda E. Automated Detection of Epileptic Biomarkers in Resting-State Interictal MEG Data, Front Neuroinform. 2017, vol. 11 (pg. 43).

16. Space Petkoski, Matias Palva, Viktor Jirsa. Phase-lags in large-scale brain synchronization: Methodological considerations and in-silico analysis. PLOS Computational Biology, July 10, 2018