

8-2018

Identification of the Seizure Onset Zone by Auto-Regressive Model Residual Modulation Applied to Intracranial EEG and its Correlation to Channels with High Preponderances of Detected HFOs

Allison L. Rogutich
Grand Valley State University

Follow this and additional works at: <https://scholarworks.gvsu.edu/theses>



Part of the [Biomedical Engineering and Bioengineering Commons](#)

ScholarWorks Citation

Rogutich, Allison L., "Identification of the Seizure Onset Zone by Auto-Regressive Model Residual Modulation Applied to Intracranial EEG and its Correlation to Channels with High Preponderances of Detected HFOs" (2018). *Masters Theses*. 903.
<https://scholarworks.gvsu.edu/theses/903>

This Thesis is brought to you for free and open access by the Graduate Research and Creative Practice at ScholarWorks@GVSU. It has been accepted for inclusion in Masters Theses by an authorized administrator of ScholarWorks@GVSU. For more information, please contact scholarworks@gvsu.edu.

Identification of the Seizure Onset Zone by Auto-Regressive Model Residual Modulation Applied to Intracranial EEG and its Correlation to Channels with High Preponderances of Detected HFOs

Allison Lea Rogutich

A Thesis Submitted to the Graduate Faculty of

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degrees of

Master of Science in Engineering, Biomedical Engineering

School of Engineering

August 2018

Acknowledgements

I would first like to thank my advisor and thesis committee chair, Dr. Bossemeyer, for his consistent guidance and expertise provided on this thesis. I whole-heartedly appreciate the time he spent meeting regularly with me, always delivering meaningful feedback on my work, and ensuring a positive direction of the thesis. In addition, he has inspired me greatly to pursue a career in embedded systems design through the fascinating and challenging classes he teaches, and I truly credit him for preparing me so well for a highly technical and rewarding career.

I would also like to thank my thesis committee members Dr. Rhodes and Dr. Fishback for their valued perspectives and input on my work. Their expertise and continuous involvement throughout the development of this thesis has been key to delivering meaningful results. Additionally, Dr. Rhodes encouraged me to pursue a M.S.E. since I had first shown interest in Biomedical Engineering. I thank her for impacting me with curiosity and a motivation to develop my knowledge further. Learning specialized engineering techniques and gaining a strong biomedical background from the classes she teaches in this program is an opportunity I will forever cherish. I owe many thanks to Dr. Choudhuri, the graduate program director, for always supporting me throughout my time in the engineering program.

Thank you to Alison Dutkiewicz BSN, RN, CCRC for coordinating the anonymization and transfer of iEEG data between Spectrum Health and GVSU. I very greatly appreciate Dr. Konstantin Elisevich, Dr. Mohamad Ayman Haykal, and Dr. David Burdette at Spectrum Health for their invaluable clinical insights on this thesis and for taking time to meet with us on multiple occasions. I greatly thank Dr. Elisevich for allowing me to shadow him on his surgeries and learn about the procedures from first hand observation; experiences I won't forget!

I would like to thank my jumps coach Steve Jones, and head track coach Jerry Baltes, for their support and encouragement of my academic studies during my time as a student-athlete. Coach Jerry, thank you for the incredible opportunity to compete in triple jump at the college level, and for creating a program that fosters athletic, academic, and personal success. Coach Steve, thank you for always working with me on balancing training and classes. I always looked forward to going to practice; I owe much of my success as a student to you.

Finally, I want to thank my parents for their never-ending love and support, and I especially appreciate the values they instilled in me at a young age. I owe all my success to them.

Abstract

The objective of this thesis was to examine the ability of the Autoregressive Model Residual Modulation (ARRm) method to identify the Seizure Onset Zone (SOZ) in intracranial electroencephalogram (iEEG) of patients with refractory epilepsy. Patients who have not become seizure free after multiple trials of antiepileptic drugs (AEDs) may seek treatment through epilepsy surgery. Cortical electrodes are implanted directly on the cerebral cortex, then iEEG is collected. A specialized neurologist reviews the iEEG, then in consultation with the neurosurgeon, the SOZ is determined and areas of the brain may be chosen for resection. The success rate of epilepsy surgery varies, so it is apparent that identifying exactly where to resect epileptic tissue is still very challenging.

In recent research, High Frequency Oscillations (HFOs) in iEEG have shown strong relations to epileptic tissue. Automated HFO detection methods have been developed, but most involve analysis in the frequency domain and are computationally expensive. The ARRm method is implemented in the time domain and has potential to be implemented for real-time analysis. AR modeling is used to predict the iEEG and should not be able to accurately model highly nonharmonic events such as HFOs. Using a coefficient of variation involving the residuals of the model (ARRm value), interpretation of results showed that high ARRm values also corresponded to channel locations with high HFO counts, which included channels of interest identified by the epileptologist. Statistically significant ($p < 0.01$) correlations were drawn between the ARRm value and HFO counts on channels of interest. Examination of the AR model residual during epileptogenic events revealed that the residual was highest when spikes and/or Fast Ripples occurred. These findings suggest that significantly correlated channels may indicate the presence of fast ripples or spikes occurring with fast ripples in the signal.

Table of Contents

Acknowledgements	3
Abstract	5
List of Tables	9
List of Figures.....	10
Abbreviations	16
Chapter 1: Introduction.....	17
Introduction	18
Purpose.....	19
Scope.....	20
Chapter 2: Review of Literature.....	22
1. Anatomy & Physiology of the Brain.....	23
1.1 Anatomy.....	23
1.1.1 The Cerebrum.....	24
1.1.2 Neurons.....	27
1.2 Physiology.....	28
2. Electroencephalography (EEG).....	30
2.1. Origins of Neural Bioelectric Potentials.....	30
2.2. Electrodes.....	32
3. Epilepsy.....	34
3.1 Etiology.....	34
3.2 Refractory Epilepsy.....	35
3.2.1 Electrode Implantation Surgery.....	35
3.2.2 Identifying the SOZ.....	38
3.2.3 High Frequency Oscillations	42
3.2.4 Epilepsy Surgery.....	44
4. HFO Detection Methods.....	46
4.1 Modified Burnos Method.....	46
4.2 MNI Detector.....	50
5. ARRM Method.....	51
Chapter 3: Methodology.....	58
6. Modified Burnos HFO Detection Method in RIPPLELAB.....	59
7. The ARRM Method in MATLAB.....	60
8. HFO/ARRM Analysis Tool.....	62

8.1	User Interface	62
8.2	Analysis.....	64
9.	Statistical Analysis	69
10.	Real-Time ARRm Method on Embedded Platform	71
Chapter 4: Results.....		74
11.	Data.....	75
12.	ARRm Value and HFO Counts Over Time.....	76
12.1.	Patient WDH-022.....	76
12.1.1	ARRm Value.....	79
12.1.2	HFO Count.....	82
12.2.	Patient DJA-033.....	85
12.2.1	ARRm Value.....	87
12.2.2	HFO Count.....	89
12.3.	Patient JMM-025.....	91
12.3.1	ARRm Value.....	94
12.3.2	HFO Count.....	98
13.	Median ARRm and Total HFO Count Plots.....	103
13.1	Patient WDH-022.....	103
13.2	Patient DJA-033.....	105
13.3	Patient JMM-025.....	107
14.	Correlation Between ARRm Value and HFO Count.....	109
14.1	HFO Detection EOI Threshold	110
14.2	Statistical Correlations	117
14.2.1	Patient WDH-022.....	118
14.2.2	Patient DJA-033.....	122
14.2.3	Patient JMM-025.....	123
Chapter 5: Discussion and Conclusions		135
15.	Discussion	136
16.	Conclusion	139
Appendices.....		141
ARRm/HFO Median/Total HFO Count Plots		142
Patient WDH-022.....		142
Patient DJA-033.....		144
Patient JMM-025.....		146
MATLAB Code.....		150

RIPPLELAB Implementation of Modified Burnos Method.....	150
ARRm Method.....	168
ARRm/HFO Analysis Tool.....	174
Statistical Calculations	191
Real-Time EEG Acquirement/Analysis System Software Design Document.....	193
References.....	211

List of Tables

Table 1: Brain Lobe Functions	24
Table 2: Terminology for Positions of Anatomical Features of the Brain	24
Table 3: Patients and Data Files from Spectrum Health EMU Analyzed using ARRM and HFO Detection	75
Table 4: Patient WDH-022 Electrode Placement.....	77
Table 5: Patient WDH-022 Seizure Times, Notes and Files.....	77
Table 6: Patient WDH-022 Channels of Interest Identified by Epileptologist.....	79
Table 7: Patient DJA-033 Electrode Placement.....	85
Table 8: Patient DJA-033 Seizure Times, Notes and Files	86
Table 9: Patient DJA-033 Channels of Identified by Epileptologist.....	87
Table 10: Patient JMM-025 Electrode Placement	92
Table 11: Patient JMM-025 Seizure Times, Notes and Files.....	93
Table 12: Patient JMM-025 Channels of Identified by Epileptologist	93
Table 13: Chosen Optimal HFO Detection EOI Thresholds for Each Patient.....	111

List of Figures

Figure 1: Diagram of the Cerebral Brain Lobes [3]	23
Figure 2: Layers of the Structures Enclosing the Cerebrum.....	24
Figure 3: The Cerebrum (Median View) [2]	25
Figure 4: The Cerebrum (Lateral View) [2]	26
Figure 5: Anatomy of a Neuron [3]	28
Figure 6: The Synapse [3]	28
Figure 7: Membrane Potential of a Neuron [3]	29
Figure 8: Pyramidal Neuron Schematic and Influencers of PSPs [3]	31
Figure 9: Schematic of Dipoles Related to the Scalp Surface	31
Figure 10: Cortical Electrode Grid and Leads.....	32
Figure 11: Cortical Electrode Strip and Lead.....	33
Figure 12: Depth Electrode and Lead.....	33
Figure 13: Surface Electrode Grid Implantation onto the Cerebral Cortex [1]	36
Figure 14: ROSA® stereo EEG Robot Allows for Accurate and Steady Placement of Depth Electrodes.....	37
Figure 15: Example of Electrode Mapping Schematic	38
Figure 16: Epileptiform Spike Patterns on Bipolar EEG Recordings [1].....	41
Figure 17: (A) HFO Ripple and (B) HFO Fast Ripple	42
Figure 18: (A) NeuroPace RNS. (B) Schematic of Implanted Neuropace. The RNS device (1) is attached to a depth electrode (2) and/or a cortical electrode strip (3) [21]	45
Figure 19: Raw iEEG Time Segment with HFO & Filtered Segment with Envelope (red) and Threshold.....	47
Figure 20: Frequency Content & PSD Plot of the Stockwell Transform of Time Segment with HFO	48
Figure 21: Example of iEEG with Sharp Artifact	49
Figure 22: ARRM Algorithm Processing Steps [27]	54

Figure 23: Performance of the Original and Improved ARRM methods in Identifying FRs [11].....	56
Figure 24: Patient Example of ARRM Results.....	57
Figure 25: ARRM Method Process.....	60
Figure 26: 3rd Order Model Fit (blue) to Original iEEG Signal (black) and Residual of the Model	61
Figure 27: Example of Definition and Selection of Channels of Interest	62
Figure 28: HFO/ARRM Analysis Overall User Interface	63
Figure 29: ARRM/HFO Value Over Time for Selected Channel.....	64
Figure 30: Boxplots of ARRM/HFO Values for each Channel.....	65
Figure 31: Scatterplot of ARRM Values vs. HFO Count Values	66
Figure 32: Total HFO Count/Median ARRM Value Plots.....	67
Figure 33: Median HFO Count/Median ARRM Value Plots.....	68
Figure 34: ARRM Residuals Plot	69
Figure 35: Hardware Block Diagram for Real-Time EEG Collection/ARRM Analysis System	71
Figure 36: Software State Diagram for Real-Time EEG Collection/ARRM Analysis System.....	72
Figure 37: Modeled EEG Signal (Blue) and Original EEG Signal (Black).....	72
Figure 38: ECG Signal, Modeled Signal, and Residual Signal Variation.....	73
Figure 39: ARRM - Sequential Files for the Beginning of EMU Stay (10/22/2014) and Channels of Interest.....	80
Figure 40: ARRM - Sequential Files for the Middle of EMU Stay (10/26/2014) and Channels of Interest.....	81
Figure 41: ARRM - Sequential Files for the End of EMU stay (10/28/2014) and Channels of Interest.....	82
Figure 42: HFO - Sequential Files for the Beginning of EMU Stay (10/22/2014) and Channels of Interest.....	83
Figure 43: HFO- Sequential Files for the Middle of EMU Stay (10/26/2014) and Channels of Interest.....	84
Figure 44: HFO - Sequential Files for the End of EMU stay (10/28/2014) and Channels of Interest	85

Figure 45: ARRm - Sequential Files for the Beginning of EMU Stay (11/04/2015) and Channels of Interest.....	87
Figure 46: ARRm - Sequential Files for the Middle of EMU Stay (11/06/2015) and Channels of Interest.....	88
Figure 47: ARRm - Sequential Files for the End of EMU stay (11/08/2015) and Channels of Interest.....	89
Figure 48: HFO- Sequential Files for the Beginning of EMU Stay (11/04/2015) and Channels of Interest.....	90
Figure 49: HFO- Sequential Files for the Middle of EMU Stay (11/06/2015) and Channels of Interest.....	90
Figure 50: HFO - Sequential Files for the End of EMU stay (11/08/2015) and Channels of Interest	91
Figure 51: ARRm – First Group of Sequential Files for the Beginning of EMU Stay (06/11/2015) and Channels of Interest	94
Figure 52: ARRm – Second Group of Sequential Files for the Beginning of EMU Stay (06/13/2015) and Channels of Interest	95
Figure 53: ARRm – First Group of Sequential Files for the Middle of EMU Stay (06/14/2015) and Channels of Interest.....	96
Figure 54: ARRm – Second Group of Sequential Files for the Middle of EMU Stay (06/16/2015) and Channels of Interest	97
Figure 55: ARRm – Sequential Files for the End of EMU Stay (06/20/2015) and Channels of Interest.....	98
Figure 56: HFO – First Group of Sequential Files for the Beginning of EMU Stay (06/11/2015) and Channels of Interest.....	99
Figure 57: HFO – Second Group of Sequential Files for the Beginning of EMU Stay (06/13/2015) and Channels of Interest	100
Figure 58: HFO – First Group of Sequential Files for the Middle of EMU Stay (06/14/2015) and Channels of Interest.....	101
Figure 59: HFO – Second Group of Sequential Files for the Middle of EMU Stay (06/16/2015) and Channels of Interest.....	102
Figure 60: HFO – Sequential Files for the End of EMU Stay (06/20/2015) and Channels of Interest	103

Figure 61: ARRm Value and HFO Count Over Time from the Beginning of EMU Stay – Patient WDH-022.....	104
Figure 62: Median ARRm Value and Total HFO Count from the Beginning of EMU Stay – Patient WDH-022.....	105
Figure 63: ARRm Value and HFO Count Over Time from the Beginning of EMU Stay – Patient DJA-033.....	106
Figure 64: Median ARRm Value and Total HFO Count from the Beginning of EMU Stay – Patient DJA-033.....	107
Figure 65: ARRm Value and HFO Count Over Time from First Group of Files at the Beginning of EMU Stay – Patient JMM-025.....	108
Figure 66: Median ARRm Value and Total HFO Count from the First Group of Files at the Beginning of EMU Stay – Patient JMM-025.....	109
Figure 67: File BA2681LD with Optimal EOI Threshold	111
Figure 68: File BA2681LF with Optimal EOI Threshold	112
Figure 69: Patient DJA-033 Optimal Threshold Tests.....	113
Figure 70: File AA536176 with Optimal EOI Threshold.....	114
Figure 71: File AA5320YE and AA5320YF With Original Thresholding.....	115
Figure 72: File AA5320YE With Optimal EOI Threshold	116
Figure 73: File AA5320YF With Optimal EOI Threshold.....	117
Figure 74: Correlation Test – Patient WDH-022, File BA2681LD.....	118
Figure 75: Spearman and Pearson Correlation Results - Patient WDH-022, File BA2681LD.....	119
Figure 76: Correlation Test – Patient WDH-022, File BA2681LF	120
Figure 77: Spearman and Pearson Correlation Results - Patient WDH-022, File BA2681LF	121
Figure 78: Correlation Test – Patient DJA-033, File AA536176.....	122
Figure 79: Spearman and Pearson Correlation Results - Patient DJA-033, File AA536176	123
Figure 80: Correlation Test – Patient JMM-025, File AA5320YE.....	124
Figure 81: Spearman and Pearson Correlation Results - Patient JMM-025, File AA5320YE.....	125
Figure 82: Correlation Test – Patient JMM-025, File AA5320YF.....	126

Figure 83: Spearman and Pearson Correlation Results - Patient JMM-025, File AA5320YF	127
Figure 84: Multiple HFOs	129
Figure 85: HFO with High Levels of High Frequency Energy > 250Hz, Possibly Classified as a Fast Ripple.....	130
Figure 86: Location of Fast Ripple from Figure 85 in Raw iEEG and Channel TB3 filtered for >80Hz	131
Figure 87: Epileptic Spikes and HFO	132
Figure 88: Sharp Spike and HFO.....	133
Figure 89: Epileptic Discharge and HFO	134
Figure 90: ARRm Value and HFO Count Over Time from the Middle of EMU Stay – Patient WDH-022.....	142
Figure 91: Median ARRm Value and Total HFO Count from the Middle of EMU Stay – Patient WDH-022.....	142
Figure 92: ARRm Value and HFO Count Over Time from the End of EMU Stay – Patient WDH-022.....	143
Figure 93: Median ARRm Value and Total HFO Count from the End of EMU Stay – Patient WDH-022.....	143
Figure 94: ARRm Value and HFO Count Over Time from the Middle of EMU Stay – Patient DJA-033.....	144
Figure 95: Median ARRm Value and Total HFO Count from the Middle of EMU Stay – Patient DJA-033.....	144
Figure 96: ARRm Value and HFO Count Over Time from the End of EMU Stay – Patient DJA-033	145
Figure 97: Median ARRm Value and Total HFO Count from the End of EMU Stay – Patient DJA-033.....	145
Figure 98: ARRm Value and HFO Count Over Time from Second Group of Files at the Beginning of EMU Stay – Patient JMM-025	146
Figure 99: Median ARRm Value and Total HFO Count from the Second Group of Files at the Beginning of EMU Stay – Patient JMM-025.....	146
Figure 100: ARRm Value and HFO Count Over Time from First Group of Files at the Middle of EMU Stay – Patient JMM-025.....	147

Figure 101: Median ARRM Value and Total HFO Count from the First Group of Files at the Middle of EMU Stay – Patient JMM-025	147
Figure 102: ARRM Value and HFO Count Over Time from Second Group of Files at the Middle of EMU Stay – Patient JMM-025.....	148
Figure 103: Median ARRM Value and Total HFO Count from the Second Group of Files at the Middle of EMU Stay – Patient JMM-025.....	148
Figure 104: ARRM Value and HFO Count Over Time from the End of EMU Stay – Patient JMM-025.....	149
Figure 105: Median ARRM Value and Total HFO Count from the End of EMU Stay – Patient JMM-025.....	149

Abbreviations

HFO: High Frequency Oscillation

EMU: Epilepsy Monitoring Unit

AED: Antiepileptic Drug

EEG: Electroencephalogram

iEEG: Intracranial Electroencephalogram

ECoG: Electrocorticogram

SOZ: Seizure Onset Zone

EZ: Epileptogenic Zone

ARRm: Auto-Regressive Residual Modulation

CSF: Cerebral Spinal Fluid

IED: Interictal Epileptiform Discharge

ILAE: International League Against Epilepsy

MRI: Magnetic Resonance Imaging

CT: Computed Tomography

AR: Auto-Regressive

EOI: Events of Interest

COI: Channels of Interest

A/D: Analog to Digital

TI: Texas Instruments

SPI: Serial Peripheral Interface

UART: Universal Asynchronous Receiver-Transmitter

Chapter 1: Introduction

Introduction

Epilepsy is a neurological brain disorder which results in unprovoked seizures. Patients who are not seizure free after multiple trials of antiepileptic drugs are considered to have refractory epilepsy, and can seek treatment through epilepsy surgery. Electrodes are implanted directly onto the brain, and then intracranial electroencephalogram (iEEG) is recorded as patients stay in the Epilepsy Monitoring Unit (EMU). The EMU is a specialized hospital unit where studies are conducted on patients with epilepsy to reach a treatment plan, ranging from brain imaging, functional brain imaging, scalp EEG, and iEEG. Current clinical methods for identifying the seizure onset zone (SOZ) solely involve visual inspection of the iEEG signals. A specialized neurologist, an epileptologist, views the iEEG, looking to identify spikes in the signals recorded between seizures and other characteristic signatures related seizures. After extensive review of the iEEG, channels of interest are chosen that exhibit these signatures, and after discussion with the neurosurgeon, tissue locations to resect are determined from electrode placements.

There is a movement to develop automated detection algorithms that can identify the channels of interest, to expedite the current process for clinical evaluation of refractory epilepsy patients. Recent research has shown that High Frequency Oscillations (HFOs), iEEG ripples between 80 and 500Hz, are associated with epileptic tissue [1]. Multiple time-frequency algorithms exist that can detect HFOs. These methods are used for post-processing the iEEG, and take multiple hours to run over signals sampled at 1000-2000Hz. Since these techniques are computationally expensive, they are unsuitable for real-time analysis. Autoregressive model residual modulation (ARRm) is a method that can be implemented in the time domain to detect HFO activity. This method also has potential to be implemented in real time. The objective of this thesis was to examine the ability of the ARRm method to identify the SOZ and compare results with a time-frequency HFO detection method.

Purpose

The purpose of this thesis was to investigate identification of the SOZ by auto-regressive model residual modulation (ARRm) applied to long-term recordings of intracranial EEG, and explore if there is a correlation between high measures of nonharmonicity in the signal and electrodes identified by the epileptologist exhibiting characteristics that could initiate a seizure. Previous studies have shown that electrode recording channels with high rates of HFO occurrences were channels marked by the epileptologist. The main research objectives and expected outcomes are as follows:

- Further validate HFO detection using time-frequency analysis, with epileptologist's identification of the seizure onset zone.
- Develop and further modify the ARRm algorithm and apply it to intracranial EEG of epilepsy surgery candidates.
- Validate ARRm algorithm with HFO detection and epileptologist's identification of the seizure onset zone.
- Apply both methods to multiple EEG files (sampled at least 1000Hz or 2000Hz) and multiple patients, statistically analyze results.
- Implement ARRm algorithm in real-time embedded system application, test with scalp EEG for run time evaluation.

Some of these objectives were modified throughout the duration of this study, which is explained in the scope of the thesis. Additional goals of this research study were to better understand the mechanisms generating HFOs, and interface with Spectrum Health physicians to learn their process for evaluating epilepsy patients, ranging from surgery to EMU monitoring.

Scope

To begin the study, the HFO detection method was implemented, as defined in literature and code developed in a previous GVSU master's thesis, into a GUI framework (RIPPLELAB), and additional recordings were processed for a previously studied patient sampled at 1000Hz. It was found that for files in the beginning, middle, and end of time spanning a period of 7 to 13 days in the EMU, consistent patterns of high HFO counts were observed in the EEG for channels of interest. The original ARRM algorithm from literature was implemented in MATLAB, and initial plots were generated to compare to HFO detection between similar patients/files. Similar patterns of consistent high ARRM values on channels of interest were also observed in the ARRM results, indicating the ARRM measure and HFO rate may be related.

From these findings, it was determined that a tool should be developed in order to directly compare these two methods of iEEG analysis. An analysis tool was implemented in MATLAB, complete with a GUI and multiple functionalities for examining the results of the two methods. The initial ARRM findings showed promise, so the method was implemented on an embedded platform to prototype real-time analysis. Real-time collection/analysis was tested with 1 scalp EEG channel at a sampling rate $<1000\text{Hz}$ due to memory limitations of the hardware. It was shown that the ARRM method could model the signal and calculate the ARRM value in real-time.

Following the additional recordings processed for 1 patient, data analysis was expanded to 2 additional patients that had data recorded at 2000Hz. Results for HFO detection yielded frequent instances of high HFO rates across all channels. It was discovered that an optimal threshold had been chosen for events of interest detection upon close examination of comments in the original code of the HFO detection method. This was done as opposed to calculating the threshold with the standard deviation and mean of the envelope of the signal, as defined in the literature and as had been implemented in RIPPLELAB. Moving forward, optimal thresholds were chosen for each

patient, and 1-2 files were reanalyzed with the HFO detection method. Statistical measures were used to determine the correlation between channels with high HFO counts and high ARR_m values.

Chapter 2: Review of Literature

1. Anatomy & Physiology of the Brain

The human brain is considered central to human well-being. This fundamental organ of the human nervous system regulates thought, memory, judgement, personal identity and other aspects of the mind. The brain is the center of learning and regulates vital bodily functions. The brain and spinal cord make up the central nervous system, which communicates information to all parts of the body.

1.1 Anatomy

The brain is contained in the neurocranium, the uppermost part of the skull, and is bathed in an extracellular fluid known as cerebral spinal fluid (CSF). The cerebrum is made up of the left and right hemispheres of the brain and consists of approximately 100 billion brain cells [2]. The cerebrum is the largest and most highly developed part of the brain where complex functions of action and thought take place. Four sections, or lobes, make up the cerebrum (Figure 1):

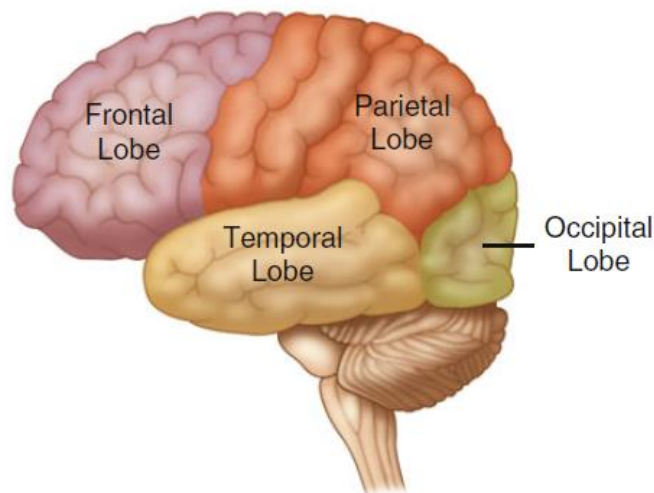


Figure 1: Diagram of the Cerebral Brain Lobes [3]

The high-level functions of each lobe can be referenced in Table 1.

Table 1: Brain Lobe Functions

Lobe	Function
Frontal	cognition: speech, planning, problem solving
Parietal	sensation: touch, pressure, judging size and shape
Temporal	visual and verbal memory, smell
Occipital	visual reception and recognition of shapes and colors

In order to navigate the brain, proper terminology is described in Table 2.

Table 2: Terminology for Positions of Anatomical Features of the Brain

Term	Definition
medial/mesial	middle
lateral	side
caudal	below and behind
superior	above, on top
inferior	below, underneath
ventral	lower
contralateral	opposite

1.1.1 The Cerebrum

The spinal cord and cerebrum are encased by the dura mater, which contains CSF that provides cushion to the brain. The dura is a thick, membrane that is attached by fibers to the inner side of the skull (Figure 2).

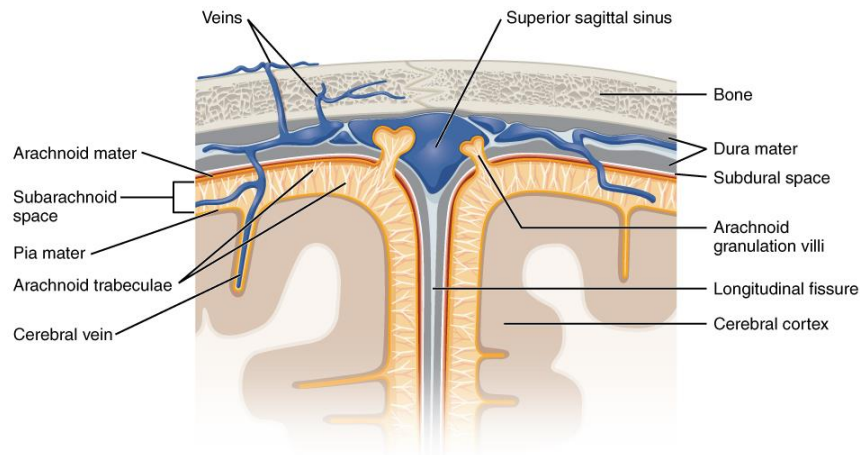


Figure 2: Layers of the Structures Enclosing the Cerebrum

The epidural space is defined as the space between the bone and dura. The surface of the brain is known as the cortex. Fissures on this surface, known as sulci, form boundaries between the lobes. The left and right hemispheres of the brain make it symmetrical in structure. Each hemisphere provides motor function to the opposite side of the body, and in each individual, one side is dominant. The two hemispheres consist of an inner core, white matter, and an outer layer (cortex) referred to as grey matter. The numerous wrinkled ridges on the cortex are known as gyri (Figure 3, Figure 4).

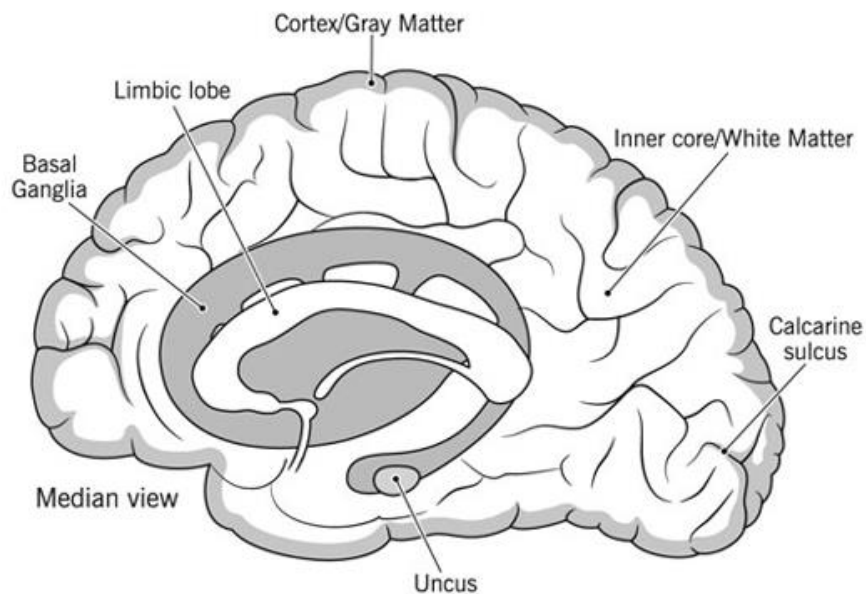


Figure 3: The Cerebrum (Median View) [2]

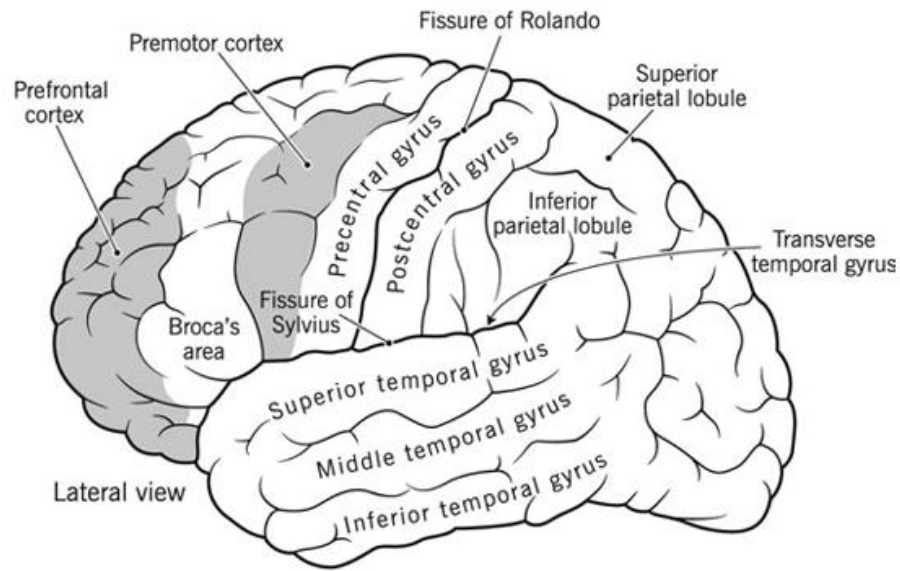


Figure 4: The Cerebrum (Lateral View) [2]

The central sulcus (fissure of Ronaldo) separates the frontal and parietal lobes, and the lateral sulcus (fissure of Sylvius) separates the temporal lobe from the frontal and parietal lobes. The precentral gyrus is an important structure in the frontal lobe, of which the primary motor region of the brain is located. Complex movements originate from the premotor cortex, and the prefrontal cortex applies inhibitory effects on actions (moral standards etc.). Broca's area is known as a region involved with speech. In the postcentral gyrus of the parietal lobe, sensory input from the opposite half of the body is received. The superior parietal lobule is considered association cortex, which means it is an area where neither sensory or motor processing occurs. Multiple sensory signals are integrated in the inferior parietal lobule. The temporal lobe is responsible for integrating movement and sensory functions, and the primary auditory area of the brain is located near the margin of the lateral sulcus. Near the lower surface of the temporal lobe, the uncus

contains the primary olfactory (smell) area of the brain. The occipital lobe is located directly below and behind the parieto-occipital/calcarine sulcus junction. This junction forms a “Y” shape, and the primary visual area lies on each side of the calcarine sulcus, which receive contralateral field of vision input [2].

1.1.2 Neurons

About 10 billion neurons make up the human brain; each neuron has a unique identity and function based on interactions with surrounding neurons and its secretions [2]. Neurons carry information to different locations throughout the brain. Each neuron consists of a cell body and branching structures that transmit and receive impulses to/from other neurons. The cell body is known as the soma (plural: somata), which contains the nucleus, and if the soma dies, so does the entire neuron. The axon is the tube-like fiber that extends from the soma, and this is the part of the neuron that carries nerve impulses. Large axons are myelinated, which means they acquire an insulating myelin sheath to reduce resistance. White matter is primarily made up of myelinated axons [4]. Some axons may be a fraction of an inch in length, and others can be as long as several feet. The branches of the neuron that extend from the cell body are known as the dendrites, and their function is to form receiving synapses with the axon terminals of other neurons [2] (Figure 5).

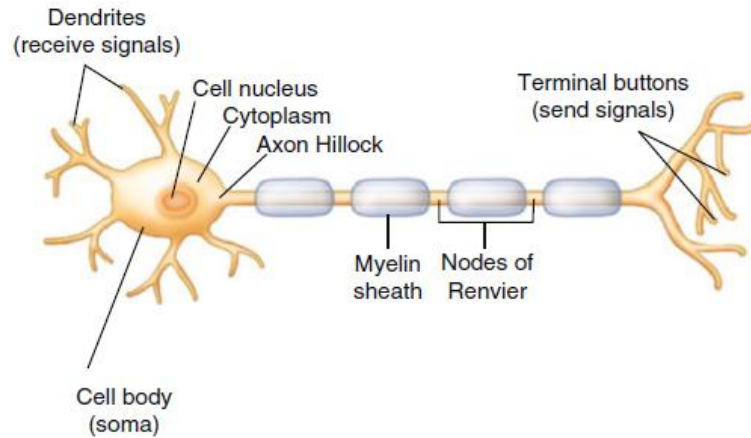


Figure 5: Anatomy of a Neuron [3]

Neurons signal between each other through electro-chemical transmission (neurotransmitters) across the synapse, which has a small space, the synaptic cleft, separating the pre- and postsynaptic membranes [3] (Figure 6).

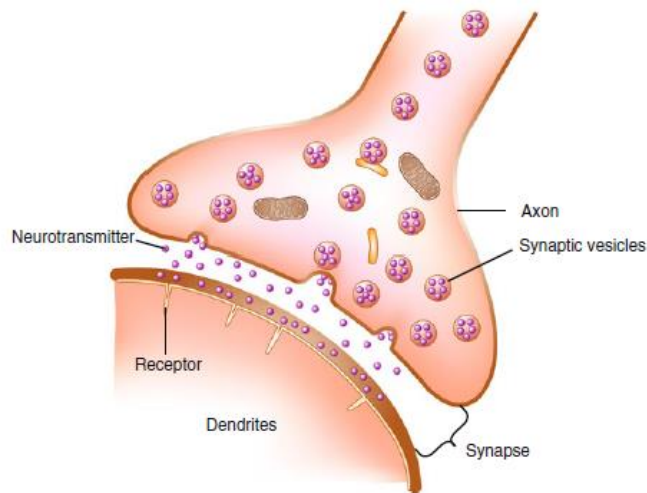


Figure 6: The Synapse [3]

1.2 Physiology

The nervous system is divided both functionally and anatomically. The central nervous system (CNS) includes the brain, spinal cord, and retina. The peripheral nervous system

(PNS) consists of all nerves that come off the brain and spinal cord. Functionally, the somatic nervous system governs voluntary actions and provides motor output for activating skeletal muscles. The autonomic nervous system controls all functions that are performed unconsciously such as respiration, heart rate, and digestion [3]. The brain fires neural impulses which control these divisions of the nervous system.

Neurons are cells specialized to transmit neural impulses. A neural impulse is described as the wave of depolarization that passes over the neuron. At resting potential, the neuron actively pumps sodium ions (Na^+) out of the cell, and potassium ions (K^+) into the cell. Large negatively charged proteins and ions such as phosphate and sulfate cannot diffuse out of the cell membrane, so the net charge is negative (-) inside the neuron and positive (+) outside the neuron. This potential is the difference in electrical charge between the inside and outside of the cell, and this unequal distribution of charges makes the cell membrane polarized at rest. The term “potential” means that there exists a drive for ions to move across the membrane [3] (Figure 7).

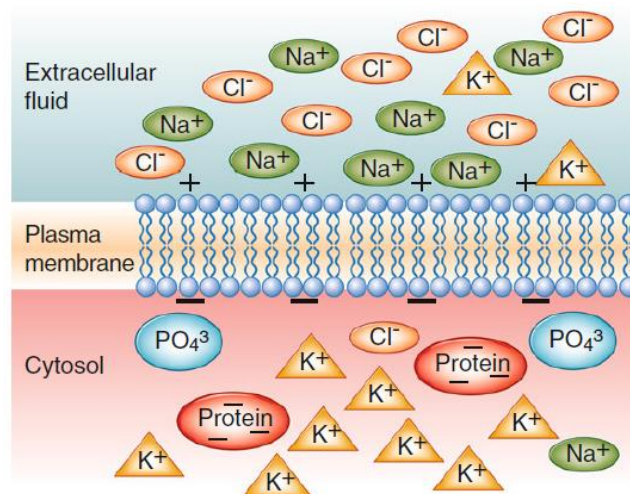


Figure 7: Membrane Potential of a Neuron [3]

To form an impulse, a stimulus is provided, which prompts an all-or-nothing response (impulse) depending if the stimulus is above a certain threshold. Once the neuron is activated, its membrane instantly becomes permeable to Na^+ , and the ions quickly diffuse into the neuron as Na^+ channels open and K^+ channels close. This inward flow of Na^+ for a brief instant causes the inside of the neuron to be positively (+) charged inside the cell, and negatively (-) charged outside the cell. This switch in polarity is known as depolarization, and causes a chain reaction wave that flows along the neuron. After the impulse flows, the cell repolarizes to resting potential [3], ready for the next impulse. At any time, a neuron is receiving multiple synaptic inputs. Some inputs are excitatory and some inhibitory. Each input creates a change in membrane potential called postsynaptic potential (PSP)

2. Electroencephalography (EEG)

EEG is the graphic display of spontaneous electrical activity of the cerebral cortex. Continuous EEG recording is used to diagnose brain pathologies and diseases, including epilepsy [3].

2.1. Origins of Neural Bioelectric Potentials

PSPs contribute to the EEG waveform, which are generated from the complex summation and integration of inhibitory and excitatory PSPs (IPSPs & EPSPs). Pyramidal cells in the cortex are the major contributors to the EEG waveform because of their linear arrangement. These neurons have an open electrical field and produce a dipole [3] .

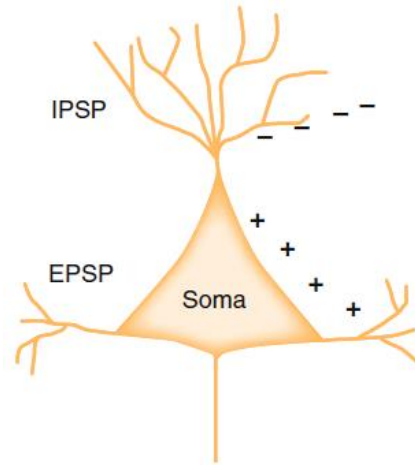


Figure 8: Pyramidal Neuron Schematic and Influencers of PSPs [3]

Since these neuronal potentials have a negative and positive dipole, if discharge is generated at the top of the cortex, then a vertically oriented dipole is created with maximum negativity above the source, and maximum positivity deep in the hemisphere (Figure 9, a).

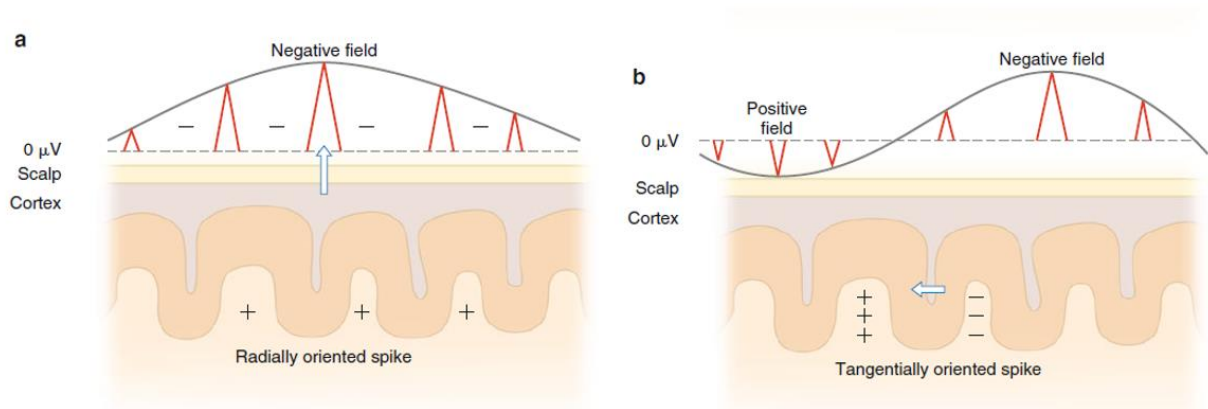


Figure 9: Schematic of Dipoles Related to the Scalp Surface

If a discharge is generated in a sulcus of the cortex, then a horizontally oriented dipole is created with maximum negative/positive fields directed anterior/posterior or medial/lateral [3] (Figure 9, b). The signals that are recorded measure a net potential of these dipolar discharges.

2.2.Electrodes

EEG can be recorded from electrodes placed on the scalp, or directly from the cortex, which is considered intracranial EEG (iEEG). Scalp EEG electrodes are used for many studies, but to record a quality signal, measures must be taken to reduce impedance such as abrading the skin and applying a conductive gel between the electrode and the scalp. iEEG electrodes are used to obtain more localized information and must be implanted via surgery. These electrodes are sterile, ready for use in the OR. iEEG is recorded with either cortical (Figure 10) or depth electrodes.

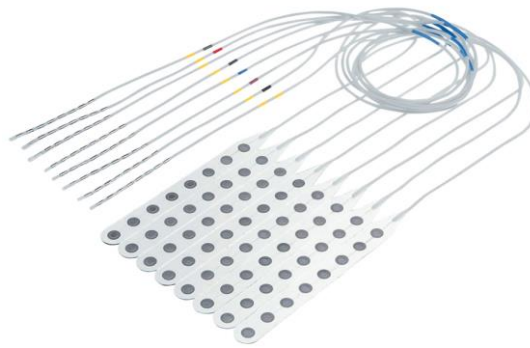


Figure 10: Cortical Electrode Grid and Leads

Cortical electrodes are organized on a sheet of flexible plastic material. The number of cortical electrodes on a grid varies, and the grid dimensions range in size. Some examples of electrode grid sizes include 8x8, 8x4, 2x8, or 2x6 (These dimensions represent the layout and number of the electrodes). Electrode strips (Figure 11) may also be used, and these only differ from grids because they have one row of multiple electrodes. Some examples of electrode strip sizes include 1x4, 1x6, or 1x8.



Figure 11: Cortical Electrode Strip and Lead

When electrodes are implanted directly on the cortex of the brain, the recording is called an electrocorticogram (ECoG). Depth electrodes (Figure 12) are thin insulated needles with multiple contacts that are advanced into brain tissue to record information from deeper brain areas/structures such as the thalamus [4]. Depth electrodes are particularly helpful in determining the SOZ in temporal or frontal lobe epilepsy [5].

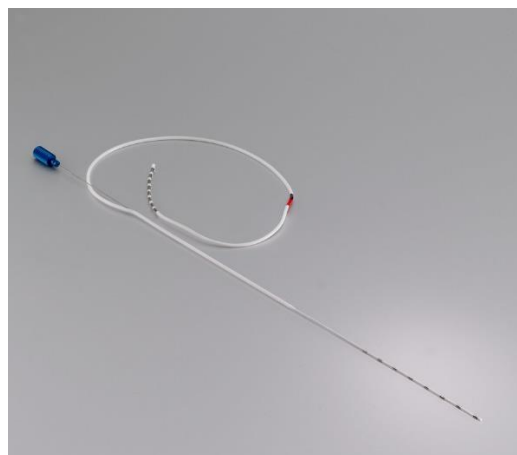


Figure 12: Depth Electrode and Lead

Some examples of depth electrode sizes include 1x6 or 1x8. iEEG electrodes are composed of metal materials that are both biocompatible and optimal for biopotential transfer. Brain

waves recorded from the surface of the brain may be as large as 10mV, whereas those recorded from the scalp have a smaller amplitude, approximately 100 μ V [4].

3. Epilepsy

Epilepsy is a neurological disorder that is defined by recurrent, unprovoked seizures. An epileptic seizure is defined as a “transient occurrence of signs and/or symptoms due to abnormal, excessive, or synchronous neuronal activity in the brain” [6]. An acute provoked seizure occurs in context of an acute brain insult such as stroke, head trauma, a toxic or metabolic insult, or intracranial infection. An unprovoked seizure occurs in the absence of an acute provoking event.

The most widely used system for classifying epilepsies was proposed in 1989 by the Commission on Classification and Terminology of the International League Against Epilepsy (ILAE) [6]. The ILAE’s operational definition of epilepsy requires that an individual have at least two unprovoked seizures on separate days, 24 hrs. apart. The ILAE also provides a conceptual definition: “epilepsy is a disorder of the brain characterized by an enduring predisposition to generate epileptic seizures and by the neurobiological, cognitive, psychological, and social consequences of this condition ... Elements in the definition of epilepsy include history of at least one seizure, enduring alteration in the brain that increases the likelihood of future seizures” [6]. There are two main categories of epilepsy: generalized epilepsy and partial epilepsy. Generalized epilepsy involves the entire brain all at once, whereas partial epilepsy involves a focal portion of the brain [4].

3.1 Etiology

According to the ILAE, there are two primary categories for the causes of epilepsy.

The first category is remote symptomatic (associated with a preexisting condition that

increases the risk of developing epilepsy). Antecedent factors may include history of a stroke, brain malformation, evident neurodevelopmental abnormality, history of brain infections, a genetic disorder, or tumors. The second category is idiopathic (reflects a group of well-characterized disorders whose initial onset occurs during childhood or infancy). The term idiopathic is intended to reflect an assumed genetic etiology where the sole manifestation is seizures. Another term, cryptogenic, means that the cause of the epilepsy is unknown. The epilepsy could be caused by a cortical malformation, such as a cortical dysplasia, or it could be caused by a genetic (idiopathic) epilepsy. New imaging techniques can uncover these malformations, and genetic investigations can identify new genetic syndromes. [6]

3.2 Refractory Epilepsy

The primary form of treatment for patients diagnosed with epilepsy is the administration of antiepileptic drugs (AEDs). Most patients become seizure free with AED treatment, but approximately 30% of patients with epilepsy have intractable disease (refractory seizures), and these cases qualify as candidates for epilepsy surgery. Three key elements are incorporated into the definition of medical intractability: number of AEDs, frequency of seizures, and duration of persisting seizures [6]. For operational purposes, medically intractable epilepsy can be defined when 2 appropriately chosen, well-tolerated AEDs have failed due to lack of effectiveness.

3.2.1 Electrode Implantation Surgery

General lateralization of the suspect epileptic areas in patients diagnosed with refractory epilepsy are determined prior to electrode implantation surgery. The team of physicians performs a preliminary evaluation on the patient using a combination of surface EEG recordings, magnetoencephalography (MEG), or imaging modalities

such as CT or MRI scans. The next step of treatment is a phase II electrode implantation surgery, in which surface and/or depth electrodes are implanted in the subdural space, directly on/into the cerebral cortex (Figure 13).

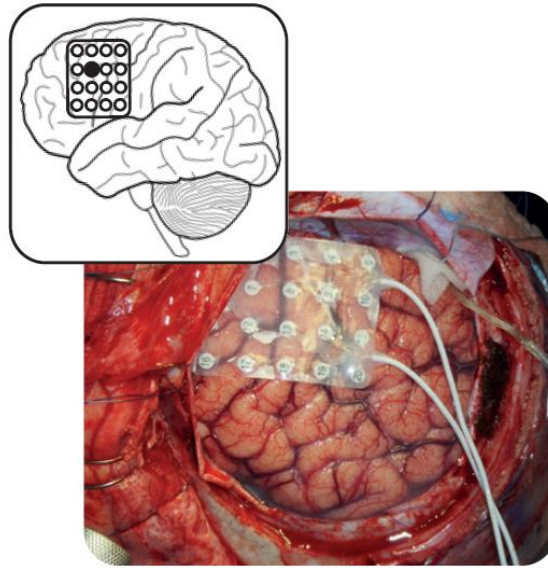


Figure 13: Surface Electrode Grid Implantation onto the Cerebral Cortex [1]

An area as wide as possible is covered when placing cortical electrode grids, and “relaxing” cuts in the plastic support between electrode rows may be helpful to avoid grid buckling and piercing of the dura with the wires. Electrode strips may also be implanted depending on anatomical location. A stereotactic frame or stereotactic robotic assistance (Figure 14) may be used to insert depth electrodes. When using a stereotactic frame, it is affixed to the patient’s head with pins, then placement of the depth electrodes is guided to the target site. More recently, frameless placement of depth electrodes is performed with the ROSA® robot [5].



Figure 14: ROSA® stereo EEG Robot Allows for Accurate and Steady Placement of Depth Electrodes

If the dura adheres to the surface of the brain, electrodes can be placed in the epidural space, which may reduce the amplitude of EEG recordings. Grid, strip, and depth electrode locations are documented with schematic drawings (Figure 15), photographs, or 3D models. The schematic drawings are completed intraoperatively, as each group of electrodes are placed. The sheets of blank brain maps are available in the OR in sterilized packages, so that the surgeon can draw diagrams in the sterile field as the surgery progresses.

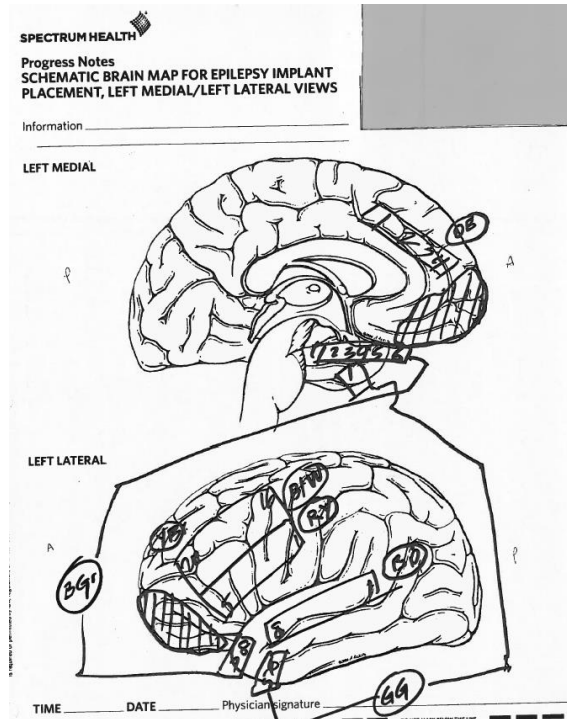


Figure 15: Example of Electrode Mapping Schematic

AEDs are continued during postoperative recovery before the patient enters the EMU. The most common complication following the placement of subdural electrodes is infection, which can be avoided by ensuring adequate closure of the dura to prevent CSF leakage, and replacing dressings frequently [7].

3.2.2 Identifying the SOZ

Pursuing identification of the epileptogenic zone is critical to guiding resection surgery. The epileptogenic zone has all driving components that truly contribute to seizure activity, and it includes the actual SOZ, and the potential SOZ. The actual SOZ is the area of cortical tissue where recorded seizures arise, and the potential seizure onset zone is adjacent or distant cortex that does not primarily generate seizures, but still may lead to seizures even if the actual seizure onset zone is resected [6]. Exactly pinpointing the epileptogenic zone is not yet a task possible to

be executed with absolute certainty. This is because the epileptogenic zone is thought of as a complicated network of nodes that communicate in chorus together to generate seizures. Some nodes are big contributors, and others are smaller contributors. There is no clear function or structural marker that can exactly delineate the epileptogenic zone. In the majority of cases, the SOZ is included in the resection, but removing the entire SOZ may not necessarily result in successful surgical outcome because some seizures can originate from outside the SOZ. Sometimes it is difficult to define the SOZ when seizures are onset by a widespread spike [8]. When a widespread spike occurs, all channels reflect that activity at the same point in time. The success rate of epilepsy surgery ranges between 36 and 84% [1]. This statistic emphasizes the fact that the chances of success from surgical intervention are determined by how well the borders of the epileptogenic zone are defined.

The current clinical method for identifying the SOZ is an extensive process. Upon entrance to the EMU, the patient is weaned off AEDs and monitored for seizure activity for up to approximately 2 weeks. During this time, iEEG is recorded, and the epileptologist reviews the signals manually. First, some iEEG is examined over time slowly, and it is noted if some channels may not be recording properly, or if noise is present on any of the signals. Next, iEEG is searched for a seizure. If a seizure (ictal activity) is recorded, the iEEG is examined in the time frame right before the seizure started to pinpoint channels that show origins of the seizures. The time between seizures (inter-ictal activity) is examined to search for epileptiform spikes known as Interictal Epileptiform Discharges (IEDs). These spikes are sharply

contoured waveforms that last 20-70ms [9]. Some main features of these IEDs include:

- 1) They are distinct from the background, i.e. are not part of a preceding rhythm.
- 2) Sometimes they can occur in trains.
- 3) Usually they are followed by a slow wave.
- 4) Often, the slopes of the IED are asymmetrical: the initial component is typically steepest, and followed by a slower second component with larger amplitude.
- 5) Commonly, the biopotential field of IEDs extend over a few electrodes [9].

These inter-ictal spikes, along with pre-ictal activity, are currently considered the gold standard biomarkers for guiding resection (Figure 16).

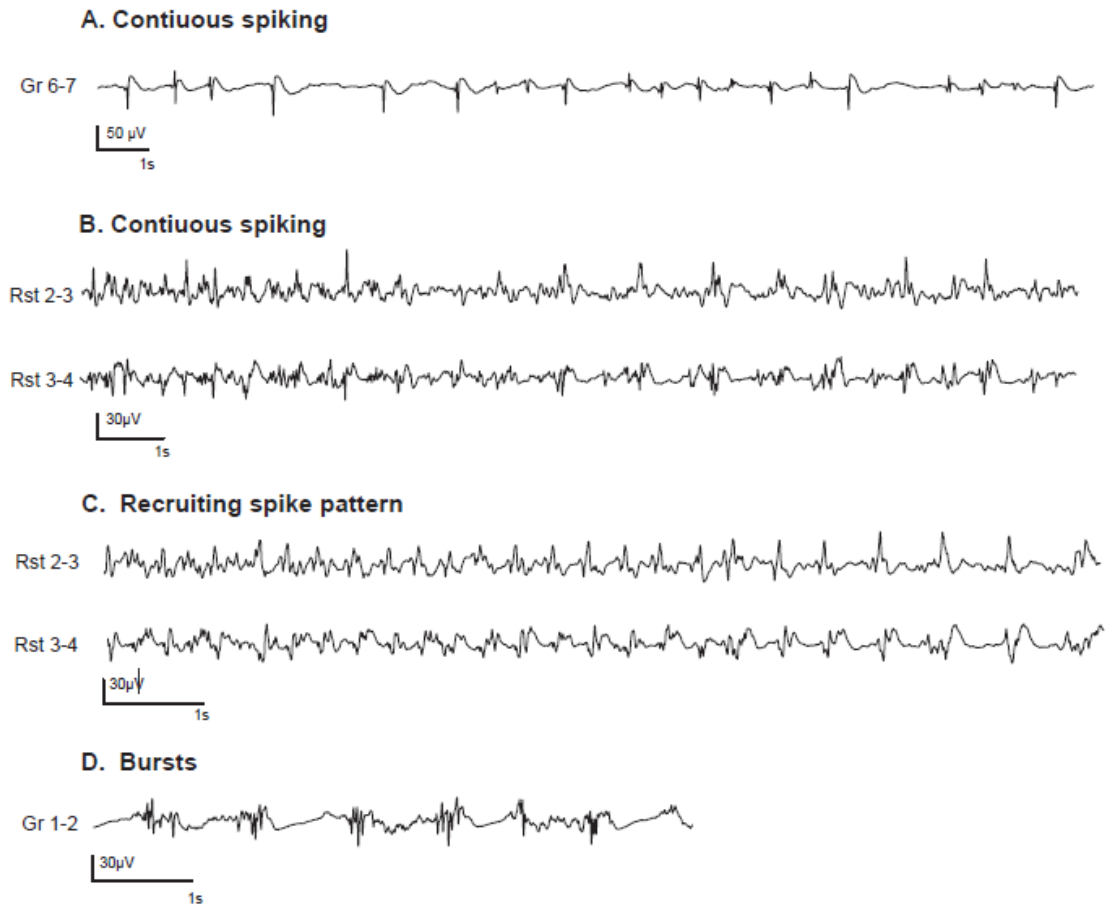


Figure 16: Epileptiform Spike Patterns on Bipolar EEG Recordings [1]

If there are any epileptogenic lesions such as cortical dysplasia, tumor, or mesial temporal sclerosis, they are also considered for resection. Patients with epileptogenic lesions are often strong candidates for resection surgery, and chances of seizure freedom are high. The borders of eloquent regions of the brain, such as speech regions or motor cortex, are defined before the patient leaves the EMU by applying electrical stimulation and observing motor responses or loss of cognition. After channels of interest are identified based on IEDs and pre-ictal activity, the epileptologist and neurosurgeon consult to determine if the patient is a good candidate for surgery, and finalize a definition of the resection area. Patients are good

candidates for surgery if the epilepsy has a focal onset zone, but if there is no indication of a focus region, there is likely an extensive network of nodes causing the seizures.

3.2.3 High Frequency Oscillations

Recent studies of iEEG in frequencies $>70\text{Hz}$ have shown that neurological tissue produces HFOs. These high frequency oscillations can be subdivided into ripples (80-250Hz) (Figure 17, A) and fast ripples (250-500Hz) (Figure 17, B) [1]. HFOs that are considered potentially pathologic/epileptic oscillatory activities occur within these defined frequency bands, persist for at least 4 oscillation cycles, and last up to 10ms [10].

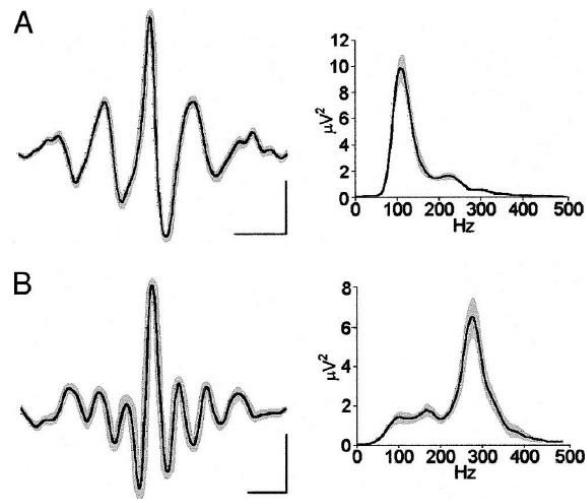


Figure 17: (A) HFO Ripple and (B) HFO Fast Ripple

The inter-ictal period shows epileptiform discharges/spikes, but these seem to represent a larger area than the epileptogenic zone since their large amplitudes affect multiple electrodes [1]. In order to record HFOs in iEEG, the sampling rate must be at least 1000Hz according to Nyquist's frequency rule, and 2000Hz is preferred to capture the HFO waveform [8]. Retrospective research has shown that inter-ictal

HFOs, especially fast ripples (FRs) overlap with, but occur in a smaller area than inter-ictal spikes and are a better representation of the SOZ than inter-ictal spikes. It has been theorized that HFOs, especially FRs, might be more local to the epileptogenic zone than solely inter-ictal spikes [11], [12].

Epileptic FRs have been hypothesized to result from tissue changes causing hyper synchronous action by electrotonic coupling via gap junctions, out-of-phase firing by groups of principle neurons, fast synaptic transmission, and ephaptic interactions [13]. Ephaptic interactions describe non-synaptic couplings between neurons through the exchange of ions or local electrical fields, which can influence the timing of AP firing. Normal myelination of axons reduces ephaptic interactions [14]. Gap junctions are groups of membrane proteins that couple cells by allowing the exchange of molecules and cytoplasm, therefore allowing electrical current to travel [14]. Helling *et. al* studied the effects of increased gap junctions on a computational neural mass model. It was shown that single neuron firing can result in fast recruitment of cells interconnected by gap junctions, which results in AP firing resembling an HFO. With these changes in behavior, cascading these models resulted in emergent bistability of the higher order neural mass model, which led to epileptic activity [15].

Neurons in pathological tissue respond abnormally to subthreshold stimuli or they become spontaneously active, recruiting interconnected cells, resulting in the synchronous AP firing which manifests in extracellular recordings as an HFO. HFOs are known to be involved in generating seizures before and during the seizure onset period [12]. Understanding the underlying mechanisms of HFOs can lead to a better understanding of these abnormal dynamics of epileptic neuronal networks.

Not all HFOs are considered abnormal activity, some HFOs overlap with pathological HFO frequency bands ($\sim 200\text{Hz}$), which are primarily located in the hippocampus or medial temporal lobe. Physiological HFOs are known to be involved in memory formation, reactivation of previous experiences or other information processing functions [8]. Studies have shown that HFOs in the FR band are specific to epileptic tissue, so this is a method of distinguishing physiological from pathological HFOs [12]. iEEG is prone to muscle or movement artifacts, and these should also be distinguished from epileptic HFOs. Muscle artifact typically has a wide frequency band, and movement spikes have abnormally large amplitudes. Sleep iEEG often has the clearest artifact-free recordings, and epileptic HFOs can be seen co-occurring with IEDs [10]. Sharp epileptogenic events in the iEEG can give rise to artificial HFOs after high pass filtering the signal, but it can be noted that for clinical purposes, HFOs that occur with spikes and regular isolated HFOs may equally reflect epileptic activity [16]. Most studies have concentrated on investigating HFOs recorded in pre-resection iEEG, showing increased seizure freedom when electrode locations with ictal or inter-ictal HFOs were removed [1], [16], [17]. Other studies have shown that after resection surgery, patients with residual FRs have been associated with poor seizure freedom [18] [19], and the removal of tissue that generates FRs is associated with seizure freedom post-surgery [20].

3.2.4 Epilepsy Surgery

In most cases, epilepsy surgery leads to significant reduction in seizure frequency and often, seizure freedom. After the area of the brain for tissue removal has been defined from SOZ localization, the patient leaves the EMU and undergoes epilepsy resection surgery, or a NeuroPace responsive neurostimulator (RNS) [21] is

implanted if the resection area overlaps with an eloquent region of the brain. The RNS system continuously monitors iEEG activity and is programmed to respond in real time with electrical stimulation if abnormal activity is detected. It is a closed loop system, designed to suppress and prevent the onset of seizures [21] (Figure 18).

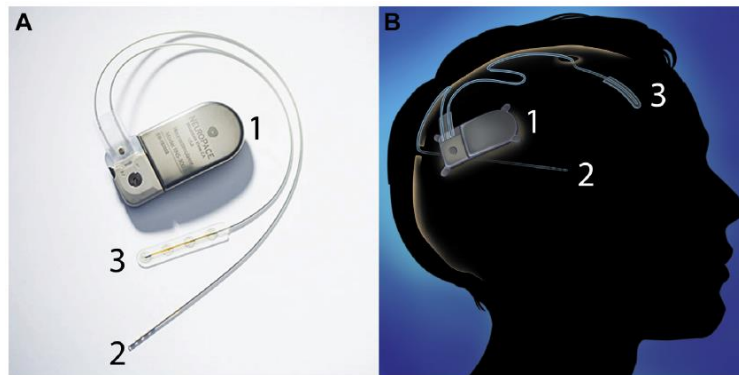


Figure 18: (A) NeuroPace RNS. (B) Schematic of Implanted Neuropace. The RNS device (1) is attached to a depth electrode (2) and/or a cortical electrode strip (3) [21]

If the resection area is near an eloquent region, the patient will be awake as resection is performed, and electrical stimulation will be applied to the brain tissue to guide the surgeon's cuts in real-time. As the surgery begins, the previous bone flap from electrode implantation surgery is re-elevated, with care taken not to disturb the electrode grids. Electrode wires are then cut at the dura, and the dural incision is re-opened. All electrodes are removed and accounted for, then the surgeon proceeds with resection [7].

In cases where the patient needs to be awake for language and motor mapping, local anesthesia is used and general anesthesia is reduced after bone flap removal and dural opening. Because of observations in HFO research, there has been movement towards tailoring epilepsy surgery to rely on the rate of HFO occurrences, and also detecting HFO activity in real-time intraoperatively [1]. If

HFOs were to be recorded intraoperatively, consideration must be given to the affect of anesthesia on the signals. Propofol is commonly the choice for inducing general anesthesia. In a study by van't Klooster *et. al*, their process for measuring iEEG intraoperatively was to reduce Propofol until a continuous background iEEG pattern was achieved [1]. Zijlmans *et. al* more closely examined the influence of propofol on epileptic HFOs in intraoperative iEEG. In this study, it was found that ripples and FRs increased between the first and last minutes of iEEG measurement after propofol reduction. It was found that the occurrence of spikes remained unchanged. This means that HFOs respond differently to propofol than spikes do. Considering the antiepileptic effects of propofol, the increase in HFOs after reduction of propofol strengthens the hypothesis that HFOs mirror epileptic activity [13].

4. HFO Detection Methods

Since it has been shown in many studies that HFOs are a promising new biomarker for epileptogenic tissue, signal processing methods have been developed to automatically extract this information. HFOs occur in high rates, and the process of visual HFO evaluation is time-consuming, so the need for automated HFO detectors is great.

4.1 Modified Burnos Method

The most recent HFO detector developed at GVSU has been adapted from a method by S. Burnos [22], [23]. The aim of this detector was to improve HFO detection and discriminate more accurately between artifacts and HFOs. Processing was applied to each channel (recording from a single electrode) in a single iEEG file representing a long-term recording of up to 3 hrs. After bandpass filtering the signal (80-488Hz) and obtaining the upper

envelope of the signal with the Hilbert Transform, a threshold was applied to obtain a large number of events of interest (EOIs). Events of interest are any instances of high power high frequency activity, which presents as an event with large oscillating amplitude in the filtered signal. The positive envelope is defined as a smooth curve that outlines the positive extremes of an oscillating signal. If the envelope was higher than the threshold, it was considered an EOI. The threshold was calculated based on the standard deviation and mean of the envelope of the signal (Equation (1)).

$$\text{Threshold} = 3 \times \text{std}(\text{envelope}) + \text{mean}(\text{envelope}) \quad (1)$$

This step has a high sensitivity and low specificity to analyze as many EOIs as possible (Figure 19).

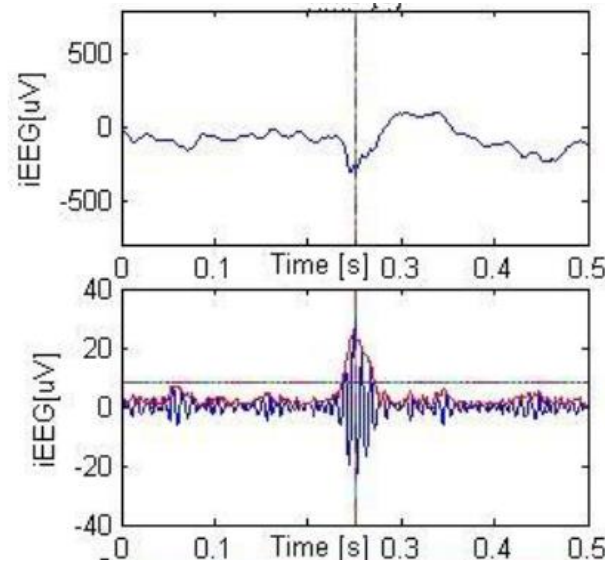


Figure 19: Raw iEEG Time Segment with HFO & Filtered Segment with Envelope (red) and Threshold

The next step of the method is to distinguish between an EOI and an HFO. HFOs will have a certain level of separation between high frequency and the lower baseline frequencies.

HFOs are characterized as short lived events that have an isolated peak within the HFO frequency. Instead of applying the Fast Fourier Transform (FFT) to quantify the frequency spectrum, the Stockwell (S) Transform [24] was chosen. As opposed to the FFT, the S Transform provides time-frequency resolution with frequency-dependent resolution, but also maintains a direct relationship with the Fourier spectrum. This allows for detection of short-lived high frequency signals. By taking the S Transform of a short time window that contains an HFO, and plotting the power spectral density, the ratio between the peak before/after the trough and the trough reveals if the high frequency event is a confirmed HFO (Figure 20). The vertical red line in the heat plot marks a detected HFO event, and a gap of lower power can be observed between the strong power at ≈ 100 -150Hz and the baseline frequencies < 100 Hz.

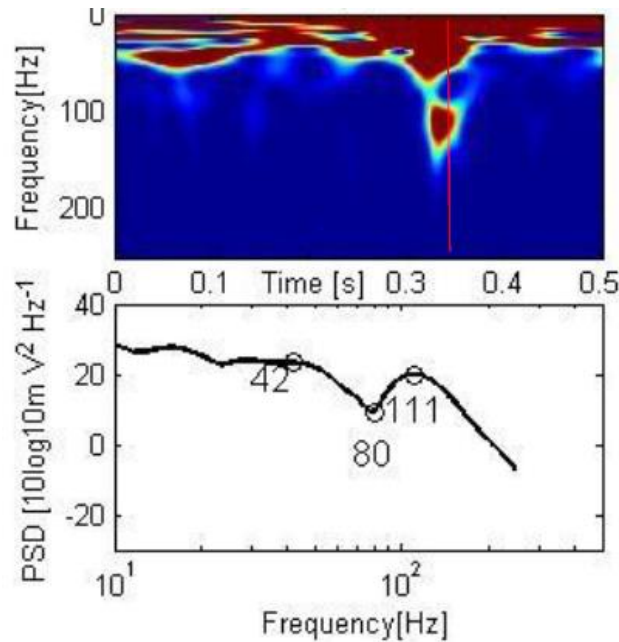


Figure 20: Frequency Content & PSD Plot of the Stockwell Transform of Time Segment with HFO

The iEEG with an artifact spike shown in Figure 21 was not classified as an HFO because the peak in frequency content was above 500Hz.

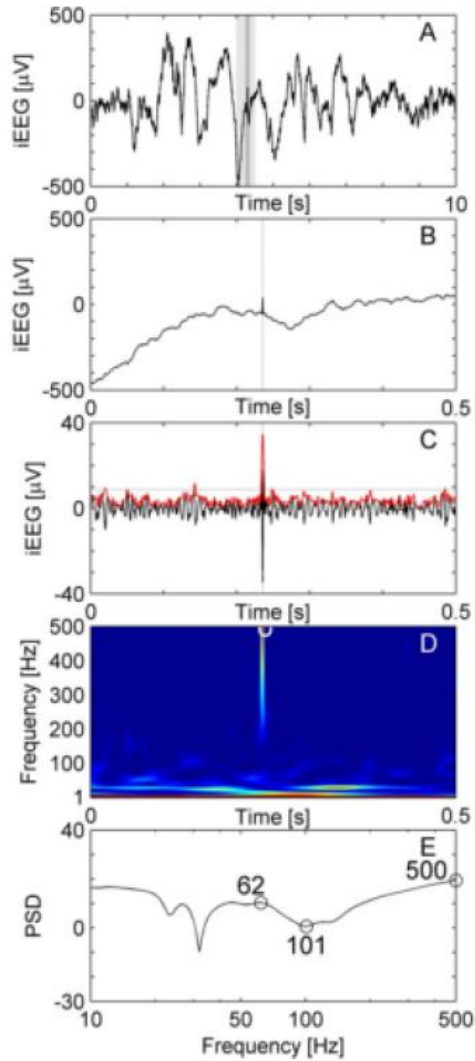


Figure 21: Example of iEEG with Sharp Artifact

The result of this detection method is that sharp transients or artifacts in the signal will be thrown out from the EOIs, and only individually detected HFOs will remain in the final selection of relevant events.

4.2 MNI Detector

In HFO detection studies by van't Klooster *et. al* [1], the MNI algorithm was used to detect HFOs. This method was developed at Montreal Neurological Institute. It consists of two detectors, which makes it adaptive to two types of HFO activity. The first detector is for HFOs in baseline EEG segments, and a second detector for HFOs in channels with continuous high frequency activity. The aim of the MNI detector is to have a large sensitivity and low specificity [25]. First, a channel of EEG is segmented into 125ms windows with 50% overlap. Segments of EEG are first bandpass filtered (80-450Hz), and the detector determines if there is baseline activity in the signal. MNI defines the baseline as EEG where there is no oscillatory activity, so the baseline detector is based on wavelet entropy applied to the autocorrelation of the filtered signal, measuring the degree of randomness in the signal (vs. oscillatory behavior) [25]. Wavelet entropy measures the degree of disorder in a signal in both time and frequency domains. Applying wavelet entropy to the autocorrelation of the filtered signal enhances the oscillatory characteristics of the signal, because if the signal is oscillating, the autocorrelation will have periodic peaks. A segment of EEG is considered baseline (no oscillation present in the signal) if the wavelet entropy is larger than a certain threshold. If there is “sufficient” baseline present for $> 5\text{s}/\text{min}$ in the signal, then the HFO detector for channels with baseline is executed, and if there is baseline $< 5\text{s}/\text{min}$, then the HFO detector for channels with continuous HFO activity is executed.

For HFO detection in channels with baseline, possible HFOs are detected based on energy above a certain threshold for $> 10\text{ms}$ [25]. The threshold is calculated by obtaining the Cumulative Distribution Function (CDF) of each baseline segment, modeling the distribution with a gamma function, and using the value at the 99.9999 percentile as the

threshold [25]. Unlike other detectors, the threshold is calculated with respect to the baseline.

For HFO detection in channels with continuous HFO activity, an iterative approach is taken. At each iteration, a threshold was computed from 1 min of filtered EEG as the 95 percentile of the CDF modeled with a gamma function [25]. Segments with energy above the threshold $> 10\text{ms}$ were considered HFOs and then removed from the signal, so this way, the events with highest energy are detected first, and removed for following iterations [25]. At each subsequent iteration, a new threshold is calculated and more HFOs are detected/removed from the signal, and iterations continue until no more HFOs are detected.

The MNI detector is relevant to HFO detection because it has been found that removal of areas of the brain with the occurrence of ripples in flat background activity is correlated with a seizure free outcome [20]. Kerber *et. al* found that background patterns should be considered when evaluating HFOs in the ripple band, and it may improve delineation of epileptogenic areas [20].

5. ARRm Method

Automatic detectors have been developed to extract individual HFOs from iEEG, but this type of analysis is done by post-processing the data in the time and frequency domain. Though most recently developed automated algorithms have been successful in detecting HFOs, they are computationally expensive, and would be difficult or impossible to implement in real time for intraoperative analysis. Since iEEG must be sampled at 1000-2000Hz for HFOs to be detected by automated algorithms, these current methods take multiple hours to process just 2 hours of iEEG because of the large number of sample points that must be transformed in the frequency domain. Furthermore, filtering is always involved in these methods, and the high-pass filtering of

sharp epileptic transients or movement/EMG artifacts in the EEG can produce false HFOs in the signal [26]. This result mimics what happens when an impulse function is high-pass filtered.

The nature of epilepsy is that it is unpredictable: a dynamic condition where seizures occur intermittently. This suggests that the underlying processes responsible for the generation of seizures are non-harmonic, i.e. non-linear: cannot be described as a solution to a linear differential equation of a certain order. Deviation from harmonic, predictable behavior can be caused by noise, nonstationary parameters, or nonlinear dynamics [27]. The non-harmonic characteristics of pathological HFOs have been examined in studies by Geertsema *et. al* [27], [11]. It was found that a high level of nonlinearity in the neurological system is needed to initiate a seizure, and epileptic iEEG biomarkers (inter-ictal spikes, HFOs, and seizures) all have a strong non-linear component. Based on these observations, an automated detection method was developed known as auto-regressive model residual modulation. ARRM has the potential to identify iEEG channels that are within the epileptogenic zone by quantifying the non-harmonic features of high-frequency iEEG components. An AR model of order p approximates an N point time series $y_{model}(n)$ as a linear combination of past observations $y(n-k)$ plus a noise/error component ε . The ε component represents the difference between the actual signal and the model signal.

$$y_{model}(n) = \sum_{k=1}^p a_k y(n-k) + \varepsilon \quad (2)$$

There is no phase shift associated with AR modeling because the first point that can be predicted is the $p+1$ point in the signal. The predicted value $y_{model}(n)$ is compared with the actual sample $y(n)$ to calculate ε . AR modeling was chosen because it is a time-domain method that can be implemented in real-time, and the frequencies and damping coefficients

resulting from AR modeling are able to quantify the oscillatory properties of iEEG. Franaszczuk *et. al* found that EEG time series can be modeled as an output of a linear system with white noise input, and when AR modeling is applied to the signal, the AR coefficients can be used to get the transfer function of the system. By the L transform, the impulse response function can be determined in the form of damped sinusoids. Differential equations can also be constructed from the transfer function, and they can be written in the form of the equations for damped oscillators, representing the generators of normal EEG rhythms [28].

The residual of a model is described as the difference between the original signal and the modeled signal, representing the noise/error component of the AR model equation (ϵ). Since the AR model describes harmonic oscillators that are stationary and linear, generally, the residual variance of the signal is considered noise, but in this application, it is reminiscent of the nonharmonic activity in the iEEG. Geertsema *et. al* performed a preliminary study and demonstrated that segments of iEEG containing ripples measured in areas close to the SOZ exhibited higher residual signal variation after AR model fit than windows with ripples from areas outside the SOZ [27]. In order to quantify frequently occurring high residuals, a coefficient of variation (normalized standard deviation: ARRm value) is applied. The process leading up to the calculation of the ARRm value is shown in Figure 22.

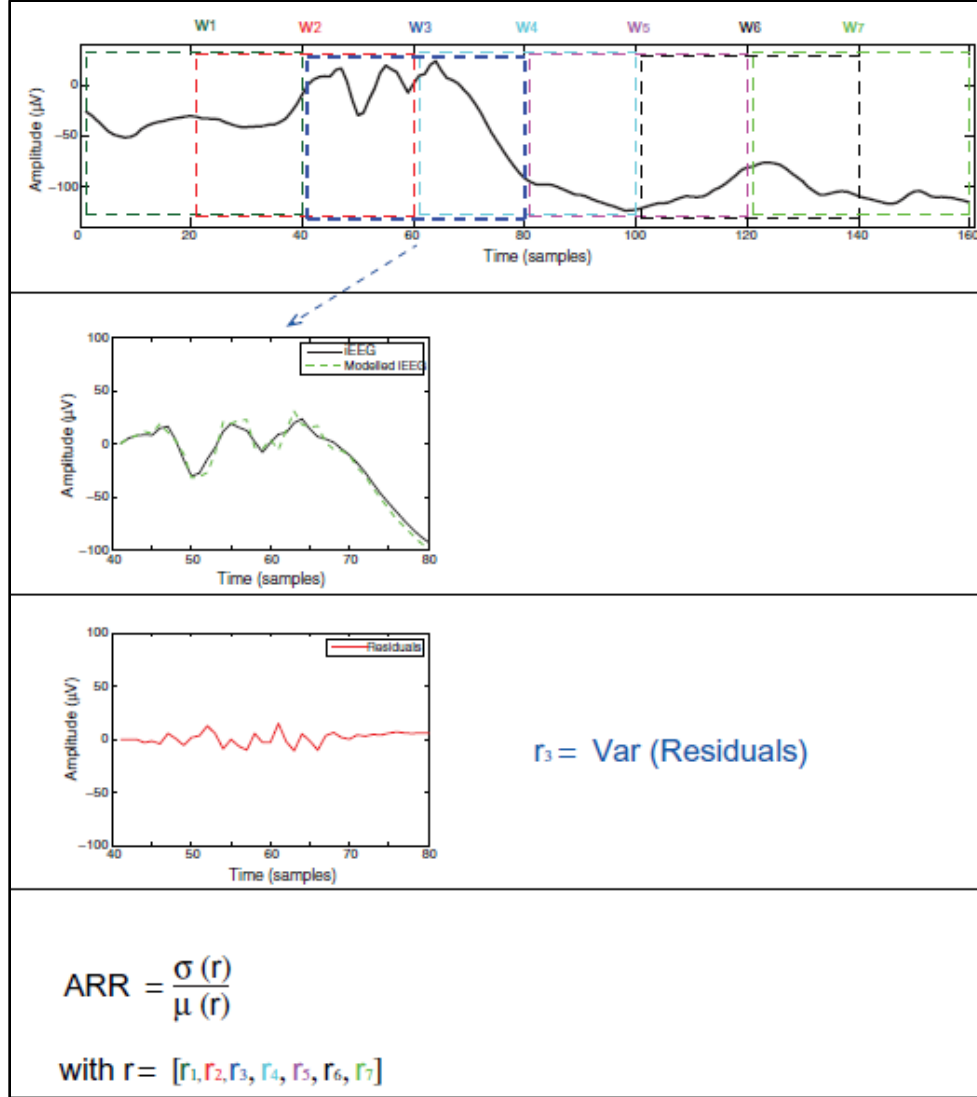


Figure 22: ARRm Algorithm Processing Steps [27]

The value r represents the residual signal variation values for each window of data, ($r_1 \dots r_n$), 50% overlapped comprising 1 minute of iEEG data. The ARRm value is therefore a ratio of the standard deviation of r to the mean of r , which is a measure of relative variability.

$$ARRm = \frac{\sigma(r)}{\mu(r)}$$

(3)

It has been found that iEEG channels with high $ARRm$ values may be related to the SOZ. Though the $ARRm$ method does not explicitly identify individual HFOs, the $ARRm$ value quantifies the level of activity that deviates from normal harmonic rhythms and could be a result of transient activity such as HFOs and/or inter-ictal spikes. In Geertsema's most recent study [11], the method was improved by implementing measures to reject artifacts that would otherwise product large residual signal variations. These measures were based on the observation that m values stayed high across all model orders when artifacts occurred, and when epileptic events occurred m values declined across increasing model orders. Residual decline was calculated over AR models 1 & 2, and if the decline was below a threshold value, it was considered artifact and the m value was ignored (artifact rejected). However, artifact rejection did not have a large impact on the ability of the $ARRm$ method to quantify FRs (Figure 23). This plot also demonstrates that high $ARRm$ values are most strongly associated with FRs.

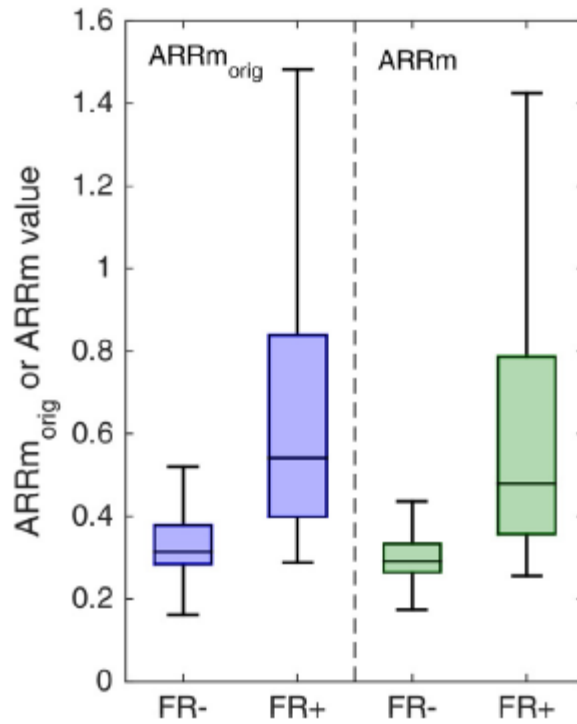


Figure 23: Performance of the Original and Improved $ARRm$ methods in Identifying FRs [11]

A clinical threshold was calculated based on the 95% specificity point to identify if a channel was likely located in the epileptogenic zone. iEEG was measured from patients pre and post-resection, the $ARRm$ value was calculated on the raw signal, and Ripples, FRs, and spikes were identified manually. The $ARRm$ value tended to be elevated in channels with ripples and spikes, but was especially high in channels with FRs.

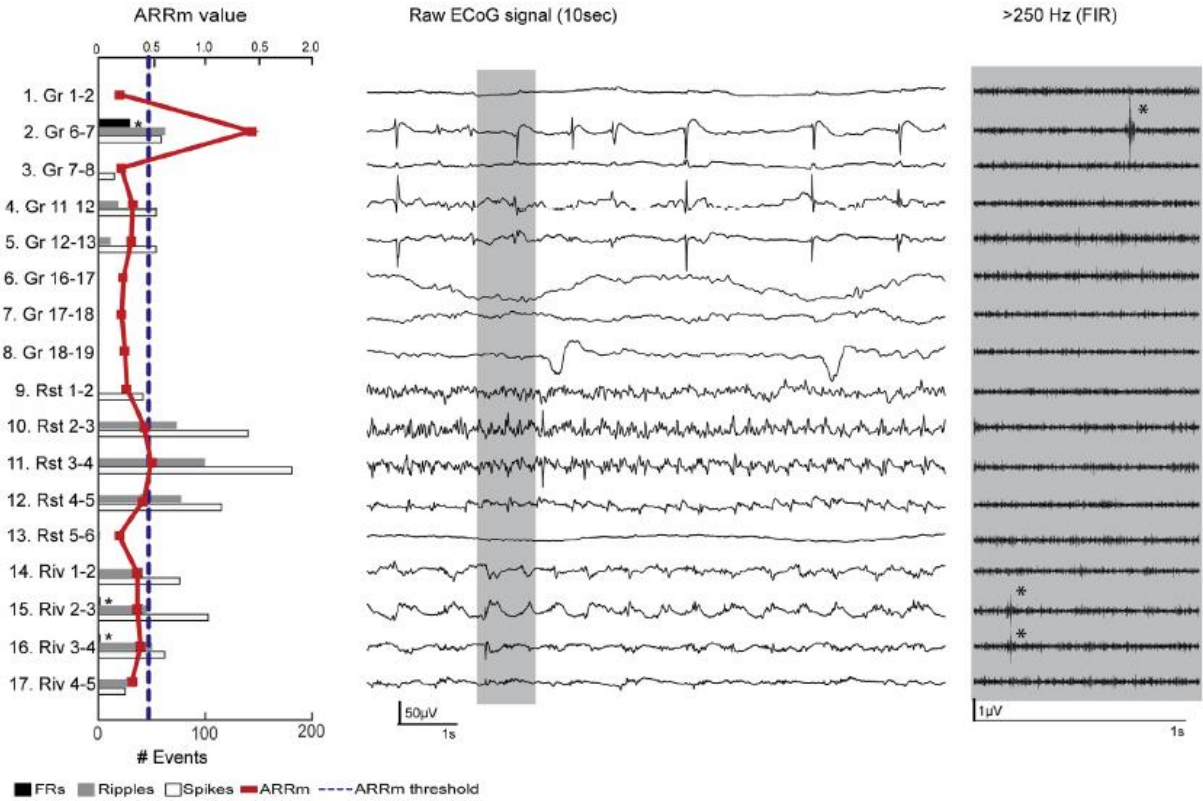


Figure 24: Patient Example of ARRm Results

In this study, it was found that at a channel level, ARRm values above the threshold in post-resection iEEG predicted poor seizure freedom for the patient [11]. It is especially promising that high ARRm values represent channels that have FR activity, which in other studies has been associated with primarily epileptogenic activity, and removal of areas with FRs has been associated with seizure freedom. Since the ARRm value presumably quantifies nonharmonic activity, and HFOs are considered nonharmonic, it is predicted that an HFO detector would yield similar results in identifying channels of interest in the SOZ, but at much higher computational cost.

Chapter 3: Methodology

6. Modified Burnos HFO Detection Method in RIPPLELAB

The Modified Burnos HFO detection method was previously implemented in a custom MATLAB application [23], but in order for this method to be usable in a clinical setting, a GUI was needed. An open-source comprehensive application, RIPPLELAB was developed to facilitate multiple different types of HFO analysis methods with a GUI [29]. Four methods for automated HFO detection were pre-implemented in the application, and new functions for other methods could be added. The Modified Burnos HFO detection method was added to the toolset in RIPPLELAB, so that users can simply load files and analyze iEEG in a GUI setting. Visual display of the raw data can also be done in this application. The steps for basic use of the GUI to run the Burnos detection method are outlined below:

- 1) Select the Open icon, load .edf (European Data Format) iEEG file.
- 2) Select channels to analyze from listbox, then click Add.
 - a. If the user wishes to view raw iEEG, select Display Ch.
 - b. Otherwise, select Apply.
- 3) Select the HFO Detection Methods icon.
- 4) Select channels to analyze from listbox, select Riazul (Author's Name) from dropdown menu, then enter the appropriate frequency parameters in Hz. (Min = 80, Max = 488).
- 5) Select Analyze HFO to begin iEEG analysis.
- 6) When the HFO detection method has finished running, a MATLAB file struct with the extension .rhfe is saved in the Analysis directory of RIPPLELAB.
- 7) This .rhfe file can be opened in MATLAB or RIPPLELAB again to extract results.

For more detailed instructions on full use of RIPPLELAB, the open source user manual should be consulted [30]. In this study, the modified Burnos method in RIPPLELAB was used to detect HFOs in iEEG, and data from the resultant .rhfe file was extracted for calculating the rate of HFO preponderances for each channel over time.

7. The AR_m Method in MATLAB

In order to apply the AR_m method to iEEG and investigate the results graphically, implementation was done in MATLAB. The Original AR_m method from Geertsema *et. al* was chosen [11], and the method was applied to long-term iEEG signal recordings. A flowchart of the AR_m process is shown in Figure 25. This process was followed for each channel in each iEEG file. $AR(3)$ represents the 3rd order AR model, where the previous 3 signal samples are used to calculate the model.

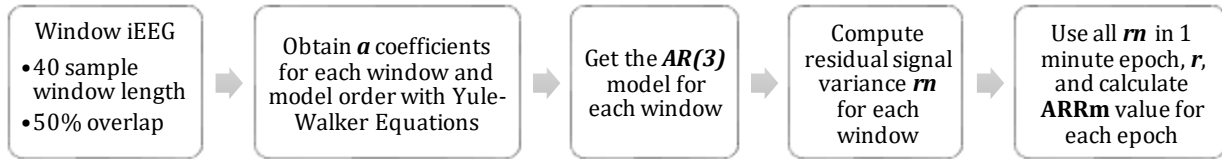


Figure 25: AR_m Method Process

A 3rd model order was chosen, because it had been shown in a preliminary investigation of multiple patient datasets that a 40 sample window length and model order of 3 provided the best distinction between channels inside and outside the SOZ [27]. With a 40 sample window length used for each AR model, the lowest oscillating frequency that can be modeled in the window, using a sampling frequency of 1000Hz is $(1/40) * 1000 = 25\text{Hz}$, 50Hz for a sampling frequency of 2000Hz. Therefore, the model is quantifying the high frequency components in the signal, and lower frequencies will not influence the residual signal variation. For each window, the a

coefficients calculated via Yule Walker Equations were obtained through the MATLAB function *aryule()*. Yule Walker Equations are based on the principle of autocorrelation, and the *aryule()* function uses Levinson-Durbin recursion to find the coefficients of a signal based on the signal's autocorrelation sequence. This method provides a least mean square fit of the model to the original signal. Using these a coefficients, the 3rd order model signal was computed using Equation (2). An example of the $AR(3)$ model fit for 1 window is shown in Figure 26. Windows were 40 sample (s) in length and had 50% overlap (20 samples) with adjacent windows.

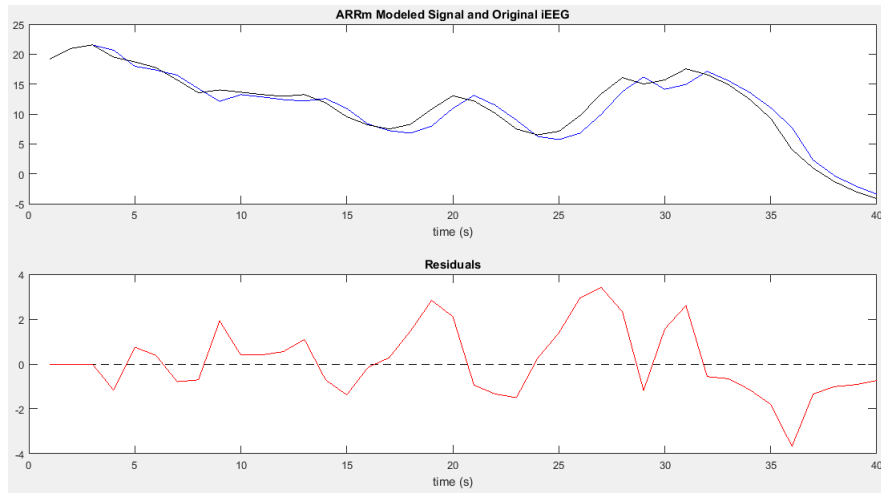


Figure 26: 3rd Order Model Fit (blue) to Original iEEG Signal (black) and Residual of the Model

Variance for the residual in the 3rd order model (r_m) was calculated for each window. Then, r_m values for the number of windows in a 1 min. epoch of iEEG data were grouped in a vector of data, r . The ARRm value was calculated using the r values for every 1 min. epoch by dividing the standard deviation of the residual variances by the mean of the residual variances (Equation (3)), with 50% overlap (30 seconds). The final matrix for a singular iEEG file contains ARRm values per 1 min. epoch over time for each channel. The non-harmonicity is quantified for each channel, for each minute of time by following this process.

8. HFO/ARRm Analysis Tool

To visualize and compare the rate of HFOs and the ARRm value over time, a tool was developed in MATLAB to aid analysis of the results. This tool can also be used to draw observational correlations between the ARRm value and HFO rate at any particular point in time.

8.1 User Interface

The tool first prompts the user to import a .mat file saved from ARRm analysis and a .rhfe file from HFO analysis before launching. The user should choose the ARRm and HFO file that correspond to the same iEEG .edf file analyzed. After import, all relevant information from the files is extracted. In order to highlight channels identified by the epileptologist as exhibiting epileptogenic characteristics, channels of interest based on comments in the patient notes are defined in a cell vector, and in code the appropriate patient is selected according to the files imported. This should be changed as necessary (Figure 27).

```
%WDH-022  
COI_patients.WDH_022 = {'B2','B3','B4','D1','D3','D4','C11','C12','C18','C19','D14'};  
  
%choose the COI labels for appropriate patient  
COI = COI_patients.WDH_022;
```

Figure 27: Example of Definition and Selection of Channels of Interest

Finally, the analysis tool launches. The overall user interface is defined in Figure 28.

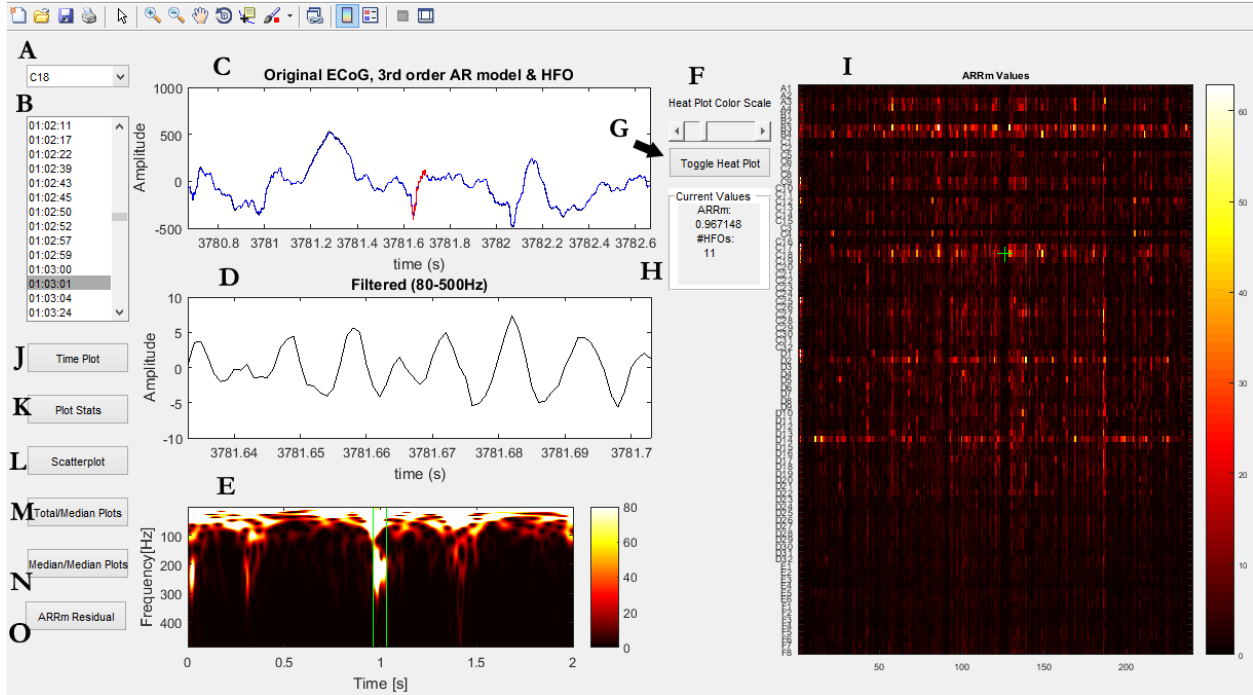


Figure 28: HFO /ARRm Analysis Overall User Interface

The user can start by selecting a channel they wish to investigate from the drop down menu (Figure 28, A). Once a channel is selected, the listbox below (Figure 28, B) populates with the times (HH:MM:SS) of each HFO detected in that channel for the iEEG file. When a time is selected, the first plot (Figure 28, C) shows the HFO in the signal at that time (red), and the iEEG signal +/- 1s around the HFO. The iEEG in this plot shows the original iEEG signal (black) and the 3rd order AR modeled signal (blue). The second plot (Figure 28, D) shows the HFO identified in the iEEG filtered between 80-500Hz. The third plot (Figure 28, E) shows the power spectral density of the frequency content in the signal plotted in (Figure 28, C). In this plot, the separation in power of the HFO from the base frequencies can be examined, which is a key element of HFO classification (Figure 20). The plot on the right (Figure 28, I) shows the matrix of ARRm values or HFO values over time for each channel, and the HFO/ARRm plot view can be toggled by the button (Figure 28, G). The

heat map represents the channels on the y-axis, and 1 min., 50% overlapped epochs on the x-axis. “Hot” colors (white, yellow, bright orange) represent high $ARRm$ values or high HFO counts. A green cross is located on this plot at the current 1 min. epoch wherein the selected time (Figure 28, B) lies, and the $ARRm$ value and HFO count for the current epoch is shown in (Figure 28, H). The color scale for this plot can be adjusted with the scrollbar (Figure 28, F).

8.2 Analysis

The buttons on the left side of the tool generate different plots for analysis. The *Time Plot* button (Figure 28, J) generates a line plot of $ARRm$ /HFO values over time for the selected channel (Figure 29). This plot also denotes if the selected channel is a channel of interest.

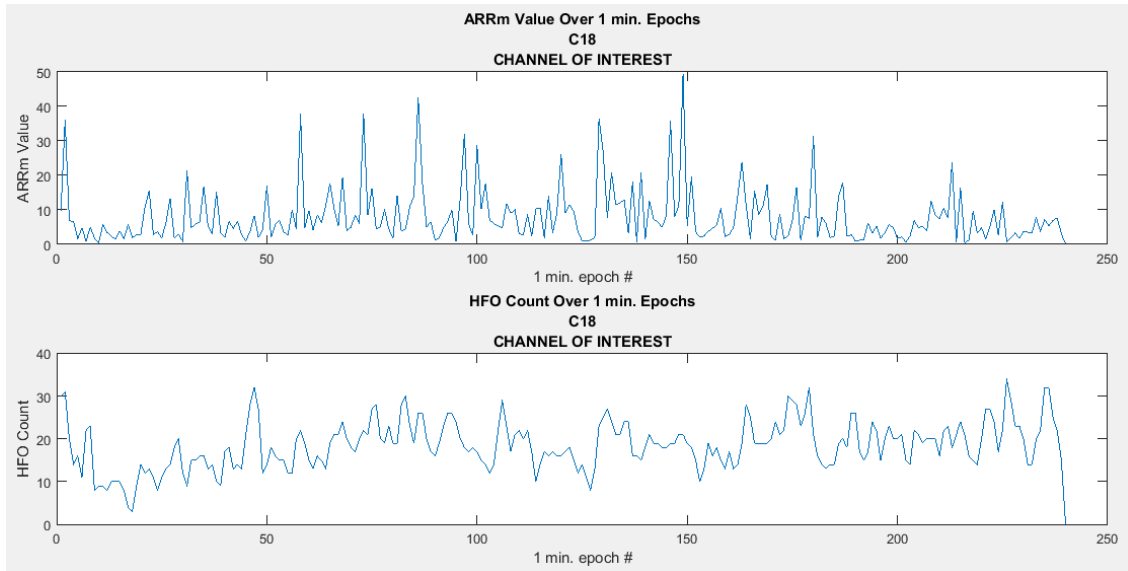


Figure 29: $ARRm$ /HFO Value Over Time for Selected Channel

The *Plot Stats* button (Figure 28, K) generates a plot that shows a boxplot for all $ARRm$ /HFO values over time for each channel (Figure 30). Each boxplot has a box where the central red mark indicates the median, the bottom and top edges of the box indicate the

25th and 75th percentiles, respectively, the whiskers extend to the most extreme data points, and the “+” symbol represents large outliers beyond extreme data points.

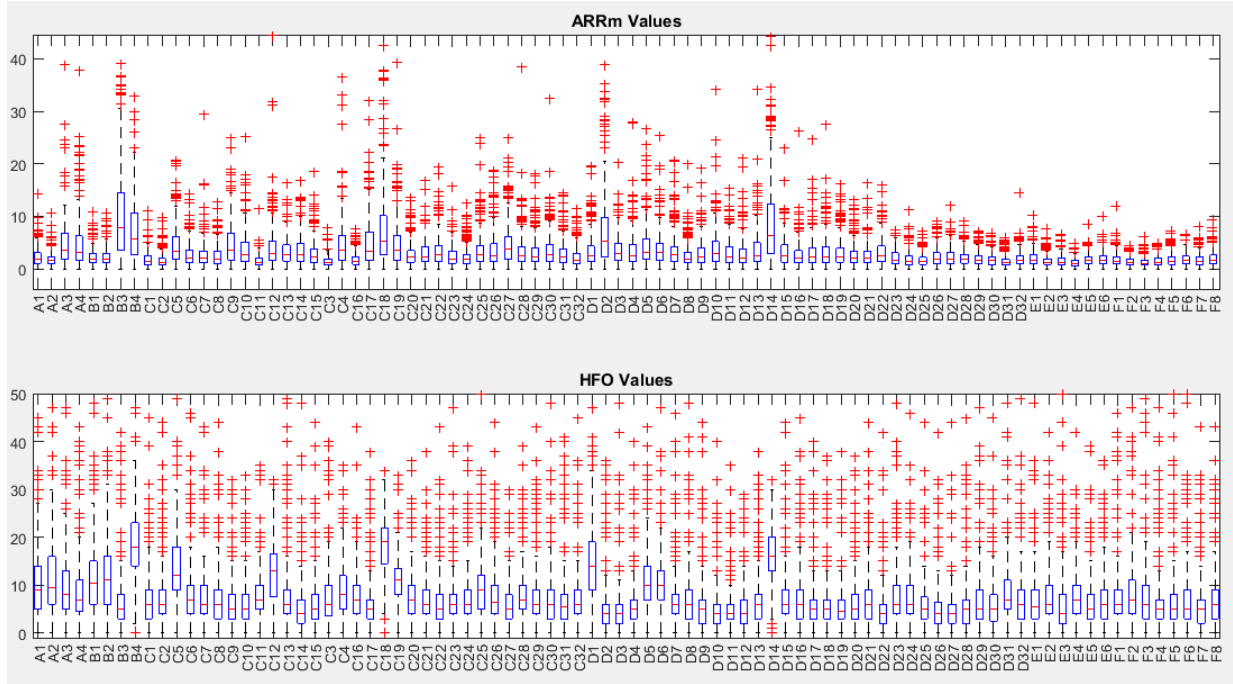


Figure 30: Boxplots of ARRm/HFO Values for each Channel

The *Scatterplot* button (Figure 28, L) generates a plot of ARRm values vs. HFO values for the selected channel (Figure 31). This plot also denotes if the selected channel is a channel of interest. This plot is meant to visualize the relationship between high ARRm values and high HFO count values, because each point on the plot represents the same time epoch for each value.

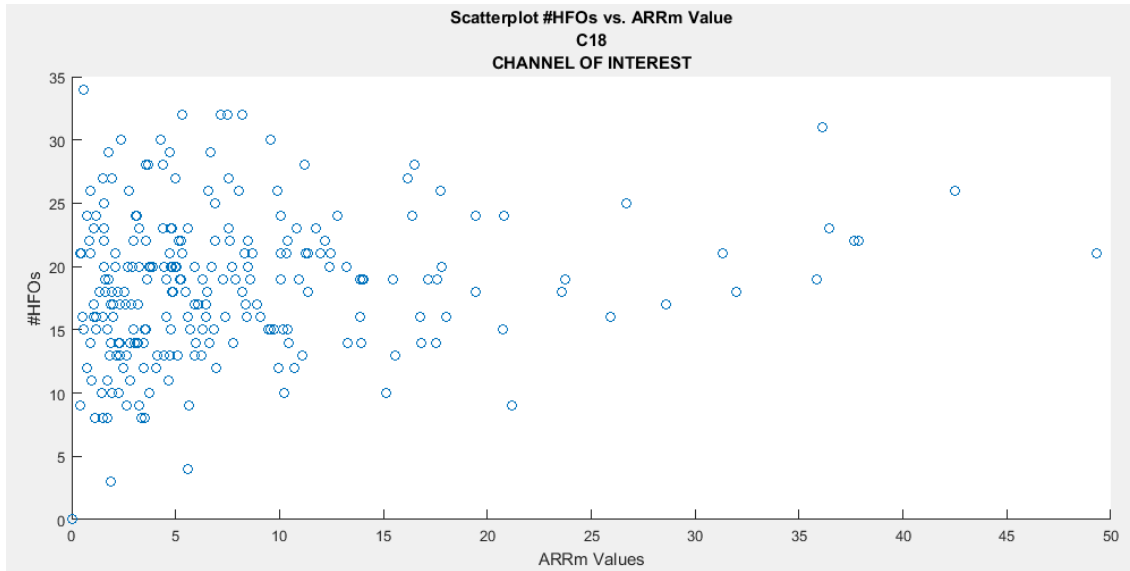


Figure 31: Scatterplot of ARRM Values vs. HFO Count Values

The *Total/Median Plots* button (Figure 28, M) generates plots of the Total HFO count and the median ARRM value for each channel for the duration of the recording (Figure 32).

Channels of interest are highlighted with a red asterisk at the peak. From this plot, certain channels should stand out that have consistently high ARRM values or a large number of HFOs. Correlations between high preponderances of HFOs and high ARRM values could be drawn from this plot.

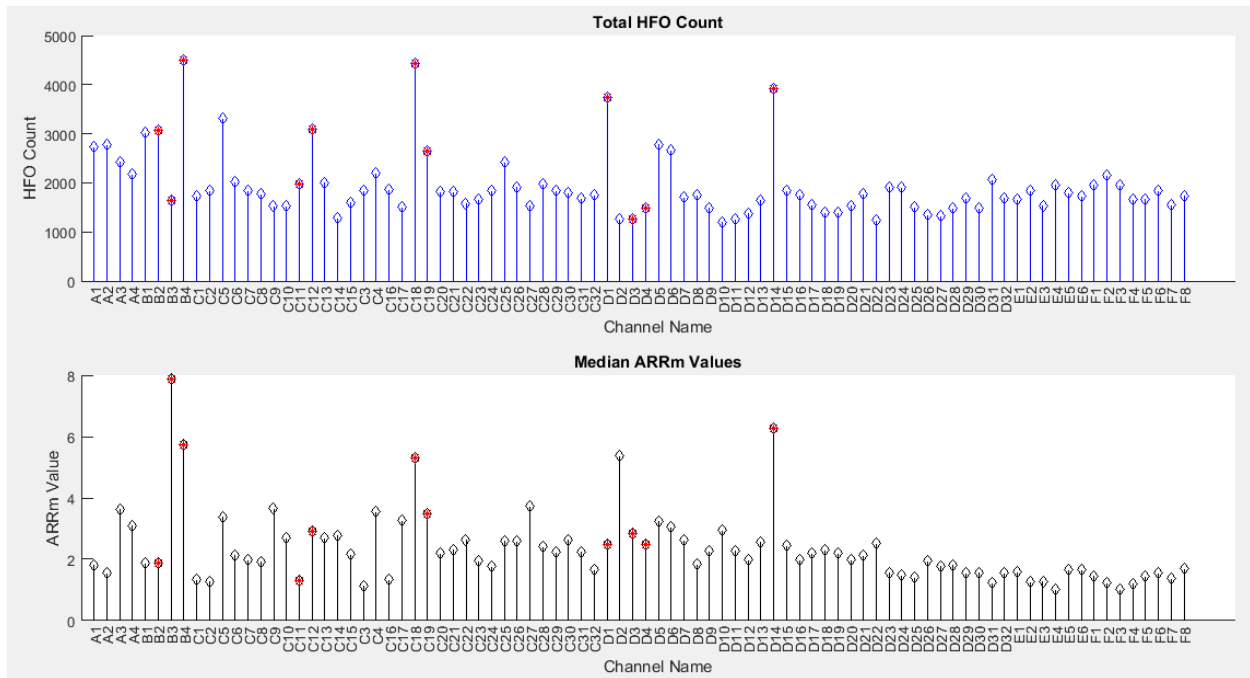


Figure 32: Total HFO Count/Median ARRM Value Plots

The *Median/MedianPlots* button (Figure 28, N) generates plots of the median HFO count and the median ARRM value for each channel (Figure 33). Channels of interest are highlighted with a red asterisk at the peak. From this plot, certain channels should stand out that have consistently high ARRM values or HFO counts. Visual correlations between consistently high numbers of HFOs and high ARRM values could be drawn from observing this plot.

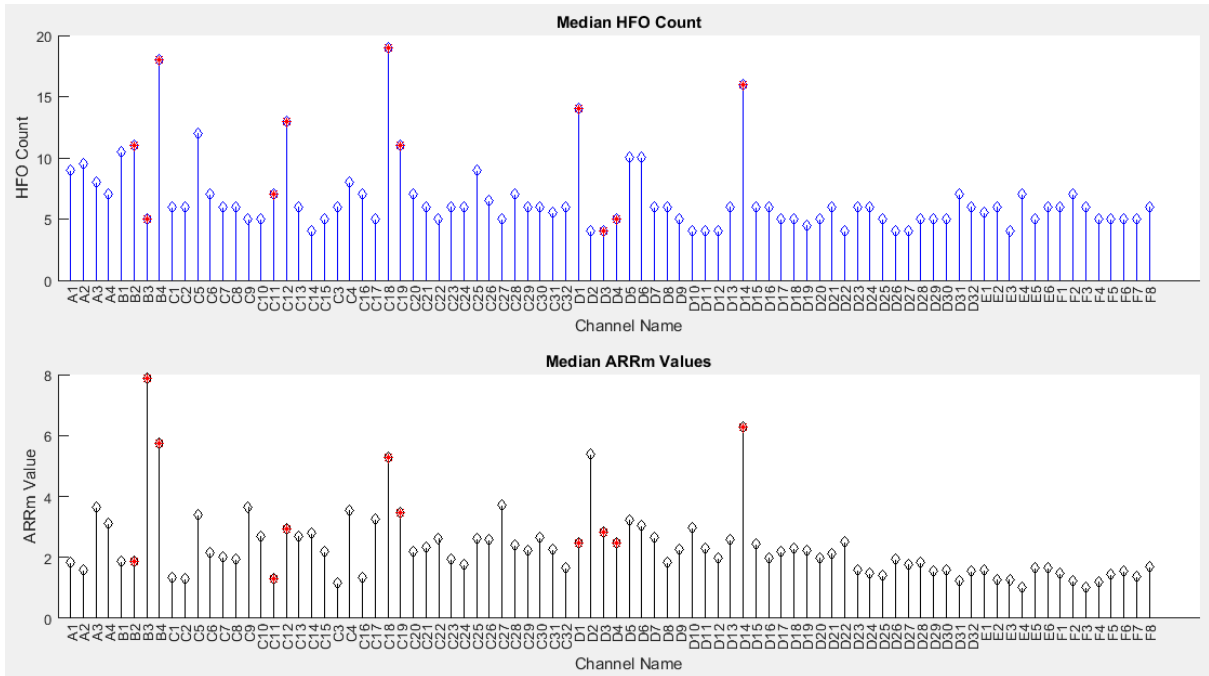


Figure 33: Median HFO Count/Median ARRm Value Plots

The *ARRm Residual* button (Figure 28,O) generates a figure of subplots including the original and modeled EEG at the current selected HFO time, the ARRm residuals, and the power spectral density of the frequency content in the signal (Figure 34). This plot can be used to investigate the relationship between the ARRm value, which is based on the residual signal variation, and epileptic events in the signal.

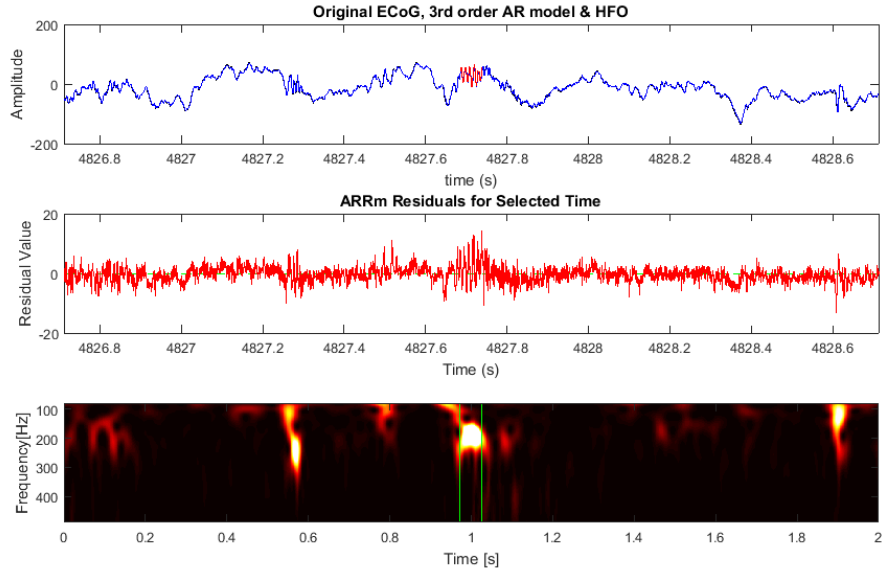


Figure 34: ARRm Residuals Plot

This tool provides multiple outlets for examining the results of HFO detection and ARRm value analysis of any two iEEG files.

9. Statistical Analysis

The correlation between the ARRm value and HFO count over time for respective channels was calculated statistically. The correlation measurement represents the level of similarity between the ARRm value and HFO counts per epoch for respective channels of the same iEEG file. The correlations were performed using the MATLAB function *corr()* [31]. With two matrices as input, the function returns a pairwise correlation coefficient matrix, which contains the correlation coefficient between each pair of columns in the input matrices. The input matrices were normalized between 0 and 1, then transposed so that channels represented each column and were considered variables, and the epoch values were represented in each row, considered observations. The function also returns a matrix of p-values for testing against the null

hypothesis of no correlation. The p-value represents the probability that the given data would have arisen if the null hypothesis were true. Each element of the p-values matrix is the p-value of the corresponding element in the correlations matrix. If the p-value is less than 0.01, then the correlation is significantly different from zero. Therefore, the values on the diagonal of the coefficient matrix and p-value matrix represented the correlations between channels (columns) of the same name (index). For instance, the correlation between column 2 of the first input matrix and column 2 of the second input matrix would exist at row 2, column 2 of the correlation coefficient matrix, and the p-value at row 2, column 2 of the p-value matrix.

The MATLAB function *corr()* has the option to execute either a Pearson or Spearman correlation between two matrices. Pearson's linear correlation coefficient assigns values from -1 and 1 for each correlation between two channels. For the Pearson correlation, it is assumed that both variables should be normally distributed. Spearman's rank correlation coefficient is a nonparametric measure that is the Pearson correlation between the rank values of two variables. For the Spearman correlation, it is assumed that at least one of the variables is ordinal, and in this case the HFO count is ordinal. Both Pearson's linear correlation and Spearman's rank correlation tests were recommended by the advising statistics professor on this thesis. Best case scenario, the ARRM value and HFO count are linearly related, but since this is not known and the relationship could also be nonlinear, the nonparametric test also was recommended. The Spearman rank correlation is most appropriate for this set of data, but the Pearson linear correlation should also be executed to check for possible significant linear relationships. In the correlation value results, a value of 1 indicates a perfect positive correlation, and a value of -1 indicates a perfect negative correlation, meaning two variables are inversely related. Correlations between channels were tested with a significance level of 0.01, which means that channels with correlations were considered significant if the p-value was < 0.01 .

10. Real-Time ARRM Method on Embedded Platform

In order to test the ability of the ARRM method to analyze EEG data in real-time, it was implemented in an embedded system application. The scope of this project was to design the software according to the limitations of the hardware. Therefore, a smaller sampling frequency of 250Hz was used, the ARRM value was calculated over an epoch of 30s, and only 1 channel was analyzed for proof-of-concept runtime analysis with limited RAM. This system was designed to receive scalp EEG signals from a human test subject, amplify and convert the analog signals to digital values, send the digital values to a microcontroller, process the data with the ARRM method as data is collected, and export data to a PC application for display. An outline of the hardware is shown in Figure 35.

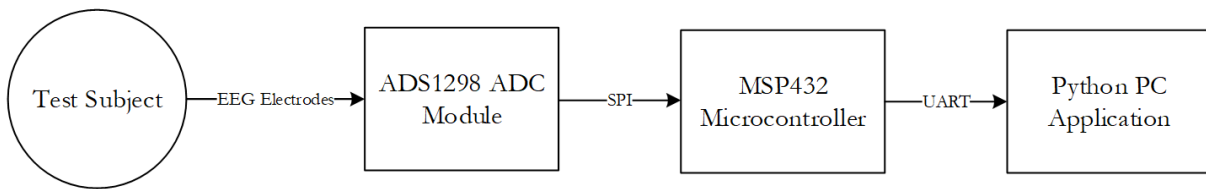


Figure 35: Hardware Block Diagram for Real-Time EEG Collection/ARRM Analysis System

The A/D module amplified EEG biopotentials and then sent serial messages containing the digital values of the signal to a microcontroller. The A/D module chosen was the ADS1298 and the microcontroller was the MSP432. The software was implemented in the language of C++ on the microcontroller, and multiple classes were created for communication with the A/D module and processing with the ARRM method. The main flow of the software design is shown in Figure 36.

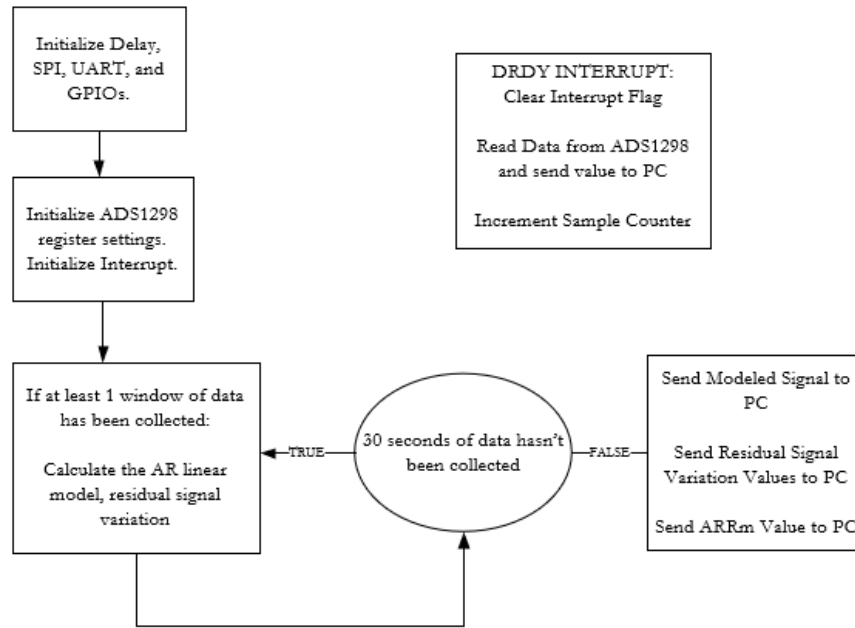


Figure 36: Software State Diagram for Real-Time EEG Collection/ARRm Analysis System

For testing of the system with a human test subject, a scalp EEG electrode was placed on the surface of the skin on the occipital lobe region of the head, the reference electrode was placed on the forehead, and the ground electrode was placed on the ankle. Recording of the signal began when the subject's eyes were closed and the subject was relaxed. 10Hz alpha waves were recorded, and the signal was accurately modeled by the 3rd order autoregressive model (Figure 37).

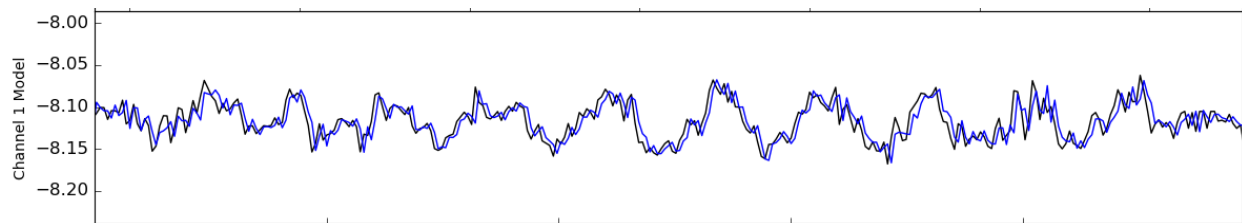


Figure 37: Modeled EEG Signal (Blue) and Original EEG Signal (Black)

Residual signal variation was tested by recording an ECG signal, where the QRS spikes were predicted to cause a spike in the residual signal variation. This hypothesis was confirmed (Figure 38).

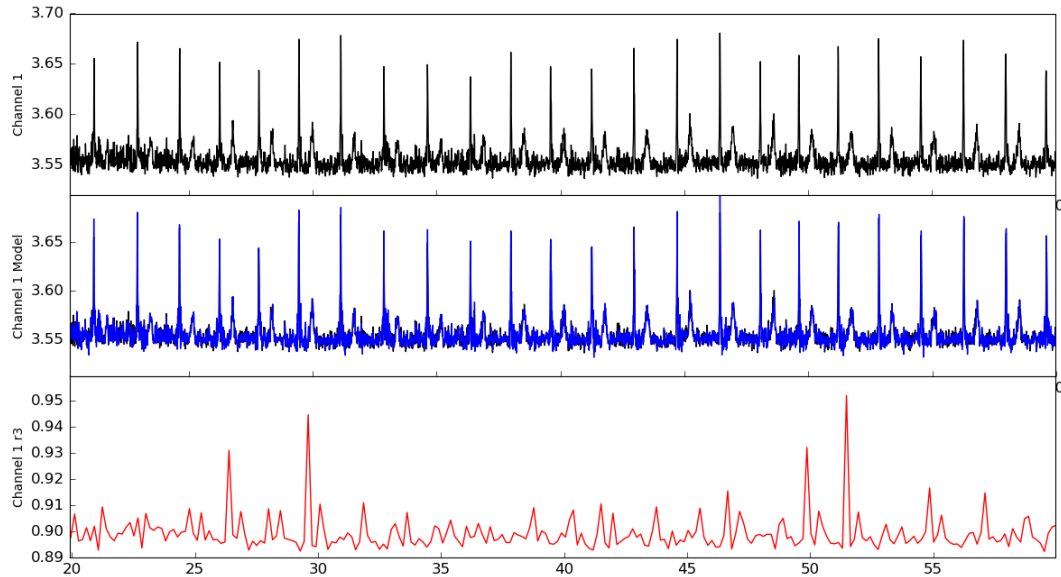


Figure 38: ECG Signal, Modeled Signal, and Residual Signal Variation

For further studies on this system, a simulated epileptic EEG signal with HFOs injected should be used as input to the system. This low-level proof-of-concept system implementation has potential to be scaled up to use sampling frequencies of 1000-2000Hz and perform continuous collection/analysis to only output the ARRm value to a display. This framework could be expanded to be implemented as a software package in an existing EEG machine or it could be developed into a standalone EEG medical device system for use in the EMU or intraoperatively. The complete Software Requirements Specification, Software Design Document, and code can be referenced in the Appendix.

Chapter 4: Results

11. Data

Data for this study was sourced from the Spectrum Health EMU, Grand Rapids, MI, in partnership with Spectrum Health physicians and Grand Valley State University's Biomedical Engineering program. The data that was provided does not contain any patient information, and the usage of this data has been approved by the Spectrum Health Institutional Review Board. Patients who were surgical candidates for epilepsy surgery had undergone electrode implantation surgery, and iEEG files were recorded continuously for each patient during their EMU stay. The durations of these studies ranged from 7 to 13 days in the EMU. The Nihon Kohden EEG-110A V01.00 EEG system was used to record and log the data. iEEG data files are saved in .edf format, and range between 500Hz and 2000Hz in sampling frequency. 3 patients were selected to study, and multiple groups of sequential files were chosen from the beginning, middle and end of the EMU stay for analysis. All data files and their related information can be viewed in Table 3. A red box <these will be red in final manuscript, when all fields are updated it removes the red so I am waiting until the final field update to add the red back in> in the "Seizure" column indicates there is seizure activity recorded in that file.

Table 3: Patients and Data Files from Spectrum Health EMU Analyzed using ARRM and HFO Detection

Patient	iEEG Filename	Date of iEEG Recording (MM/DD/YYYY)	Start Time of iEEG Recording (HH:MM:SS)	Fs (Hz)	Seizure
WDH-022	BA2681LD_1-1.edf	10/22/2014	14:07:02	1000	
	BA2681LE_1-1.edf	10/22/2014	16:07:14		
	BA2681LF_1-1.edf	10/22/2014	18:07:25		
	BA2681MQ_1-1.edf	10/26/2014	09:01:14		
	BA2681MR_1-1.edf	10/26/2014	11:01:25		
	BA2681NQ_1-1.edf	10/28/2014	21:52:12		
	BA2681NR_1-1.edf	10/28/2014	23:52:23		
DJA-033	AA536174_1-1.edf	11/04/2015	07:02:47	2000	
	AA536175_1-1.edf	11/04/2015	10:02:59		
	AA536176_1-1.edf	11/04/2015	13:03:11		
	AA536177_1-1.edf	11/04/2015	16:03:22		

	AA53617L_1-1.edf	11/06/2015	10:06:03		
	AA53617M_1-1.edf	11/06/2015	13:06:15		
	AA53617N_1-1.edf	11/06/2015	16:06:26		
	AA536181_1-1.edf	11/08/2015	05:07:53		
	AA536182_1-1.edf	11/08/2015	08:08:04		
JMM-025	AA5320XH_1-1.edf	06/11/2015	13:34:24	2000	
	AA5320XJ_1-1.edf	06/11/2015	14:27:35		
	AA5320Y0_1-1.edf	06/13/2015	00:30:51		
	AA5320Y2_1-1.edf	06/13/2015	00:58:03		
	AA5320YE_1-1.edf	06/14/2015	01:00:22		
	AA5320YF_1-1.edf	06/14/2015	03:00:33		
	AA5320Z2_1-1.edf	06/16/2015	01:04:57		
	AA5320Z3_1-1.edf	06/16/2015	03:05:08		
	AA53210G_1-1.edf	06/20/2015	05:14:48		
	AA53210H_1-1.edf	06/20/2015	07:15:00		
	AA53210I_1-1.edf	06/20/2015	09:15:11		

12. ARRM Value and HFO Counts Over Time

Heat plots were image plots generated for the ARRM value over time for each channel, and the HFO count over time for each channel. The plots for each method for the same file represented the same duration of time. The maximum time value on the time axis on the heat plots represents 2 times the number of 1 minute epochs in the file, because the ARRM value calculation and HFO count was done for each minute (epoch), overlapping 50% (30s) with adjacent epochs. Hot colors in the heat plot (white, yellow, bright orange) represent high ARRM values or HFO counts. In previous observations of HFO detection, it was noted that consistently high HFO counts were seen over time on channels of interest. This pattern was also observed for the ARRM value, and direct visual comparisons of the two methods are described in the following sections for each patient. The focus in observation is the inter-ictal activity and pre-ictal activity for channels of interest.

12.1. Patient WDH-022

This patient had a history of anoxic brain injury (brain was deprived of oxygen) at birth, which resulted in a left hemispheric encephalomalacia (loss of brain

tissue), and weakness on the right side of the body. The patient underwent phase II implantation surgery of subdural grids and strip electrodes. Location of the electrodes can be referenced in Table 4.

Table 4: Patient WDH-022 Electrode Placement

Electrode Channels	Electrode Type	Location
A1-4	1x4 Strip	Left parietal convexity, posterior to grid D. Contact 1 posterior and contact 4 was anterior.
B1-4	1x4 Strip	Left parietal convexity, inferior to strip A. Contact 1 was posterior and 4 was anterior.
C1-32	4x8 Grid	Left frontoparietal convexity. Contact 1 posterior & superior, 8 posterior & inferior, 25 anterior & superior and 32 anterior & inferior.
D1-32	4x8 Grid	Left frontal convexity. Contact 1 posterior & superior, 8 posterior & inferior, 25 anterior & superior and 32 anterior & inferior.
E1-8	1x8 Strip	Left inferior frontal/orbitofrontal region. Contact 1 posterior and contact 4 was anterior.
F1-8	1x8 Strip	Left temporal region. Contact 1 anterior and contact 8 was posterior.

The sampling frequency for this patient was 1000Hz. Some seizures were recorded on the files included in this analysis. Seizure times, notes, and files effected can be referenced in Table 5.

Table 5: Patient WDH-022 Seizure Times, Notes and Files

Seizure Time	Seizure Notes	File Affected
14:14:30	Sharp transient at C12>13 followed by 120-130Hz activity at C12>C13. Spread to B1-4, D1,2 C19, 20 and gradual decrease in frequency.	BA2681LD
14:27:00	Similar to previous event.	BA2681LD
14:29:42	110-120Hz Activity at C18 and B1-3, very quick spreads to C4, C12, D1 then D5,	BA2681LD

	D14. The ictal discharge evolved into lower frequency.	
14:30:00	Sharp transient at B1-3, followed by 120-130Hz activity at B1-3, C18, and D1.	BA2681LD
14:31:17	Fast activity first seen at B2-3, C18-19, and D1.	BA2681LD
14:34:21	A sharp transient is seen at D14, followed by 130-140Hz activity, almost immediately involving B2-3, D1-2, C18-19.	BA2681LD
17:51:45	120Hz activity at C12. Spreads to D1 and B1-3 then C19, then C25-26. The ictal discharge evolves into lower frequency. Fast activity persists at C10-12.	BA2681LE
18:00:00	Sharp transient at B1-3, followed by 120-130Hz activity at B1-3, C18, and D1.	BA2681LE
18:01:00	Onset was with a sharp transient followed by > 130Hz fast activity at B1-3, C18-19, C10-11, D1-2 and D13-14	BA2681LE
18:03-04:00	Sharp transient at B1-3 followed by gamma activity at B1-3, C18, D1.	BA2681LE
18:06:00	120Hz activity at B1-3 > C18-19 > F1-2, C10-12 then D14	BA2681LE
18:07:00	Gamma activity at B2-3, C18-19, D1	BA2681LE
19:10:00	>140Hz activity at C12	BA2681LF
09:41:00	Ictal discharges starting with gamma activity at C18-19 > B2-3 with rapid spread to C12, D1, D14	BA2681MQ
12:13:00	Onset with gamma activity at B2-4, C18>D1, quick spread to C4 and C12. Activity evolved into lower frequency. Some seizures had a more restricted onset involving B3-4 only.	BA2681MR
12:28:00	Ictal discharges starting with gamma activity at B3-4 with spread to C17-19 and D1-2> D21-22.	BA2681MR
12:45:00	Sharp wave at D14>D5 followed by >200Hz activity gradually decreasing in frequency.	BA2681MR
12:51:00	Gamma activity starting at C18, spreading to C11-12, C2-5, B2-4, D1-4.	BA2681MR

Channels of interest identified by the epileptologist according to patient notes are defined in Table 6.

Table 6: Patient WDH-022 Channels of Interest Identified by Epileptologist

Channel Names	Notes
B2-4, D1-5, C11-C12, C18-19, D14	“Subclinical ictal discharges seen with onset at D3-4 or C18-19, or a wide field involving both. Frequent periodic spikes and bursts of high frequency activity at D14. Frequent spikes at D3-4, C18-19, C12, D1 ... Some seizures had an onset localized to B3-4... Several mild motor seizures were recorded and suggest a SOZ superior to the encephalomalacia (C12 in some seizures and wider onset involving D2, C18, B3”

The seizure onset zones were determined to be primarily in grids C and B, and it was decided that due to overlap with primary sensory and motor cortex, surgical resection was ruled out for those areas. It was decided to proceed with implantation of a NeuroPace RNS, and perform limited resection in the inferior frontal region underneath electrodes D14 and D5.

12.1.1 ARRm Value

The ARRm value was calculated and displayed for all channels over time in a heat plot. The y-axis represents the electrode channels and the x-axis represents time in 1 min. epochs, 50% overlap. The lighter shades of hot colors (white, yellow, bright orange) indicate epochs with high ARRm values (see plot color scales), and channels of interest are indicated with red arrows on the channels axis. The group of files at the beginning of the EMU stay are plotted in Figure 39.

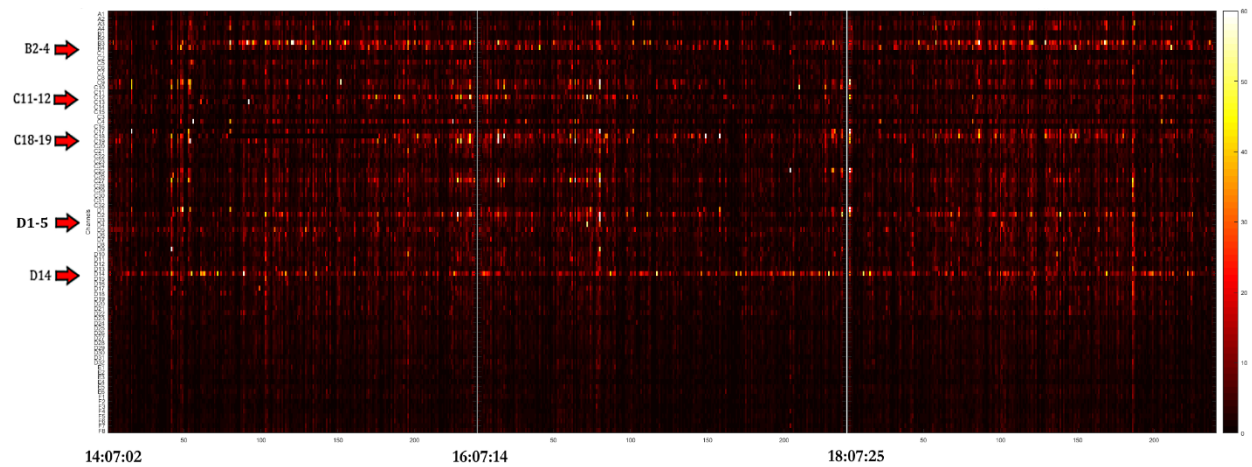


Figure 39: ARRm - Sequential Files for the Beginning of EMU Stay (10/22/2014) and Channels of Interest

Consistently high ARRm values can be observed over the full 6 hour interval of these recordings on channels of interest. There also appears to be higher ARRm values across all channels near the seizure activity that was noted in the patient records occurring at 14:14-14:40. The group of files recorded in the middle of the EMU stay are plotted in Figure 40.

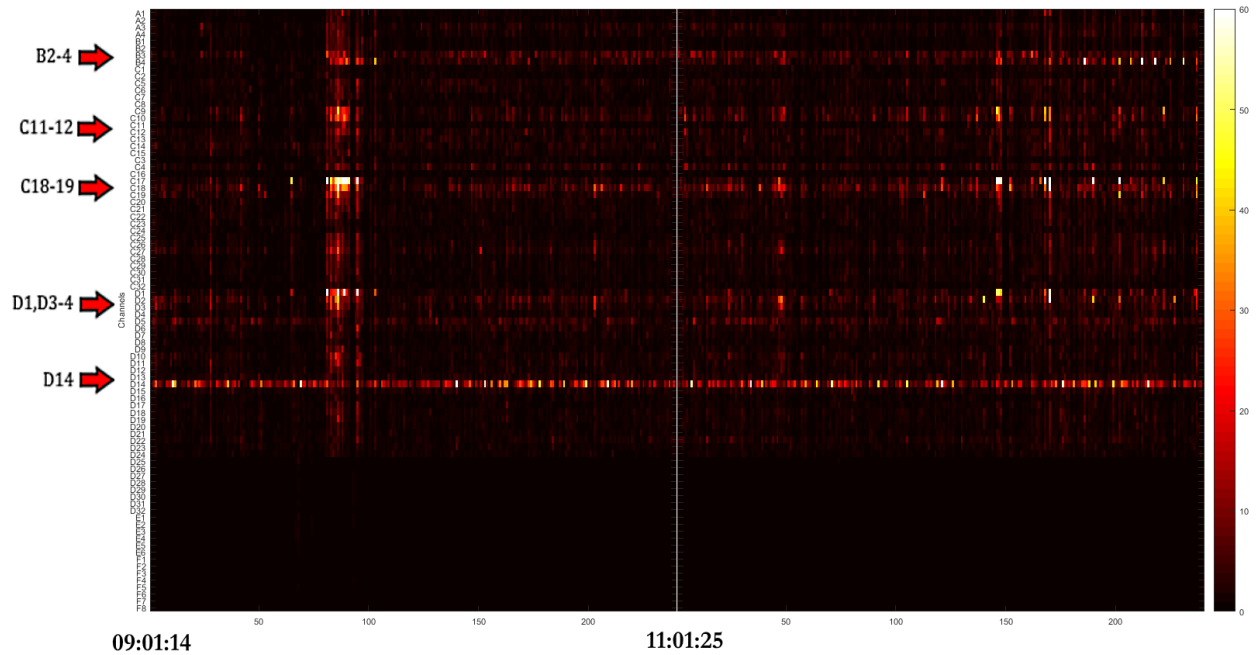


Figure 40: ARRm - Sequential Files for the Middle of EMU Stay (10/26/2014) and Channels of Interest

Similar observations can be made about the channels of interest with consistent high activity across time for these files as well. There is high activity occurring on channels of interest, with fields spreading to nearby channels at approximately 9:30:00 – 9:40:00 across all channels just prior to when the first seizure occurred at 09:41:00. High activity can also be seen on channels of interest, with fields spreading to other nearby channels between approximately 12:13:00 – 12:51:00 when multiple seizure events occurred in that time frame. The group of files recorded towards the end of the EMU stay are plotted in Figure 41.

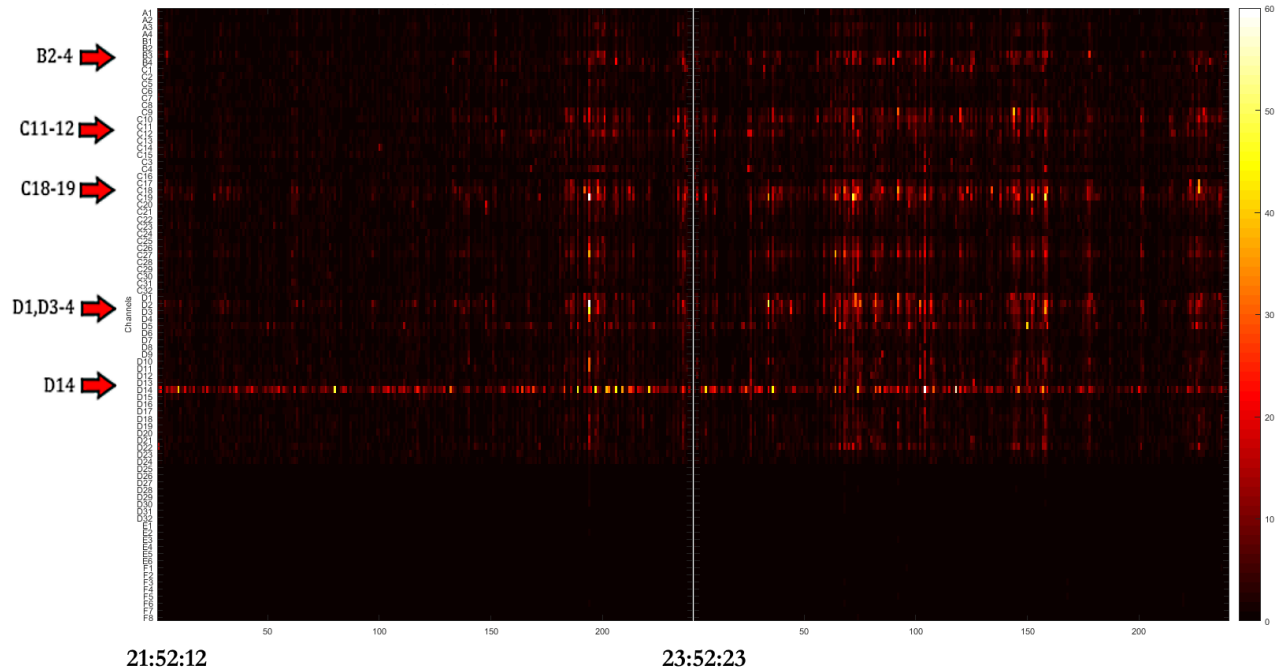


Figure 41: ARRM - Sequential Files for the End of EMU stay (10/28/2014) and Channels of Interest

No seizure events were recorded in the notes for these two files. Consistent activity can still be observed over time on channels of interest. In the second half of this time frame, high values can be seen on channels of interest at the same time, with fields spreading to nearby channels. This activity could indicate inter-ictal discharges. Multiple instances of this activity can be noted between approximately 23:30:00 - 25:10:00.

12.1.2 HFO Count

The HFO count was calculated and displayed for all channels over time in a heat plot. The brighter shades of hot colors indicate epochs with high HFO counts, and channels of interest are indicated on the channels axis.

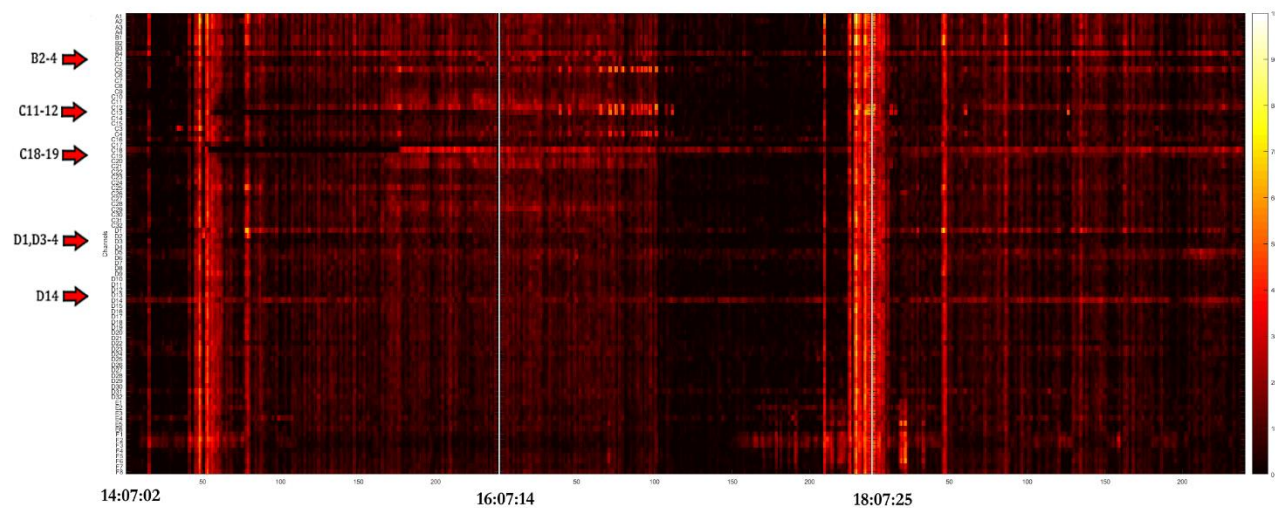


Figure 42: HFO - Sequential Files for the Beginning of EMU Stay (10/22/2014) and Channels of Interest

Consistently high values can be observed over time on channels of interest, and there also appears to be high values across all channels at once in multiple instances, especially during times of seizure activity at 14:14-14:30, 17:51, and 18:00.

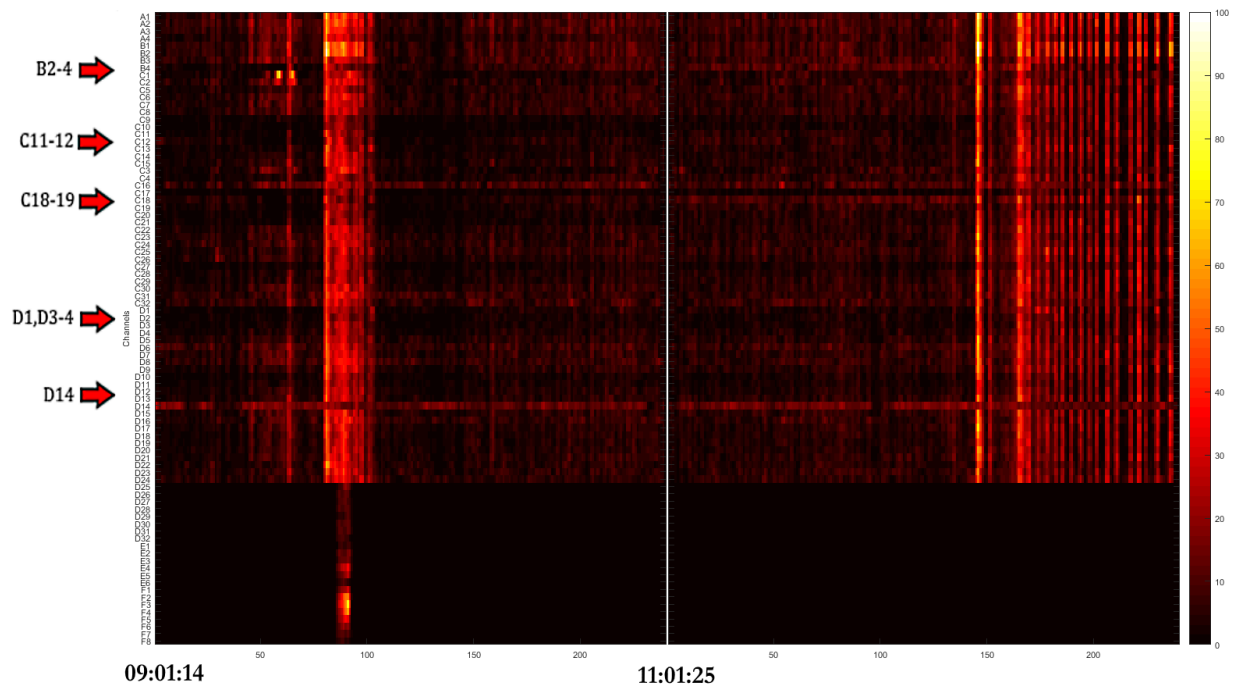


Figure 43: HFO - Sequential Files for the Middle of EMU Stay (10/26/2014) and Channels of Interest

Consistently high values can be observed over time on channels of interest, and there also is especially high activity across all channels during the times of seizure events.

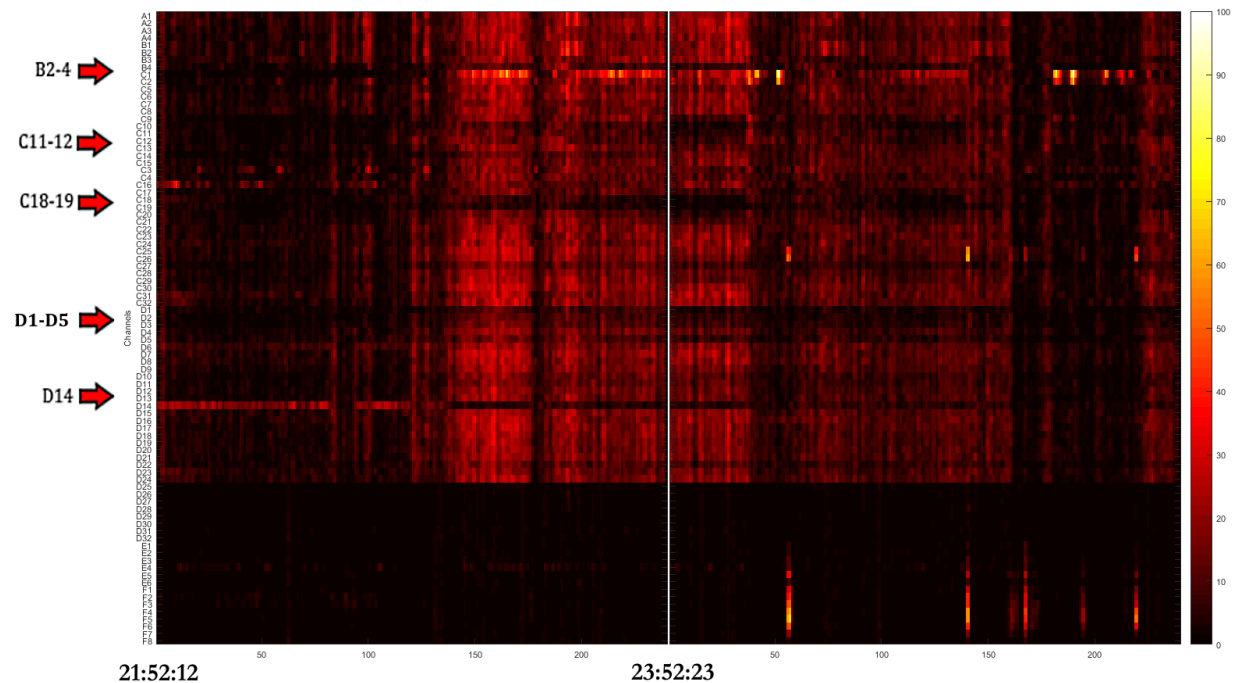


Figure 44: HFO - Sequential Files for the End of EMU stay (10/28/2014) and Channels of Interest

Some consistently high activity can be observed on channels of interest over time, but there are more instances of high activity across all channels.

12.2. Patient DJA-033

This patient with medically intractable epilepsy presented with a semiology characterized by a complex partial manifestation of an aura and often a ringing in the left ear. A previous evaluation had identified epileptogenicity in the superior temporal region and parahippocampal region. Though stable, concern was raised that there may be a single focus of the seizures, requiring a further evaluation. The patient underwent phase II implantation surgery of depth electrodes using the ROSA®. Electrode locations are noted in Table 7.

Table 7: Patient DJA-033 Electrode Placement

Electrode Channels	Electrode Type	Location
A1-6	1x6 Depth	Superior posterior temporal gyrus. Contact 1 most mesial and contact 6 most lateral.
B1-8	1x8 Depth	Superior middle temporal gyrus. Contact 1 most mesial and contact 8 most lateral.
C1-8	1x8 Depth	Superior anterior temporal gyrus. Contact 1 most mesial and contact 8 most lateral.
D1-6	1x6 Depth	Anterior hippocampal. Contact 1 most mesial and contact 6 most lateral.
E1-8	1x8 Depth	Posterior hippocampal. Contact 1 most mesial and contact 8 most lateral.

The sampling frequency for this patient was 2000Hz. Some seizures were recorded on the files included in this analysis. Seizure times, notes, and files effected can be referenced in Table 8.

Table 8: Patient DJA-033 Seizure Times, Notes and Files

Seizure Time	Seizure Notes	File Affected
11:58:26	Rhythmic 15-20Hz sharply contoured activity seen at B6,B7,B8. Rhythmic sharp waves seen in B and C grids.	AA536175
11:14:40	Rhythmic 15-20Hz sharply contoured activity seen at C3,C4,B7, B8. Rhythmic sharp waves seen throughout B and C grids.	AA53617L
17:45:25	Rhythmic 15-20Hz sharply contoured activity seen at C3,C4,B7,B8. Rhythmic sharp waves seen throughout B and C grids.	AA53617N
7:48:41	Rhythmic 15-20Hz sharply contoured activity seen at B7,B8. Rhythmic sharp waves seen throughout B and C grids.	AA536181

Channels of interest identified by the epileptologist according to patient notes are defined in Table 9.

Table 9: Patient DJA-033 Channels of Identified by Epileptologist

Channel Names	Notes
B3-5, C1-C6, E1-2	“Interictally, sharp waves were seen at C2, C3, C4 > C5 > C1, C6. Sharp waves were also seen occasionally at E2>E1. Sharp waves were seen less frequently at B3 > B4,B5.”

It was determined that this patient suffers from localization related epilepsy as evident by the interictal and ictal recordings. The patient notes do not describe the proceedings from this patient’s EMU study.

12.2.1 ARRM Value

The ARRM value was calculated and displayed for all channels over time in a heat plot. The lighter shades of hot colors indicate epochs with high ARRM values, and channels of interest are indicated on the channels axis. The group of files at the beginning of the EMU stay are plotted in Figure 45.

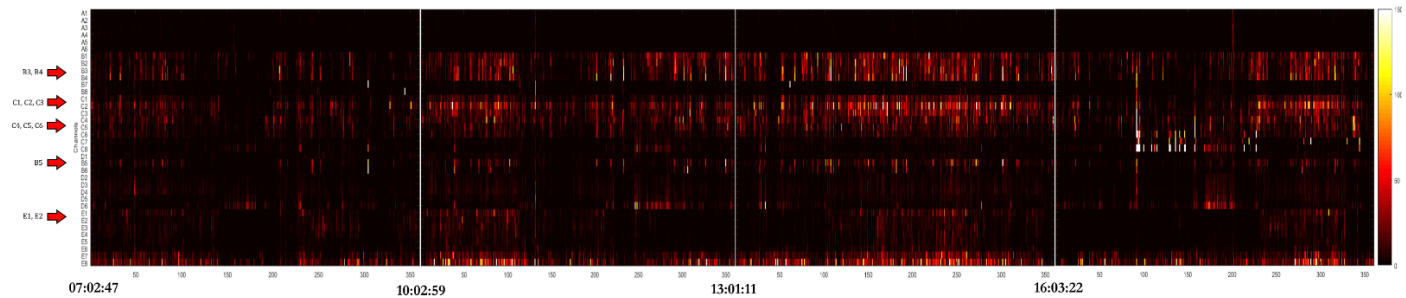


Figure 45: ARRM - Sequential Files for the Beginning of EMU Stay (11/04/2015) and Channels of Interest

Consistently high ARRM values can be seen across time in some channels of interest and also adjacent channels of interest. Strong activity spreading to adjacent channels might be expected, since each group of electrodes is a depth electrode which has multiple channels in the same area of cortex. Channel C2 seems to exhibit the

strongest activity from the C depth electrodes, Channel D6 of the D electrodes, and Channels E7 & E8 of the E electrodes. The ARRm value is elevated for a few epochs in grids B and C near the seizure at 11:58:26. The group of files in the middle of the EMU stay are plotted in Figure 46.

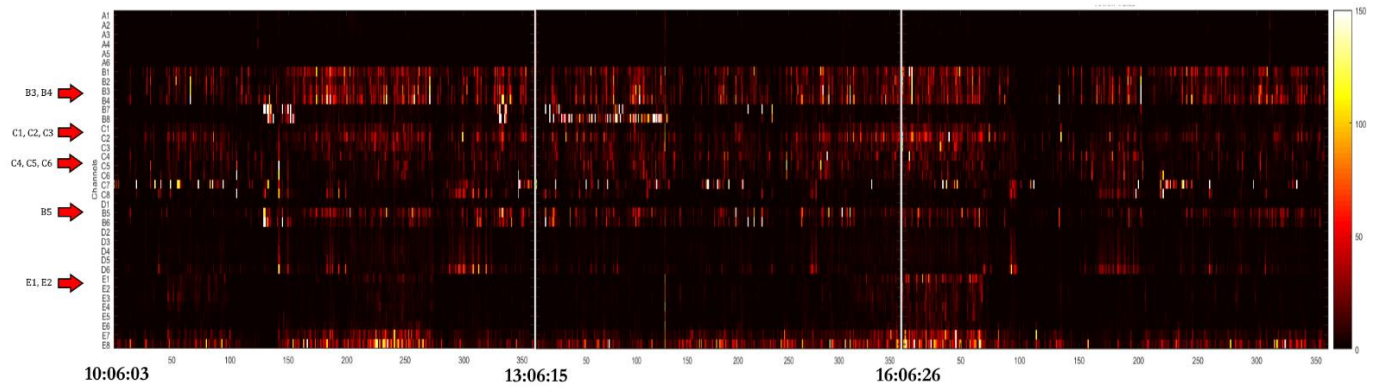


Figure 46: ARRm - Sequential Files for the Middle of EMU Stay (11/06/2015) and Channels of Interest

Consistently high ARRm values can still be observed over time, including C2 most notably. Strong activity appears on B7&B8 at approximately the time of the seizure at 11:14:40. There is also another instance during an inter-ictal period approximately 13:06-14:06 when B7&B8 exhibit high ARRm values. Channels C7, E7, E8, B5, and B6 also exhibit some instances of strong activity. The group of files towards the end of the EMU stay are plotted in Figure 47.

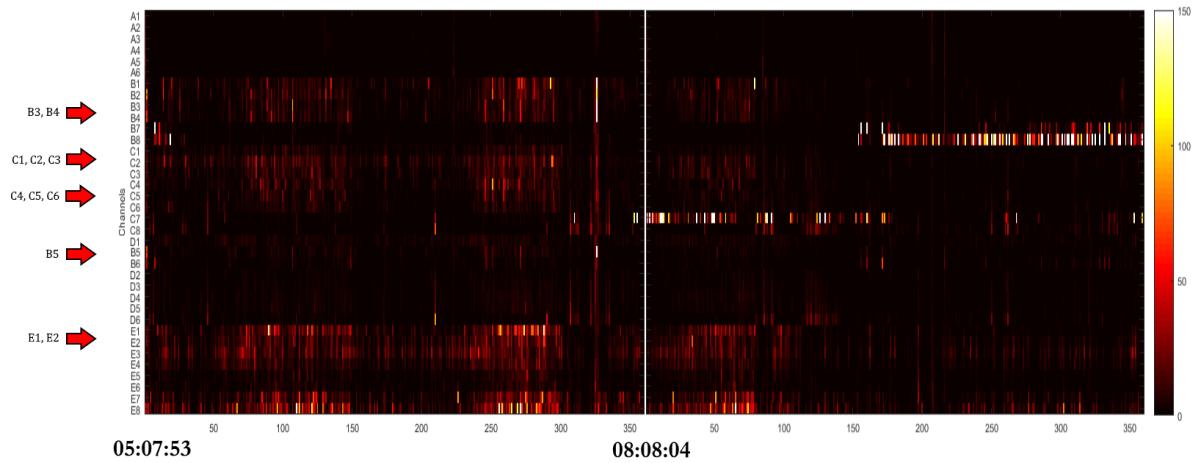


Figure 47: ARRm - Sequential Files for the End of EMU stay (11/08/2015) and Channels of Interest

The same channels still exhibit consistency in the ARRm value over time. High ARRm values across all channels can be observed around the seizure at 7:48:41, and channels C7 & C8 show high values before the seizure. Channels E1 and E8 have a train of notably high ARRm values before this seizure as well. C7 exhibits consistently high ARRm values between approximately 08:08 and 09:30. B8 has consistently high ARRm values between approximately 09:30 and 10:08.

12.2.2 HFO Count

The HFO count was calculated and displayed for all channels over time in a heat plot. The lighter shades of hot colors indicate epochs with high HFO counts, and channels of interest are indicated on the channels axis. The group of files at the beginning of the EMU stay are plotted in Figure 48.

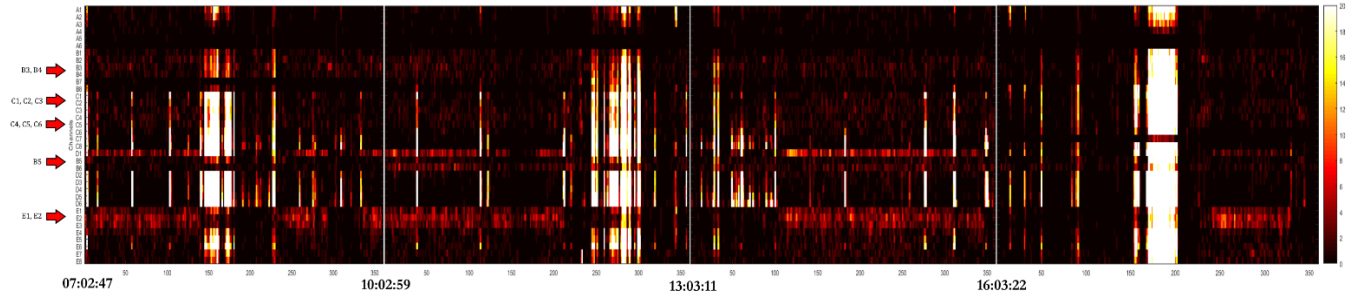


Figure 48: HFO- Sequential Files for the Beginning of EMU Stay (11/04/2015) and Channels of Interest

Consistently high HFO counts are noted on channels D1, B6, and E1-E3. There are multiple instances where the HFO count escalates on all channels at one point in time. It is expected that the fields from strong activity may spread across multiple depth electrodes in the same area of cortex. One of these instances happens at the time of a seizure at approximately 11:58. The other events are unexplained, but may indicate a clusters of synchronized HFO activity. The group of files in the middle of the EMU stay are plotted in Figure 49.

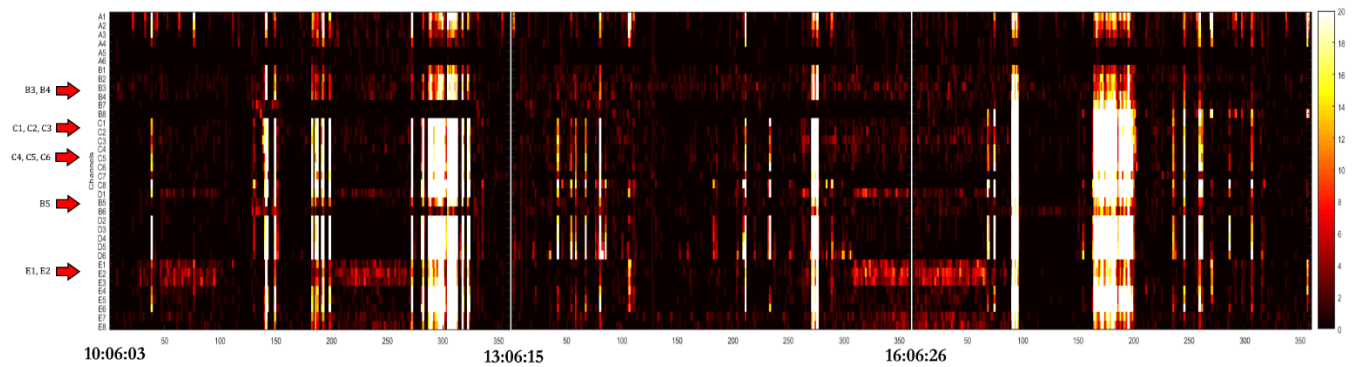


Figure 49: HFO- Sequential Files for the Middle of EMU Stay (11/06/2015) and Channels of Interest

These files do not indicate very consistent patterns of high HFO counts over time.

There are instances of high counts of HFOs on channels E1-3 right before events of

extremely high HFO counts across all channels. Seizures are included in one of these instances at 11:14. The group of files towards the end of the EMU stay are plotted in Figure 50.

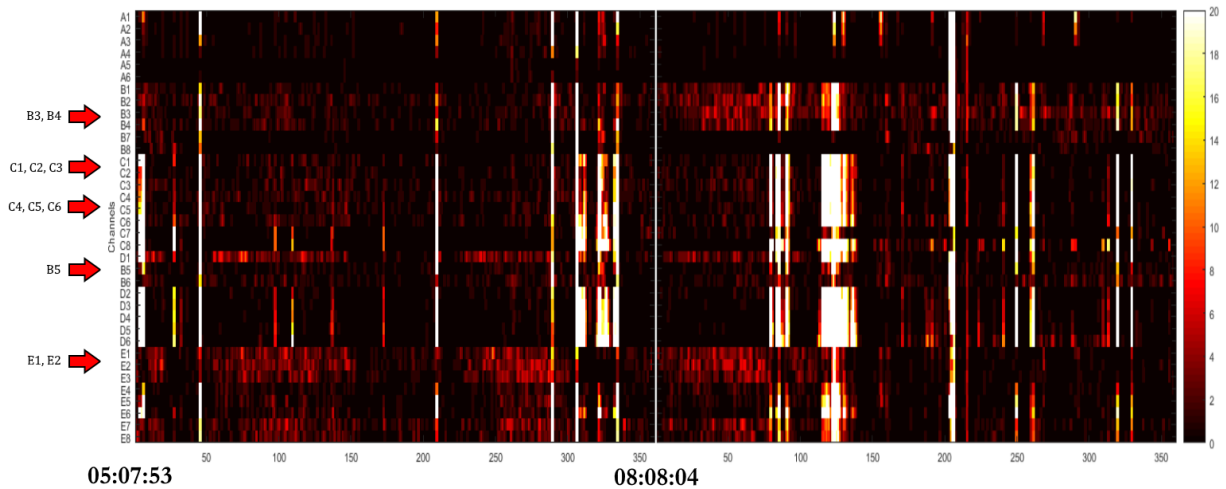


Figure 50: HFO - Sequential Files for the End of EMU stay (11/08/2015) and Channels of Interest

There is some consistently high HFO counts over time on D1, E1&E2. There also are several instances of very high HFO counts across all channels at once. The HFO counts on D1, E1 and E2 tend to elevate before these instances, and for one event, B2 and B3 are also elevated. One of these instances happens around the same time as the seizure activity at approximately 07:48.

12.3. Patient JMM-025

This patient presented with drug-resistant epilepsy, and was evaluated for potential resection surgery. Seizures start with a “yucky taste” and feeling that “sound level is going down”, then an urge to urinate, zoning out and gibberish speech. A previous workup revealed ictal activity in the left anterior-mid temporal area, and interictal activity in the left anterior and mid temporal, occasionally the left posterior temporal. MEG/EEG seizure source localization showed left

frontotemporal spikes, and the EEG dipole solutions were most consistent with the orbitofrontal region, with propagation to the left temporal region. Two populations of MEG activity suggest an irritable region in the left anterior-mid basal temporal region. The patient underwent phase II electrode implantation surgery. Electrode locations are noted in Table 10.

Table 10: Patient JMM-025 Electrode Placement

Electrode Channels	Electrode Type	Location
TD1-16	2x8 Grid	Left lateral temporal surface, mostly covering the middle and inferior temporal gyri. Contact 1 was most posterior & inferior, 8 anterior & inferior, 9 posterior & superior, and 16 anterior & superior.
TB1-6	1x6 Strip	Wrapped around inferior edge of the left temporal pole. Contact 1 being in the inferomesial temporal region and 6 being lateral.
TC1-6	1x6 Strip	Extended from the left inferior temporal gyrus to the interobasal temporal region (perpendicular orientation to the inferior temporal gyrus). Contact 6 was most lateral and contact 1 most medial.
FA1-8	1x8 Strip	Left orbitofrontal region. Contact 1 being most anterior and 8 most posterior.
FB1-6	1x6 Strip	Dorsolateral frontal region. Contact 1 being most anterior and 6 most posterior.
FC1-12	2x6 Grid	Posterior frontal convexity. Contact 1 being most anterior & superior, 6 anterior & inferior, 7 posterior & superior.

The sampling frequency for this patient was 2000Hz. Some seizures were recorded on the files included in this analysis. Seizure times, notes, and files effected can be referenced in Table 11.

Table 11: Patient JMM-025 Seizure Times, Notes and Files

Seizure Time	Seizure Notes	File Affected
14:03:05	Spike at TB1 > TA2-4 followed by 80-90Hz activity.	AA5320XH
00:45:31	Train of semiperiodic spikes at TB1-2 (associated with 85-95Hz activity). Fast activity also seen at TB3 and TA4-5. Irregular slow activity at FA2. Rhythmic activity at FA1-3.	AA5320Y0
04:01:35	Bursts of 80-90Hz activity at TB2-4, TC2. Train of sharp waves at TD7, rhythmic activity at TB3-4, B5, FA2-3, TA2-4, TC6. Spiking at TB1.	AA5320YF

Channels of interest identified by the epileptologist according to patient notes are defined in Table 12.

Table 12: Patient JMM-025 Channels of Identified by Epileptologist

Channel Names	Notes
TB1-3, TA2-4, TC1-2, 4-5, TD3-6	“Interictally, frequent spikes were seen in TB1>TB2-3, TC1>TC2, and TA2>TA3,4... Frequent 80-90Hz activity seen at TB3, TA2-3, occasionally TC1... Less frequent spikes or sharp waves seen at TD3-4>TD5-6, TC4-5... Ictally, a train of spikes were seen at TB1-2, fast activity also seen at TB3 and TA4... ”

Multiple complex partial seizures were recorded for this patient, and one atypical aura. In multiple seizures initial ictal changes were seen in the left inferomesial temporal region, and in one seizure the left orbitofrontal region. IEDs were most commonly seen in the inferomesial temporal region, and less frequently in the lateral temporal region. This study suggest the presence of at least 2 independent SOZs. One likely in the mesial temporal region (TB1-3), and the other could not be accurately lateralized, but changes in the EEG suggest a lateral origin (TD electrodes). The patient notes do not describe the proceedings from this patient’s EMU study.

12.3.1 ARRm Value

The ARRm value was calculated and displayed for all channels over time in a heat plot. The lighter shades of hot colors indicate epochs with high ARRm values, and channels of interest are indicated on the channels axis. The first group of files at the beginning of the EMU stay are plotted in Figure 51.

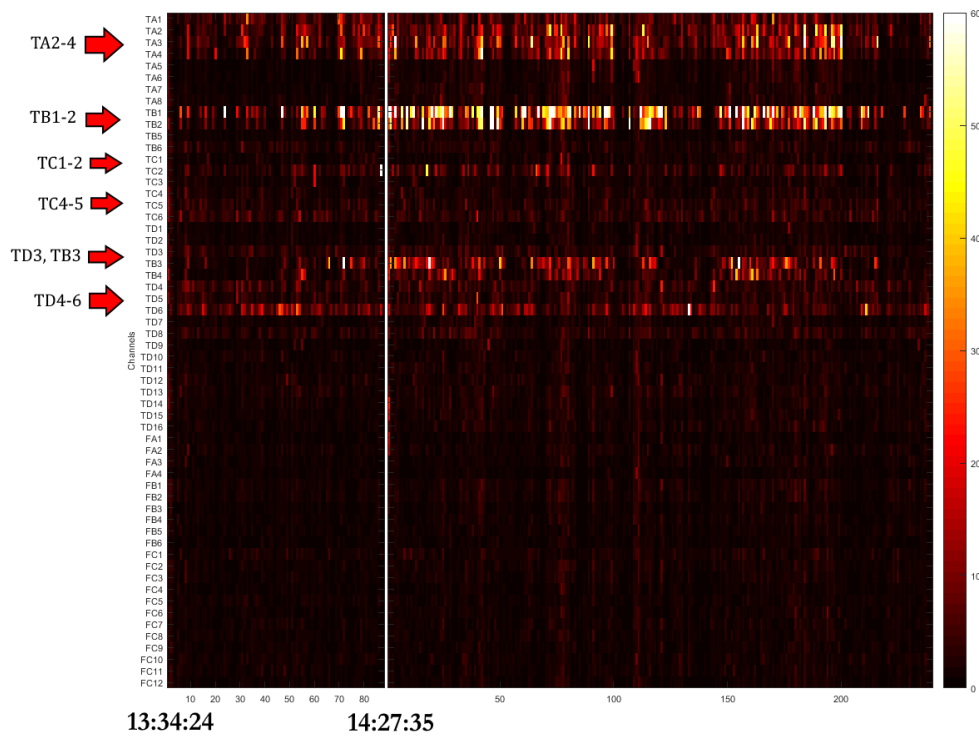


Figure 51: ARRm – First Group of Sequential Files for the Beginning of EMU Stay (06/11/2015) and Channels of Interest

Consistently high ARRm values appear on channels of interest, including TA2-TA4, TB1-2, TC2, TB3-4, and TD6. TB1-2 are showing the highest ARRm values. A seizure was recorded at 14:03, and some epochs with elevated ARRm values can be

seen around that time. The second group of files at the beginning of the EMU stay are plotted in Figure 52.

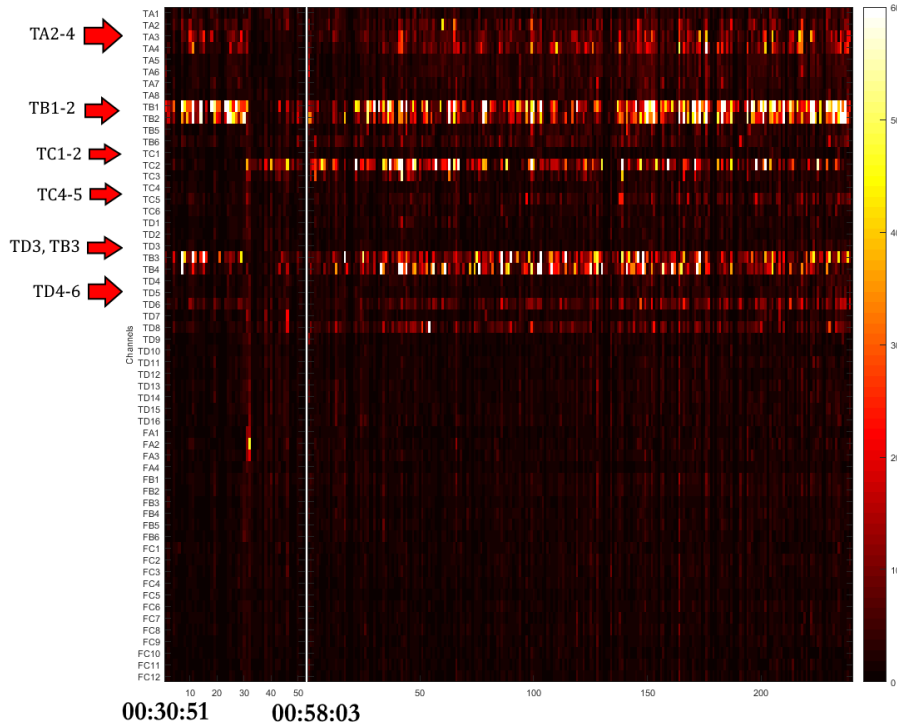


Figure 52: ARRm – Second Group of Sequential Files for the Beginning of EMU Stay (06/13/2015) and Channels of Interest

The same patterns are observed on the same channels for consistently high ARRm values over time, with the addition of TD8. The ARRm value over time for TC2 appears higher in these files than the previous file. A drop in ARRm values is noted at approximately 00:45 when a seizure event occurs, and at this time is when the ARRm value on TC2 increases. After the seizure ends, the other channels are brought back up to the consistent values they were at pre-ictally. The first group of files in the middle of the EMU stay are plotted in Figure 53.

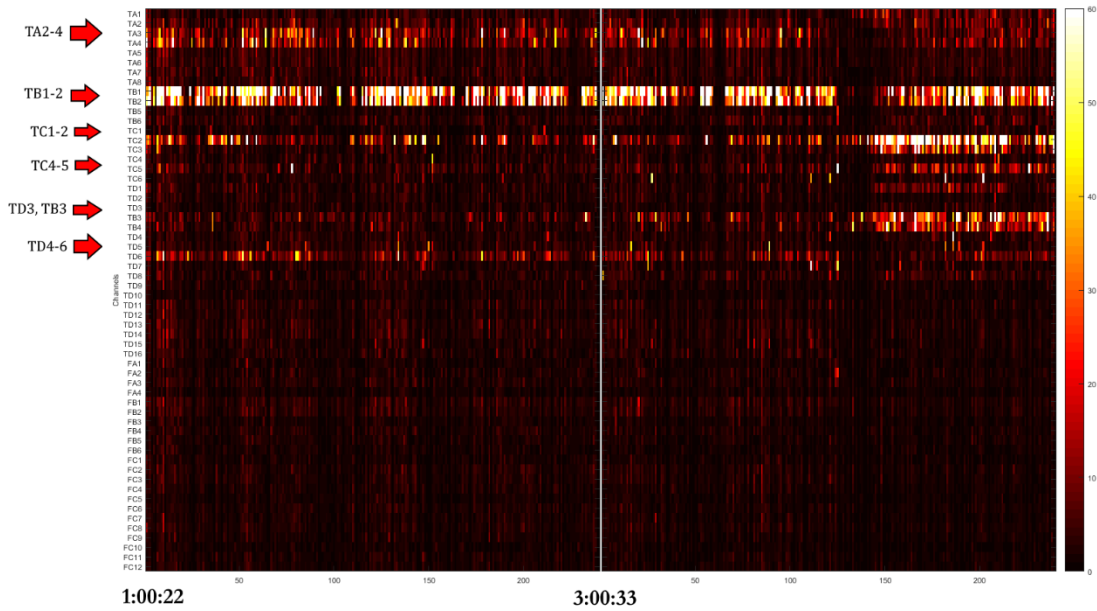


Figure 53: ARRm – First Group of Sequential Files for the Middle of EMU Stay (06/14/2015) and Channels of Interest

The same patterns of consistent ARRm values over time are observed in these files as the previous files. The same pattern when the seizure happens is also observed at approximately 04:01. The ARRm values drop across all channels during the seizure, then after the seizure they return to their pre-ictal values, but channels TC2, TB3, TC5, TD1, and B4 return to higher values than they were pre-ictally. The second group of files in the middle of the EMU stay are plotted in Figure 54.

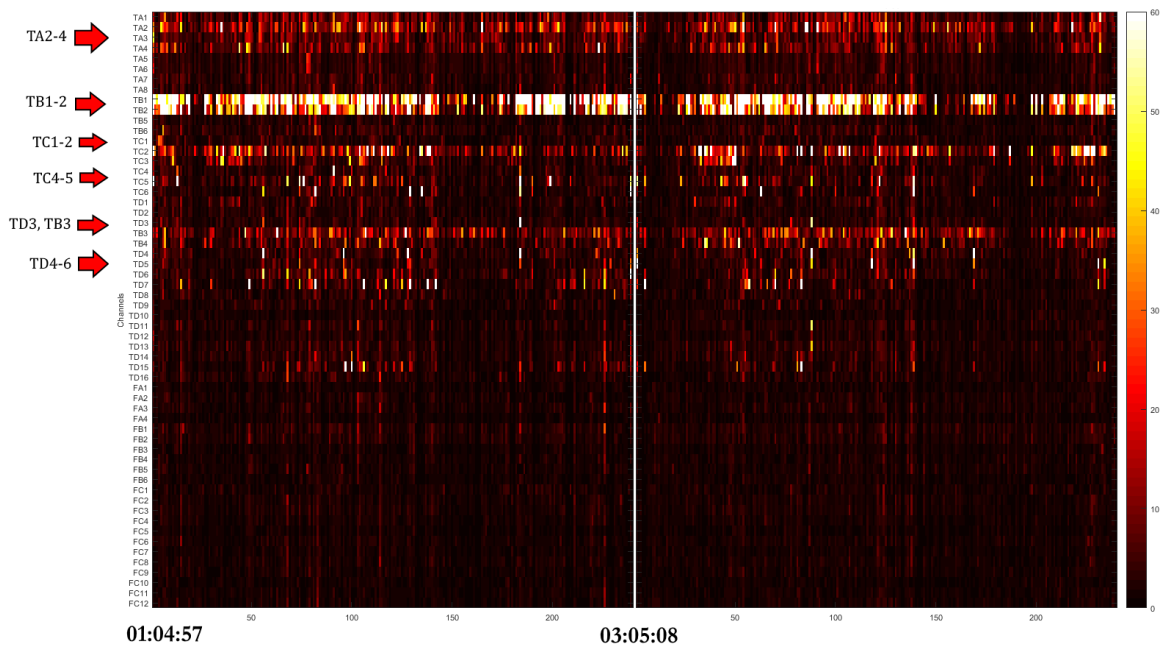


Figure 54: ARRm – Second Group of Sequential Files for the Middle of EMU Stay (06/16/2015) and Channels of Interest

The same patterns can be observed for consistently high ARRm values over time on channels of interest. Channels B1-B3 stand out in the B group and TC2 stands out in the C group. The group of files at the end of the EMU stay are plotted in Figure 55.

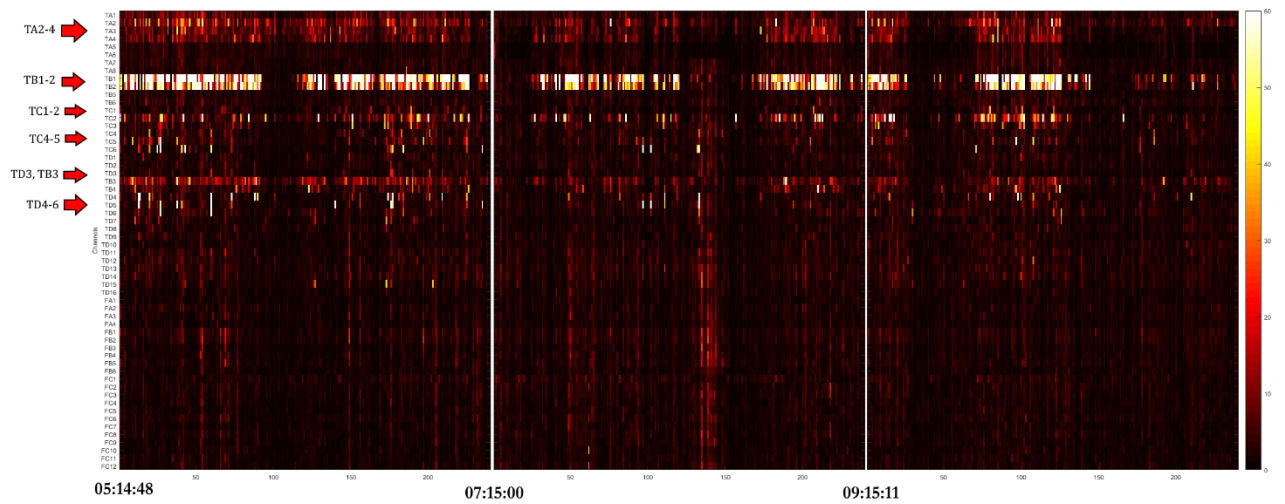


Figure 55: ARRm – Sequential Files for the End of EMU Stay (06/20/2015) and Channels of Interest

Consistently high ARRm values are observed over time for channels of interest, similarly to the previous observation. The ARRm value drops across all channels at multiple different instances in time, but no seizures were recorded in these files according to patient notes.

12.3.2 HFO Count

The HFO count was calculated and displayed for all channels over time in a heat plot. The lighter shades of hot colors indicate epochs with high HFO counts, and channels of interest are indicated on the channels axis. The first group of files at the beginning of the EMU stay are plotted in Figure 56.

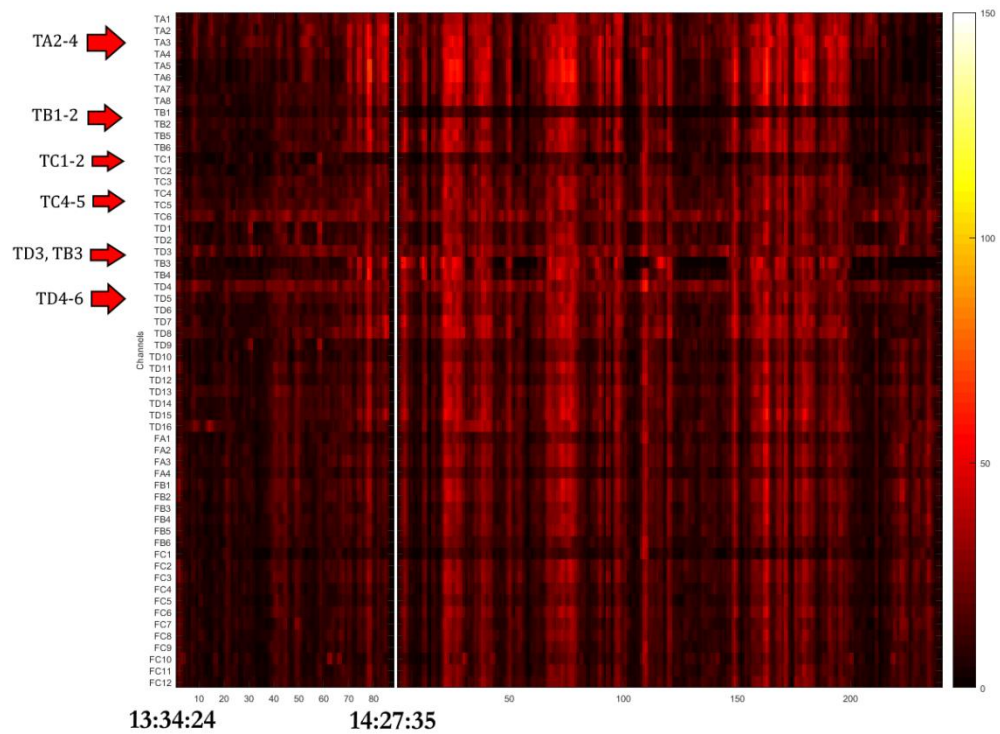
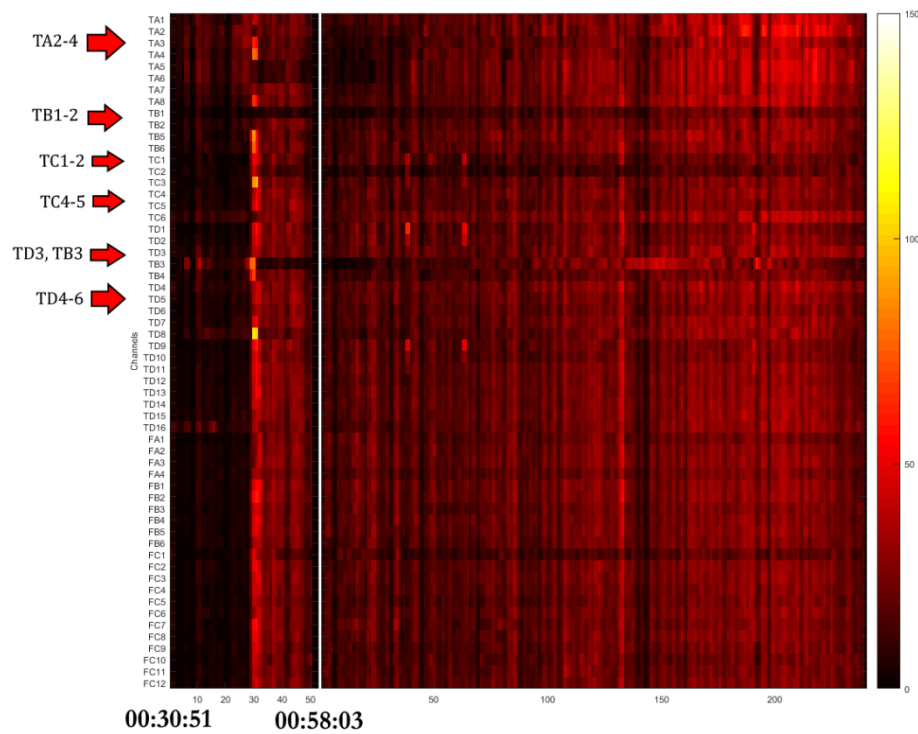


Figure 56: HFO – First Group of Sequential Files for the Beginning of EMU Stay (06/11/2015) and Channels of Interest

Consistently high HFO counts over time can be seen in channels TC6, TD3, and TD4. There is a large number of instances where the HFO counts are high across all channels at one point in time. The second group of files at the beginning of the EMU stay are plotted in Figure 57.



**Figure 57: HFO – Second Group of Sequential Files for the Beginning of EMU Stay (06/13/2015)
and Channels of Interest**

No consistent patterns in HFO count are observed over time for these files. There is a notable increase in HFO count across all channels at approximately 00:45 when a seizure occurs, with the highest HFO counts on channels TD8, TC3, TB3, and TB5-6. The first group of files in the middle of the EMU stay are plotted in Figure 58.

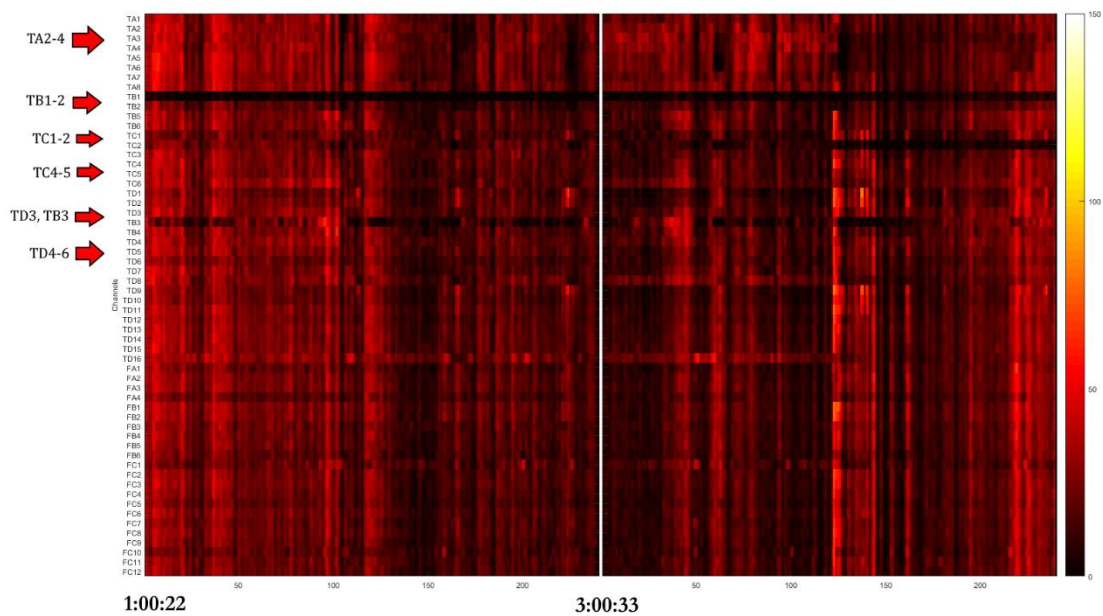


Figure 58: HFO – First Group of Sequential Files for the Middle of EMU Stay (06/14/2015) and Channels of Interest

Channel TB1 and TB3 have a consistently lower HFO count over time than other channels. Some other channels tend to stand out but there are many instances of high counts of HFOs across all channels at multiple points in time. There is a notable increase in HFO counts at approximately the time of seizure activity at 04:01, near the middle of the second file. The second group of files in the middle of the EMU stay are plotted in Figure 59.

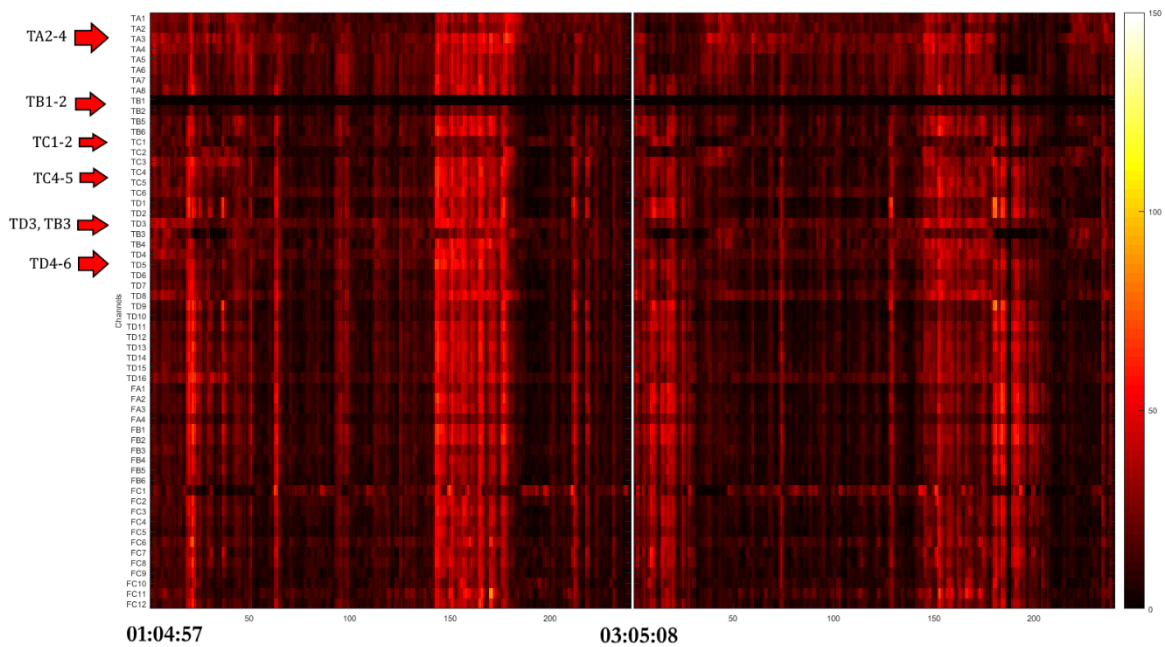


Figure 59: HFO – Second Group of Sequential Files for the Middle of EMU Stay (06/16/2015) and Channels of Interest

Channel FC1 appears to have consistently high HFO counts over time, along with a few other channels of interest that are not as distinguished. There are multiple instances of high HFO counts across all channels at one point in time, but no seizures were recorded in these files. The group of files at the end of the EMU stay are plotted in Figure 60.

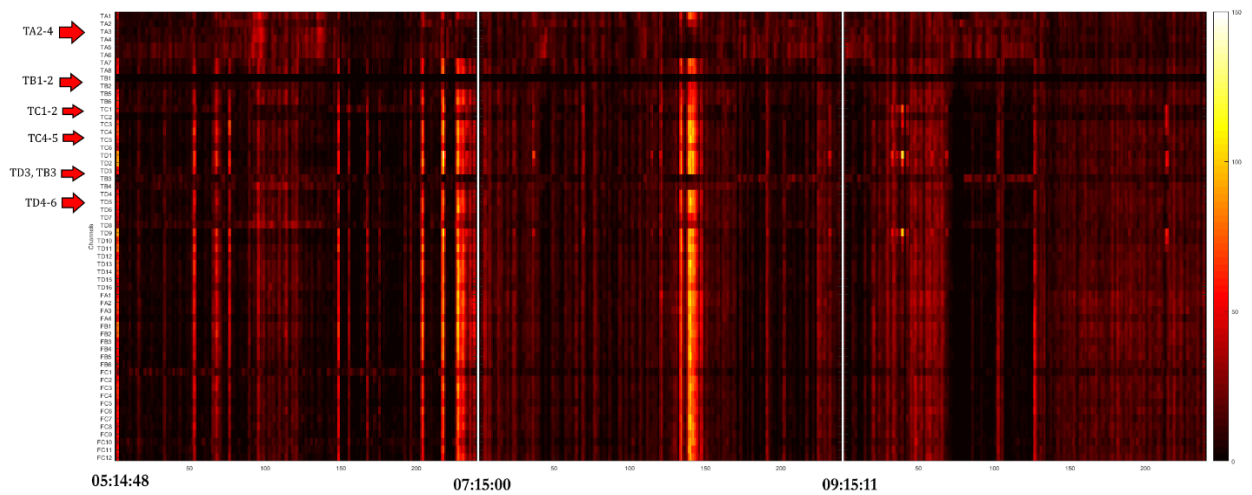


Figure 60: HFO – Sequential Files for the End of EMU Stay (06/20/2015) and Channels of Interest

There seem to be consistent patterns of HFO counts over time on channels FC1, TD8, and TB3-4. Multiple instances of high counts of HFOs across all channels at one point in time are observed through out these files, most notable at approximately 7:00 and 8:20.

13. Median ARRM and Total HFO Count Plots

To examine the consistency of the ARRM value and HFO count over time, the median ARRM value and Total HFO count for each channel was calculated on sequential heat plot datasets.

13.1 Patient WDH-022

The results from the beginning of the EMU stay were analyzed (Figure 61).

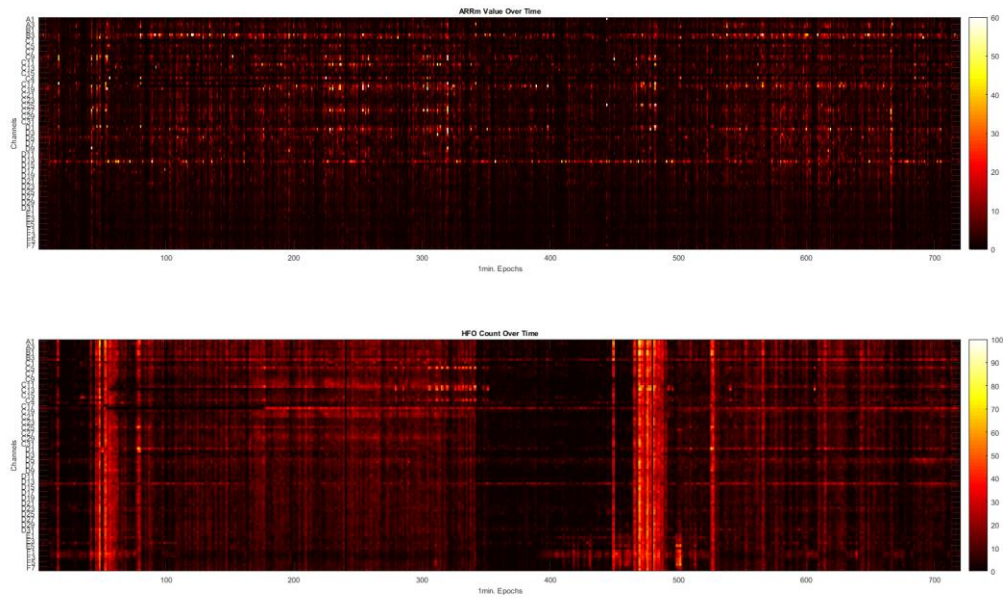


Figure 61: ARRm Value and HFO Count Over Time from the Beginning of EMU Stay – Patient WDH-022

The median ARRm Value and Total HFO Count are plotted in Figure 62. Channels of interest are highlighted by red asterisks.

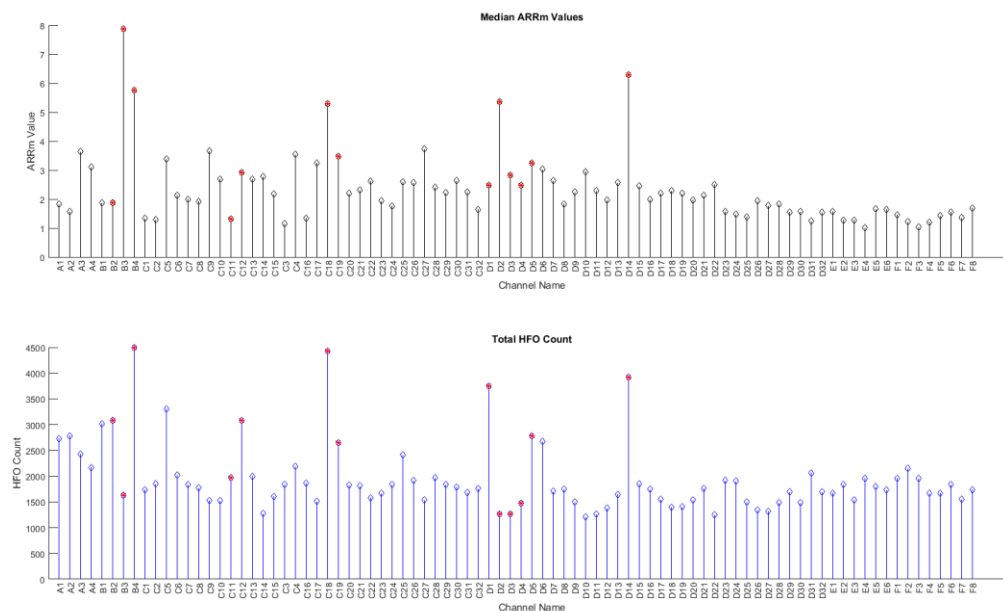


Figure 62: Median ARRM Value and Total HFO Count from the Beginning of EMU Stay – Patient WDH-022

Similarities can be observed between the channels with a high median ARRM value and high Total HFO count. Most channels with high values of either measure are channels of interest. This may indicate a correlation between the ARRM value, HFO Count, and channels of interest. An additional observation is that for some channels, when one has a high median ARRM value, the other has a high HFO count on an adjacent channel, or vice versa. Channels of this example include B3&B4, and D1&D2. Additional plots of this patient’s files can be found in the Appendix.

13.2 Patient DJA-033

The results from the beginning of the EMU stay were analyzed in Figure 63.

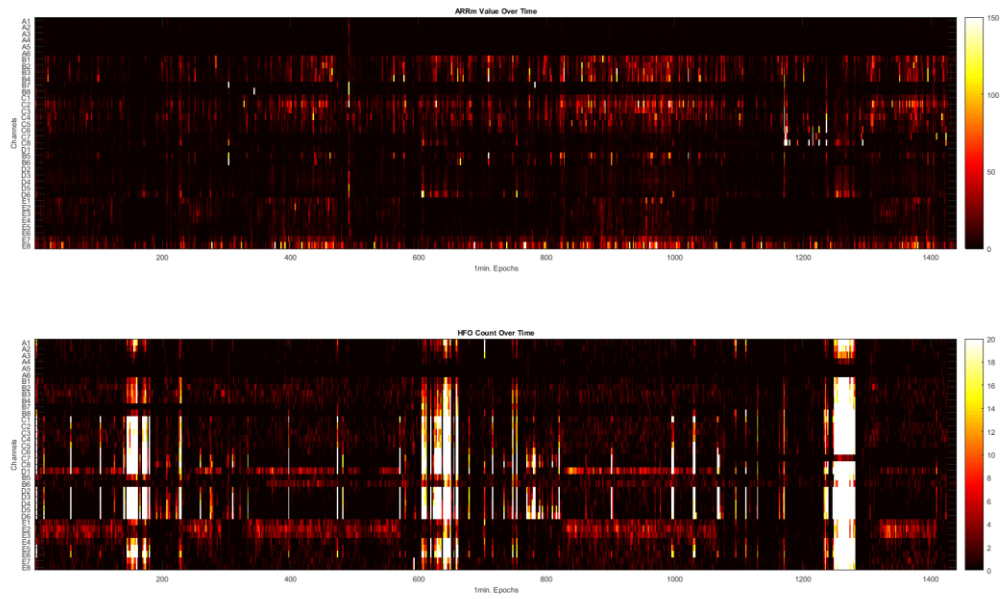


Figure 63: ARRm Value and HFO Count Over Time from the Beginning of EMU Stay – Patient DJA-033

The median ARRm Value and Total HFO Count are plotted in Figure 64. Channels of Interest are highlighted by red asterisks.

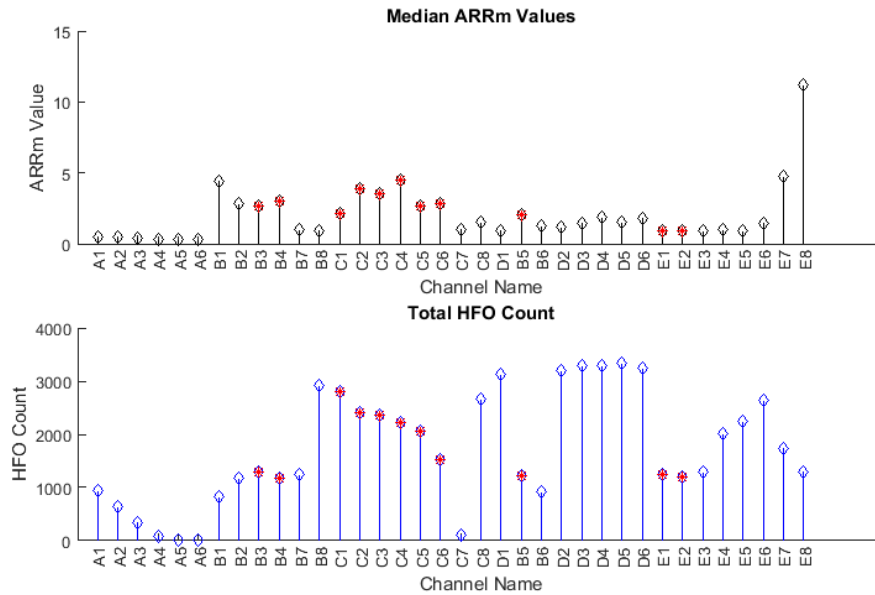


Figure 64: Median ARRM Value and Total HFO Count from the Beginning of EMU Stay – Patient DJA-033

In the median ARRM values plot, channels of interest in the B and C groups of electrodes have elevated values compared to the rest of the channels. Channels E7 and E8 stand out with the highest counts, but they were not identified as channels of interest by the epileptologist. It would be interesting to investigate the signal activity on these channels in the ARRM/HFO analysis tool. In the total HFO count plot, it is difficult to determine which channels of interest stand out based on inter-ictal activity because the multiple instances of high HFO counts across all channels has elevated the total HFO counts for all channels. Additional plots of this patient's files can be found in the Appendix.

13.3 Patient JMM-025

The results from the first group of files at the beginning of the EMU stay were analyzed in Figure 65.

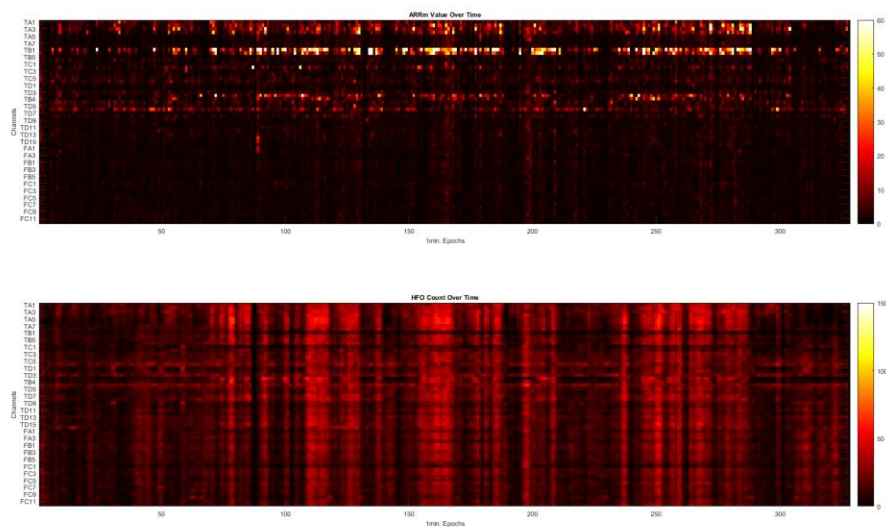


Figure 65: ARRm Value and HFO Count Over Time from First Group of Files at the Beginning of EMU Stay – Patient JMM-025

The median ARRm Value and Total HFO Count are plotted in Figure 66. Channels of Interest are highlighted by red asterisks.

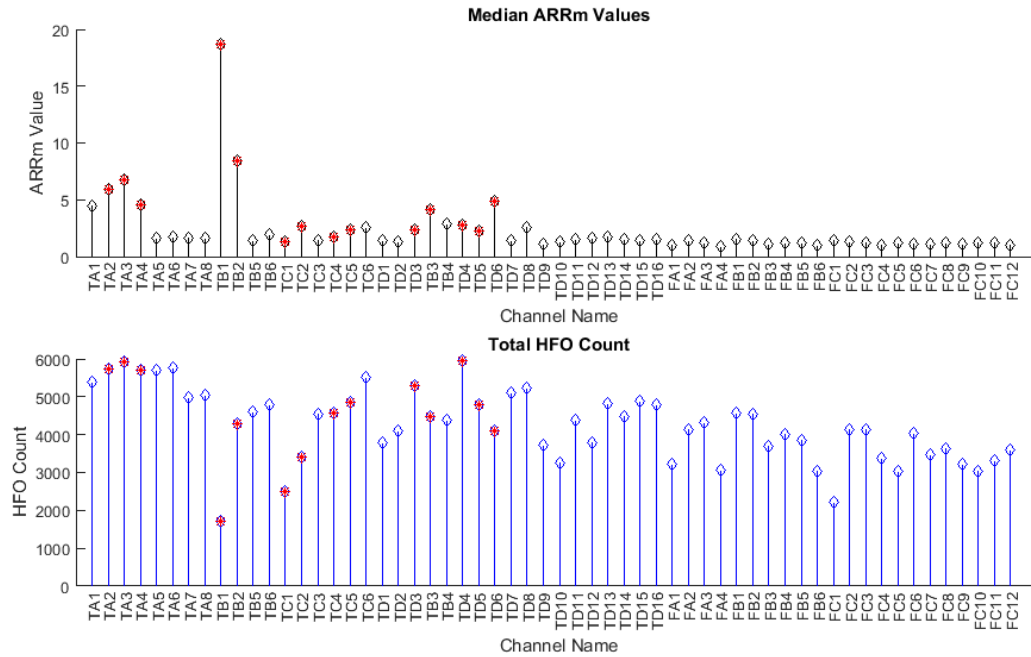


Figure 66: Median ARRM Value and Total HFO Count from the First Group of Files at the Beginning of EMU Stay – Patient JMM-025

In the median ARRM Values plot, channels TB1 and TB2 stand out the most, and are indicated as channels of interest. In the patient notes it was indicated that these two channels had frequent occurrences of inter-ictal spikes. Elevated values in the TA electrode group are also channels of interest. TB3 and TD6 also stand out from other electrodes and are indicated as channels of interest. The total HFO count plot does not have any channels that stand out from inter-ictal activity because the multiple instances of high HFO counts across all channels has elevated the total HFO counts for all channels. Additional plots of this patient's files can be found in the Appendix.

14. Correlation Between ARRM Value and HFO Count

To statistically test if the ARRM value is related to the HFO Count, a linear and rank correlation was found between corresponding channels of the datasets for ARRM value over

time and HFO count over time. Correlations were calculated between corresponding channels for the two methods of analysis, and the results were expected to indicate whether the ARR_m value over time is related to the HFO rate over time.

14.1 HFO Detection EOI Threshold

As was observed in the Results, the plot of HFO counts over time yielded frequent instances of extremely high HFO rates across all channels, which muddled the interictal activity. Therefore, correlations cannot be drawn between the HFO count over time and ARR_m value over time because the ARR_m value over time showed strong consistent activity for some channels of interest, and minimal instances of high values across all channels at once. Some of these instances of high rates at a certain point in time were due to seizure activity, but others seemed to be due to other signal activity.

It was discovered that an optimal threshold had been chosen for events of interest detection upon close examination of comments in the original code of the HFO detection method. This was done as opposed to calculating the threshold with the standard deviation and mean of the envelope of the signal, as defined in the previous study [23] and as had been implemented in RIPPLELAB. However, it was found in the previous study that increasing the threshold by scaled amounts does not alter the rate of consistent inter-ictal activity over time, and it reduces the amount of HFOs detected across all channels at once at many of the epochs. Moving forward, optimal thresholds (Table 13) were chosen for each patient, and 1-2 files were reanalyzed with the HFO detection method and the EOI threshold set to the optimal chosen threshold.

Table 13: Chosen Optimal HFO Detection EOI Thresholds for Each Patient

Patient	Optimal EOI Threshold Value
WDH-022	18.0
DJA-033	4.0
JMM-025	30.0

For patient WDH-022, the optimal threshold was chosen to be 18.0 by the previous study of the modified Burnos HFO detection method. The plots for two files run with this threshold are shown in Figure 67 and Figure 68.

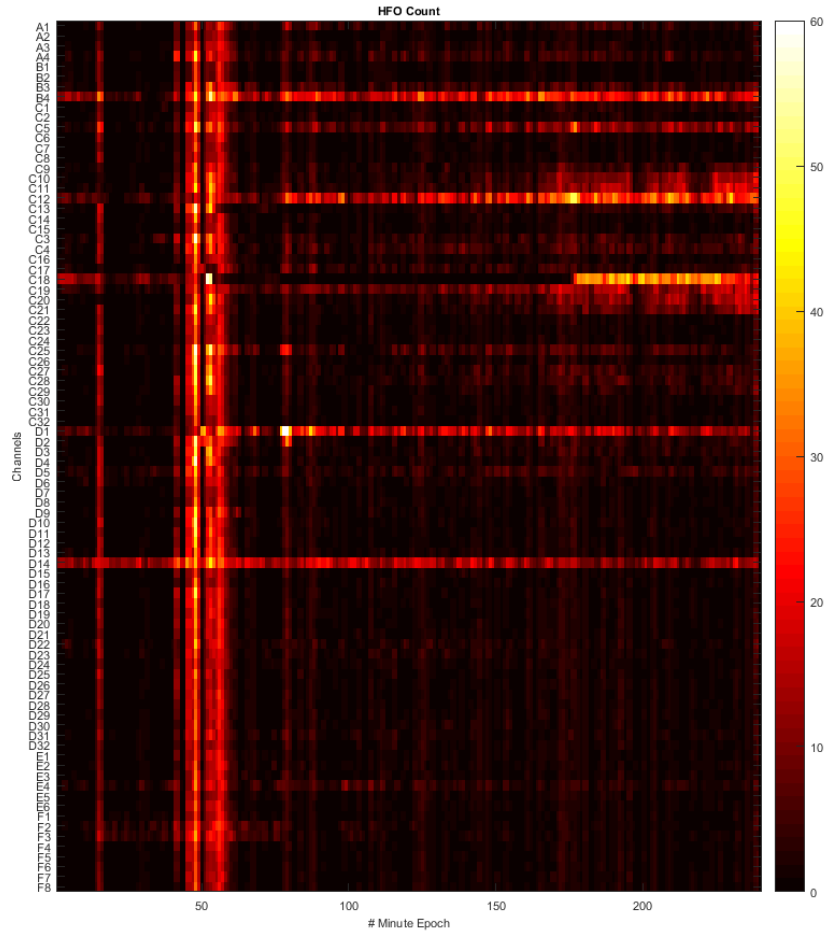


Figure 67: File BA2681LD with Optimal EOI Threshold

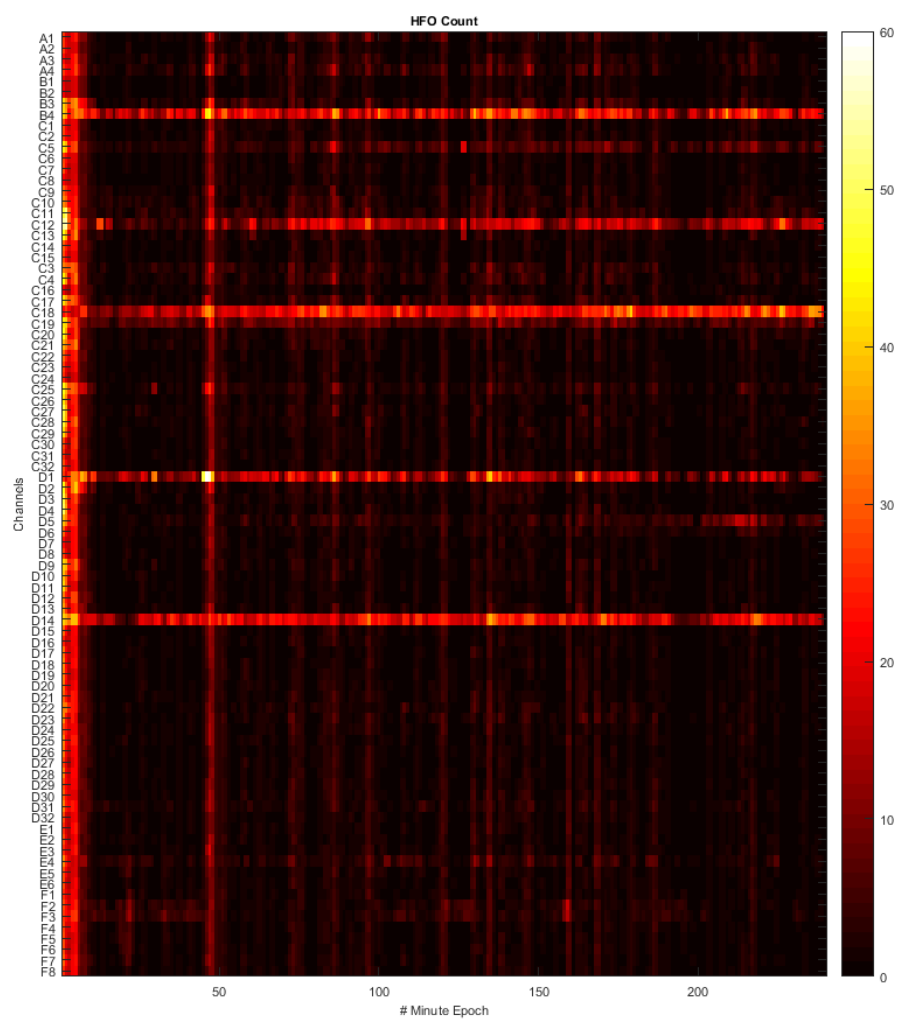


Figure 68: File BA2681LF with Optimal EOI Threshold

For patient DJA-033, three new thresholds were tested Figure 69.

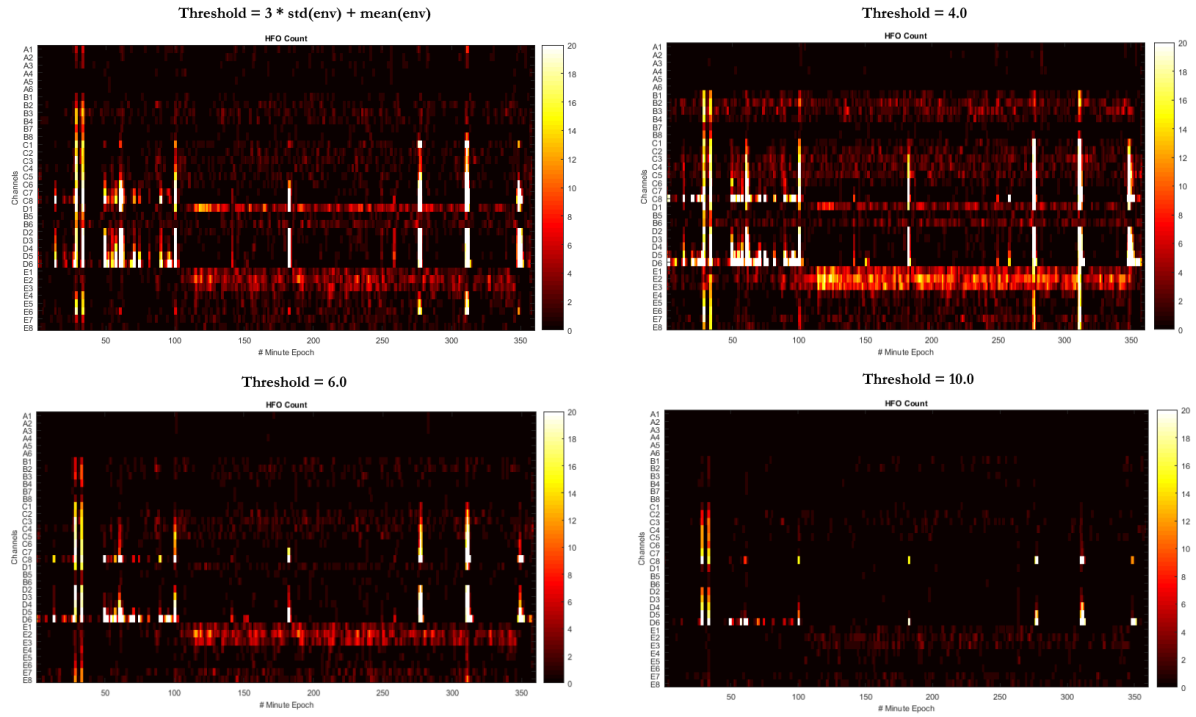


Figure 69: Patient DJA-033 Optimal Threshold Tests

The chosen threshold was 4.0 because although there still were some instances of high activity over time, the most consistent interictal activity was highlighted. The plot for the file run with this threshold is shown in Figure 70.

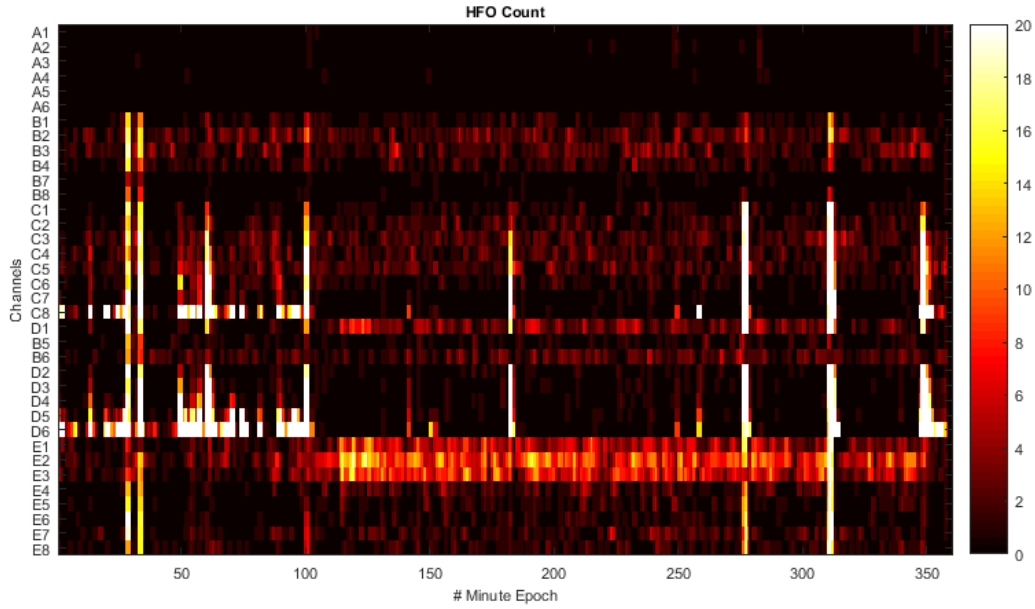


Figure 70: File AA536176 with Optimal EOI Threshold

One new threshold of 30.0 was tested for patient JMM-025, and it yielded strong consistent inter-ictal activity over time, therefore this threshold was chosen. The original plots for the two files chosen for analysis with the optimal threshold are shown in Figure 71.

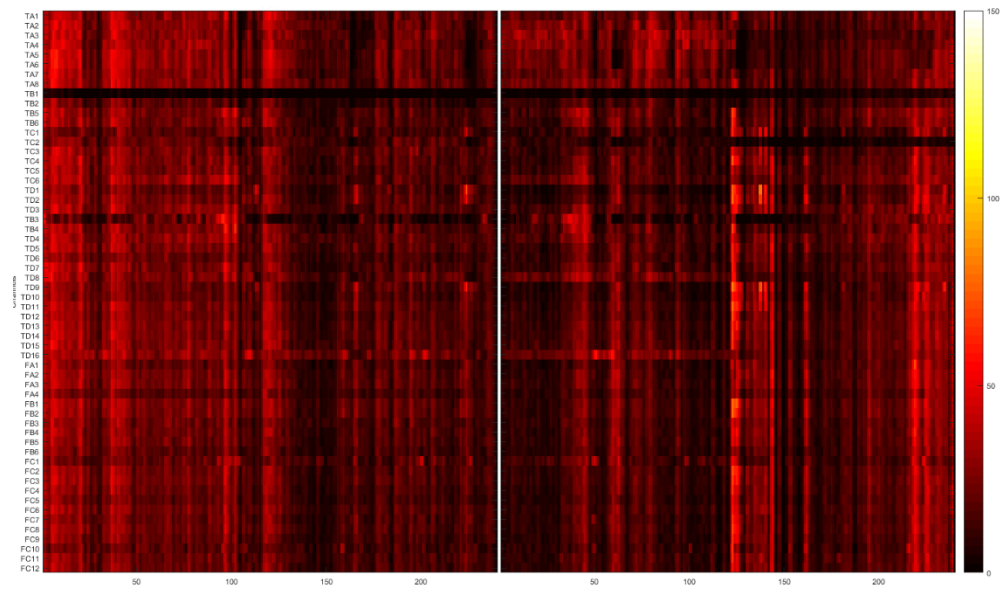


Figure 71: File AA5320YE and AA5320YF With Original Thresholding

The plots for two files run with this threshold are shown in Figure 72 and Figure 73.

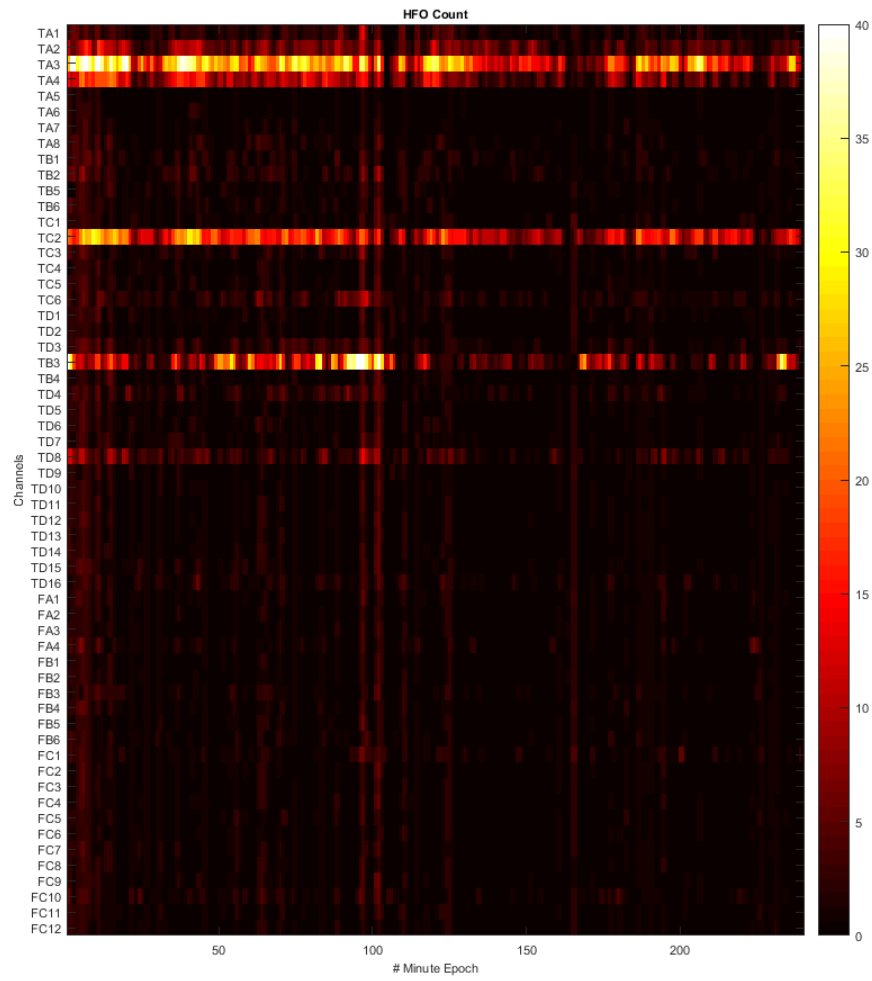


Figure 72: File AA5320YE With Optimal EOI Threshold

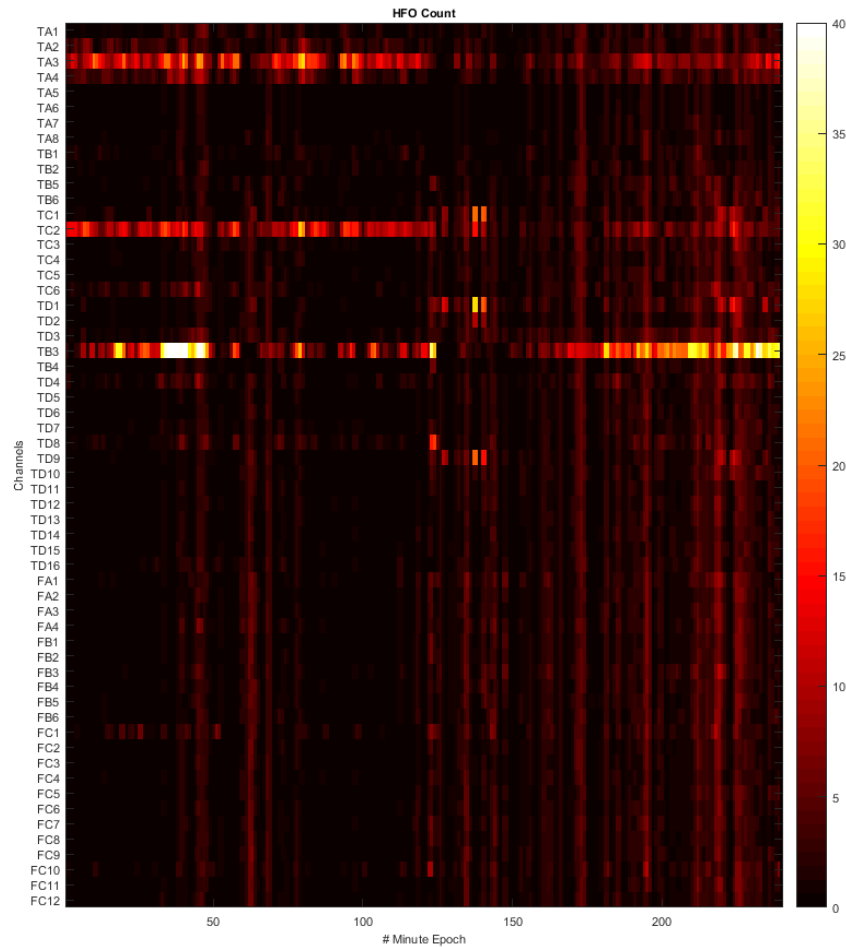


Figure 73: File AA5320YF With Optimal EOI Threshold

14.2 Statistical Correlations

The Pearson linear correlation and Spearman rank correlation between the ARRm value and HFO count over time was calculated. Only HFO count files that were analyzed with an optimized EOI threshold were included in statistical analysis. Correlations between channels were considered significantly different from zero if the p-value was < 0.01 , and the null hypothesis was rejected. The null hypothesis of these correlation tests is that respective channels between the ARRm value and HFO count will not be correlated (correlation coefficient = 0).

14.2.1 Patient WDH-022

The first set of data that was analyzed for correlation is shown in Figure 74.

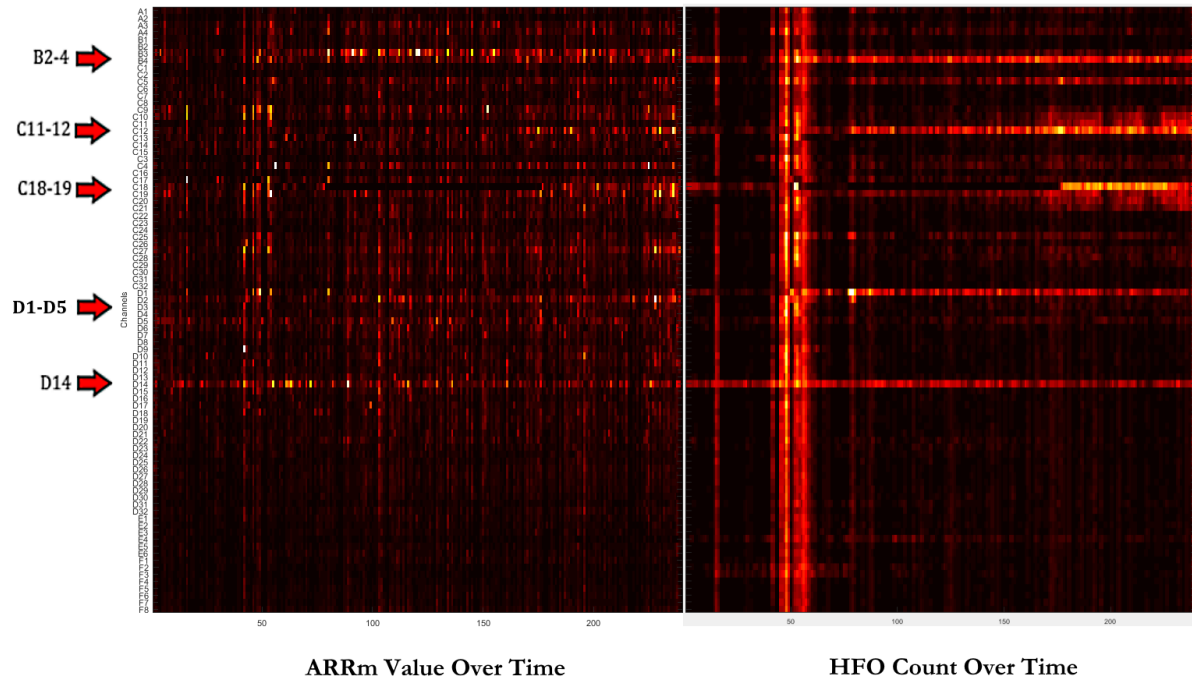


Figure 74: Correlation Test – Patient WDH-022, File BA2681LD

The results of the Pearson and Spearman correlations are shown in Figure 75.

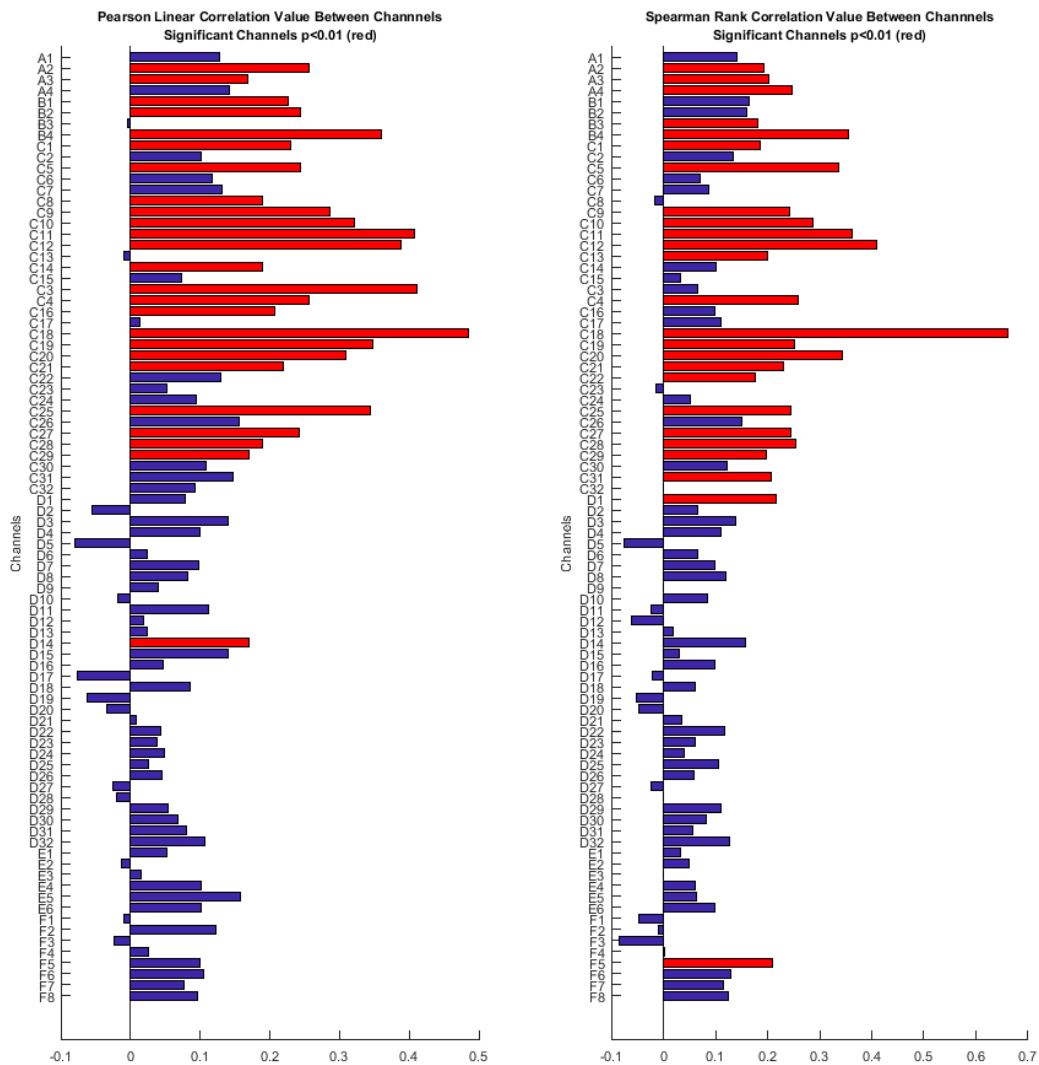


Figure 75: Spearman and Pearson Correlation Results - Patient WDH-022, File BA2681LD

In reference to the plots above, channel correlations highlighted in red indicate significant correlations. According to these results, the null hypothesis is rejected for channels with significant correlations. This indicates that there are some significant linear and nonparametric correlations between channels in this file. Channels of interest with significant Pearson correlations include: B2, B4, C11, C12, C18, C19,

and D14. Channels of interest with significant Spearman correlations include: B3, B4, C11, C12, C18, and C19.

The second set of data that was analyzed for correlation is shown in Figure 76.

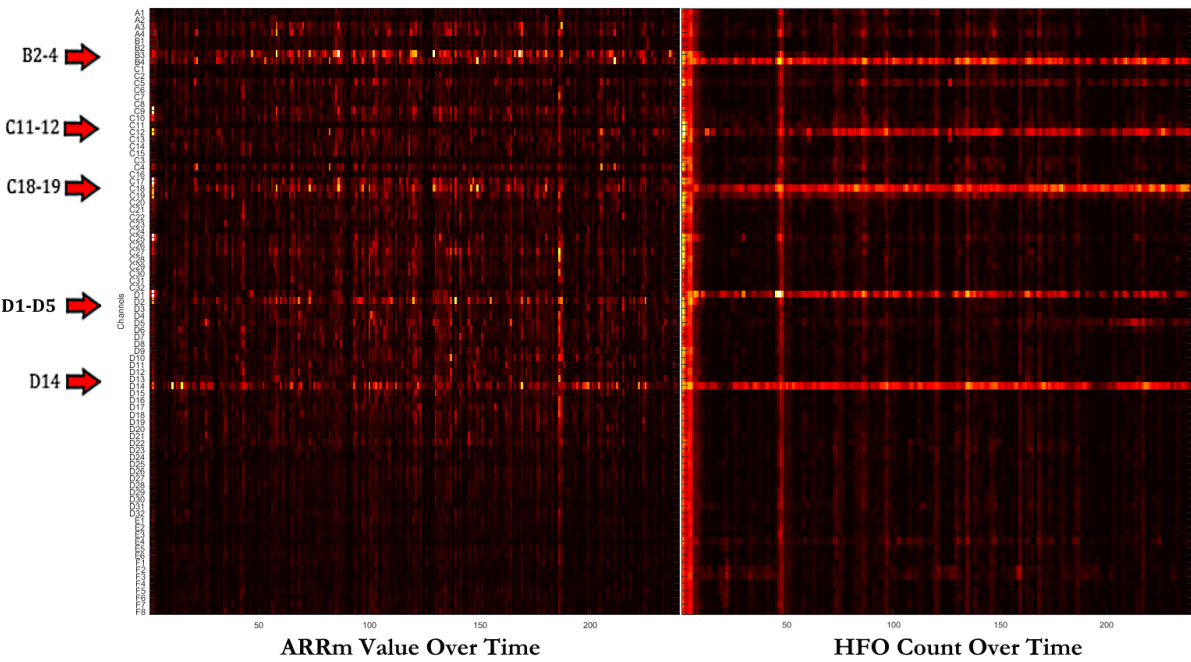


Figure 76: Correlation Test – Patient WDH-022, File BA2681LF

The results of the Pearson and Spearman correlations are shown in Figure 77.

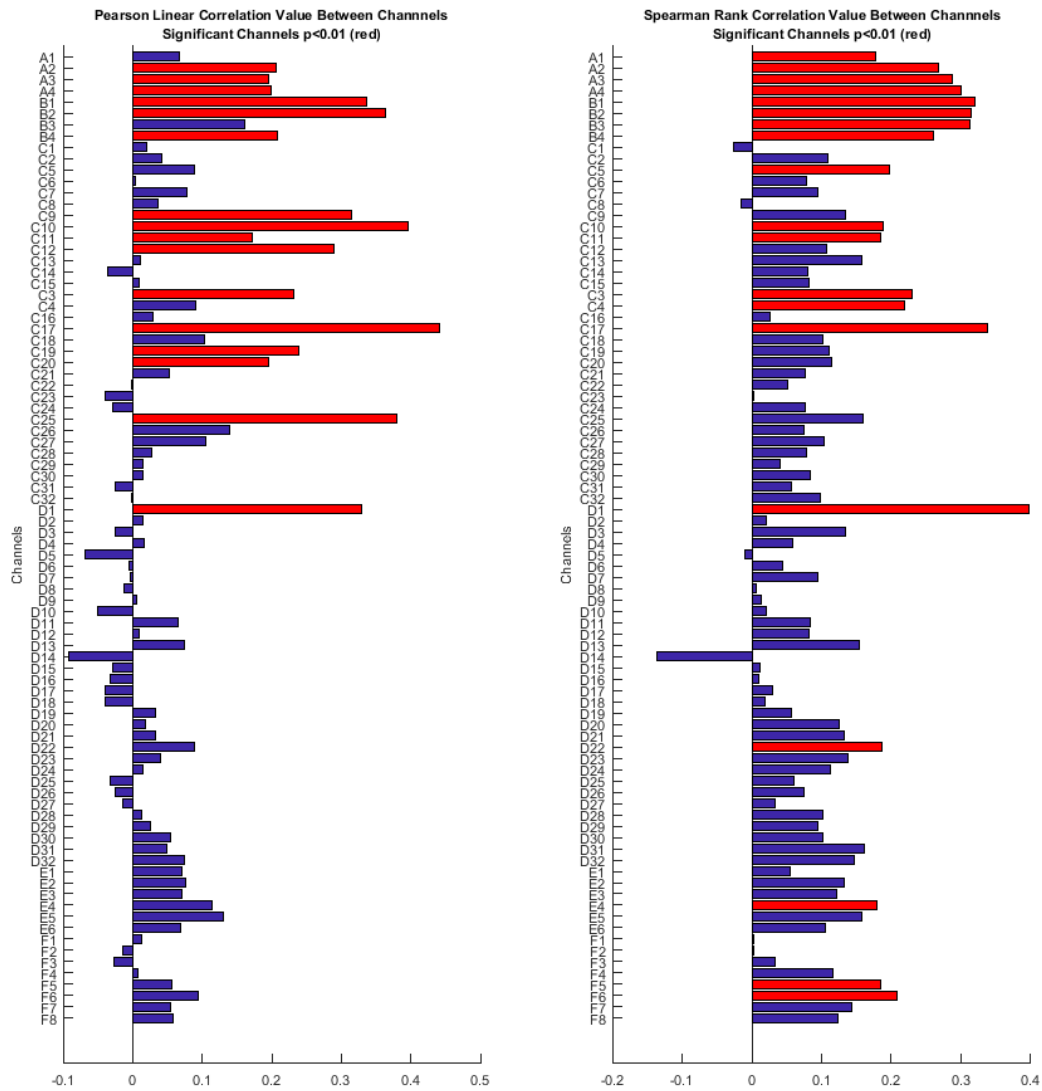


Figure 77: Spearman and Pearson Correlation Results - Patient WDH-022, File BA2681LF

According to these results, the null hypothesis is rejected for channels with significant correlations. This indicates that there are some significant linear and nonparametric correlations between channels in this file as well. Channels of interest with significant Pearson correlations include: B2, B4, C11, C12, C19, and D1. Channels of interest with significant Spearman correlations include: B2, B3, B4, C11, and D1. It is interesting to note in this file that channel D14 was strongly negatively

correlated in both correlation tests. This may indicate an inverse relationship between the ARRm value and HFO count on that channel for this file.

14.2.2 Patient DJA-033

The set of data that was analyzed for correlation is shown in Figure 78.

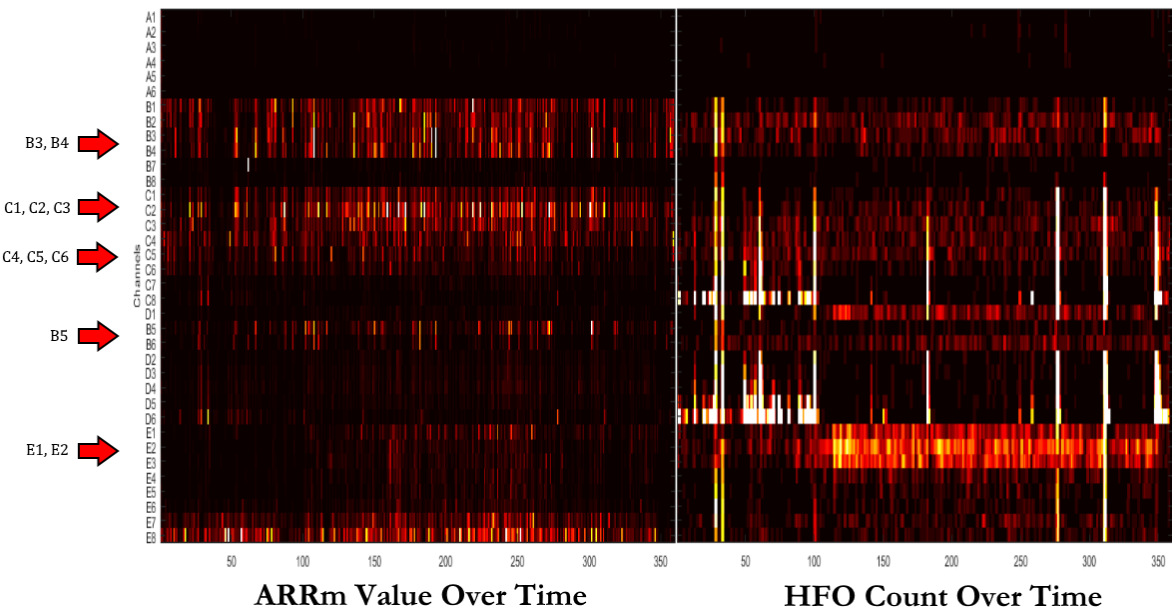


Figure 78: Correlation Test – Patient DJA-033, File AA536176

The results of the Pearson and Spearman correlations are shown in Figure 79.

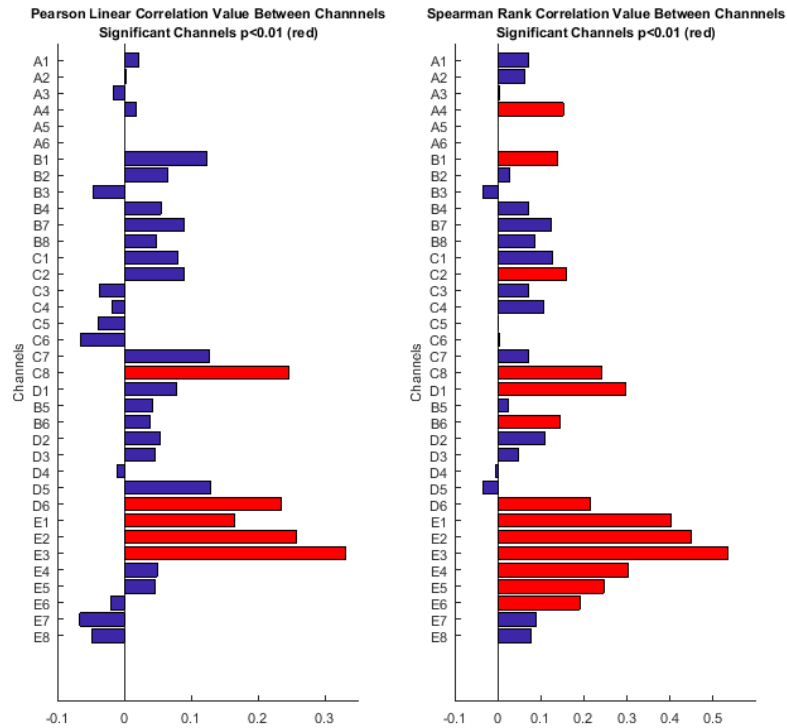


Figure 79: Spearman and Pearson Correlation Results - Patient DJA-033, File AA536176

According to these results, the null hypothesis is rejected for channels with significant correlations. This indicates that there are some significant linear and nonparametric correlations between channels in this file. Channels of interest with significant Pearson correlations include: E1 & E2. Channels of interest with significant Spearman correlations include: C2, E1 and E2. Not many significant correlations between channels of interest were made for this file, which may be due to the peaks in HFO count across all channels at a few instances in time.

14.2.3 Patient JMM-025

The first set of data that was analyzed for correlation is shown in Figure 80.

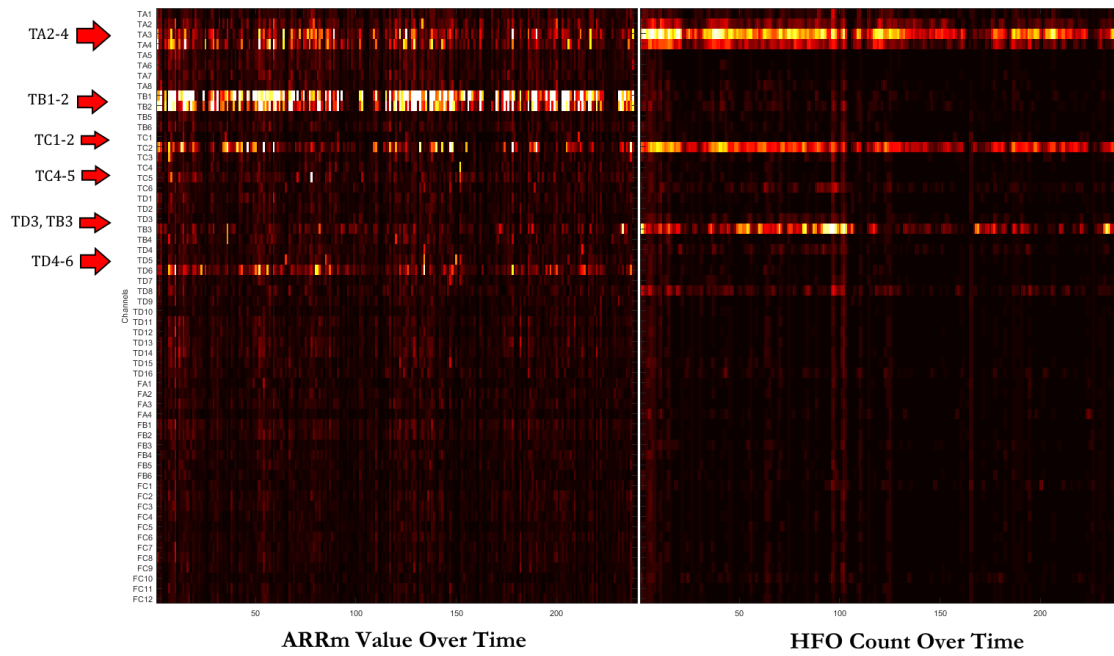


Figure 80: Correlation Test – Patient JMM-025, File AA5320YE

The results of the Pearson and Spearman correlations are shown in Figure 81.

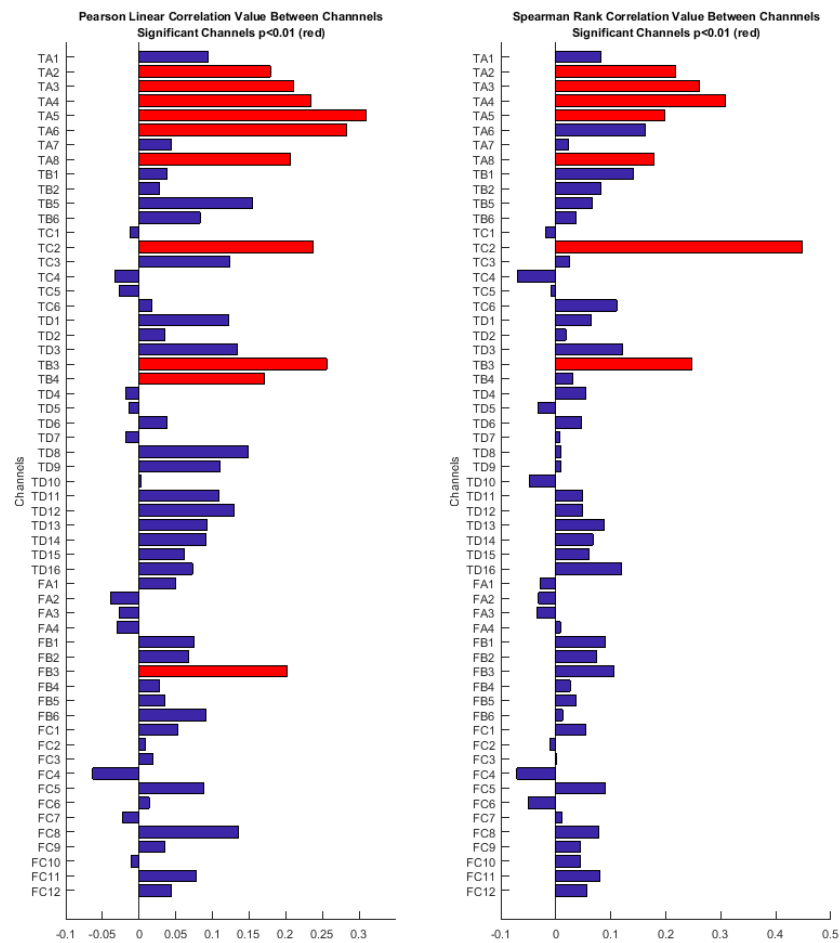


Figure 81: Spearman and Pearson Correlation Results - Patient JMM-025, File AA5320YE

According to these results, the null hypothesis is rejected for channels with significant correlations. This indicates that there are some significant linear and nonparametric correlations between channels in this file. Channels of interest with significant Pearson correlations include: TA2, TA3, TA4, TC2 and TB3. Channels of interest with significant Spearman correlations include: TA2, TA3, TA4, TC2 and TB3. The second set of data that was analyzed for correlation is shown in Figure 82.

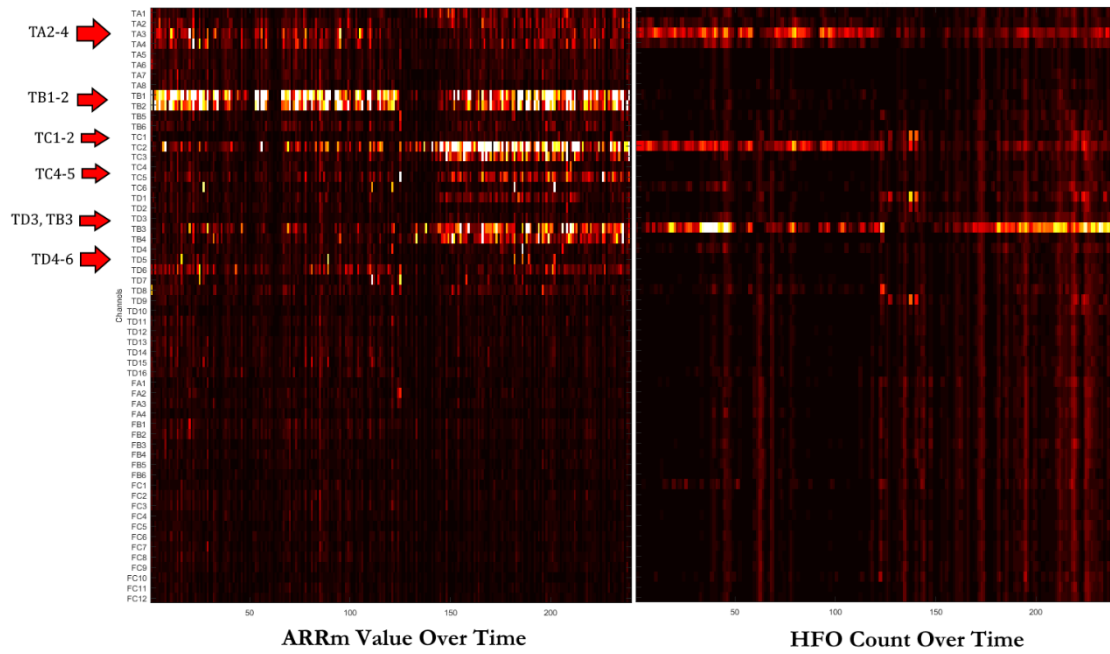


Figure 82: Correlation Test – Patient JMM-025, File AA5320YF

The results of the Pearson and Spearman correlations are shown in Figure 83.

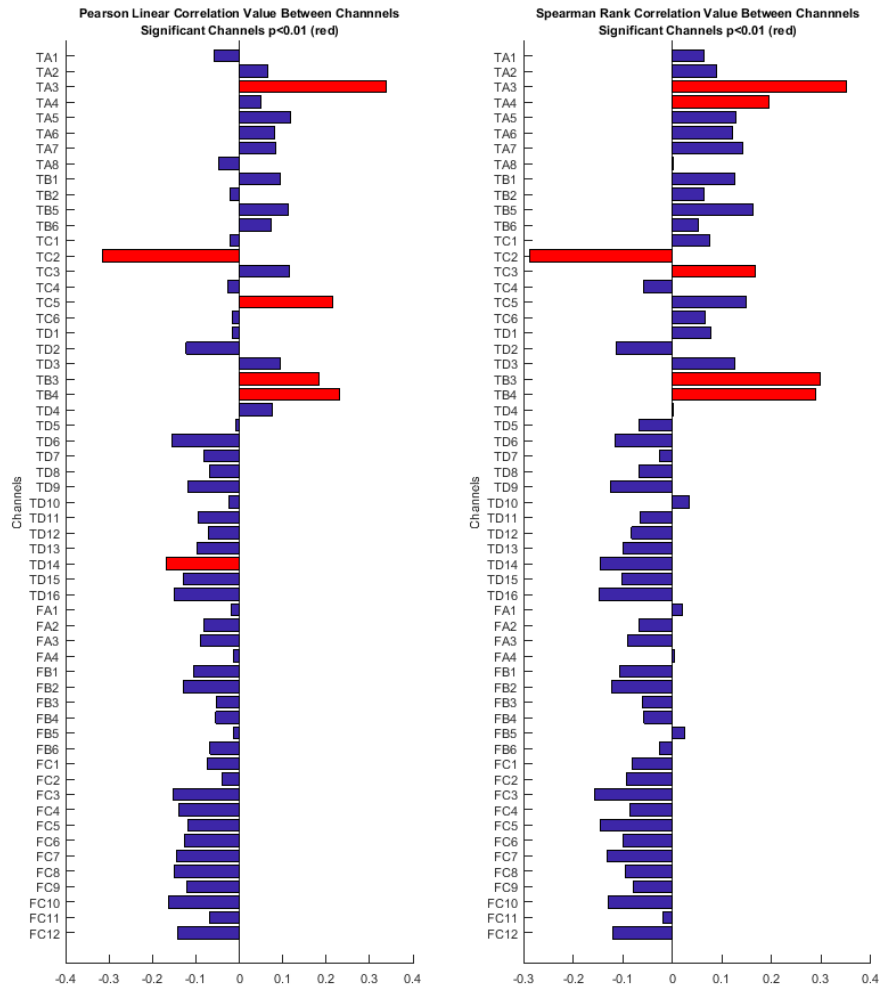


Figure 83: Spearman and Pearson Correlation Results - Patient JMM-025, File AA5320YF

According to these results, the null hypothesis is rejected for channels with significant correlations. This indicates that there are some significant linear and nonparametric correlations between channels in this file. Channels of interest with significant Pearson correlations include: TA3, TC2, TC5 and TB3. Channels of interest with significant Spearman correlations include: TA3, TA4, TC2 and TB3. It should be noted that the correlation between channel C2 is significantly negatively

correlated, which is apparent from the heat plot (Figure 82). The HFO count is high when the ARRm value is low before the seizure, and then after the seizure, the ARRm value is high and the HFO count decreases. The claim in literature was that high values of the ARRm value is most specific fast ripple activity, so perhaps there was a low rate of fast ripples present after the seizure (i.e. the HFO count is lower and ARRm value is higher). Further investigation with the HFO/ARRm analysis tool is warranted in order to determine if the two methods are detecting different signal characteristics.

The relationship between the ARRm value and HFO count with respect to the significantly positive correlation in channel B3, and the significantly negative correlation in channel C2 was investigated using a combination of the ARRm residuals plotting feature in the HFO/ARRm analysis tool and examining the raw iEEG. In order to determine the epileptogenic events or combination of events contributing to either correlation between the ARRm value and HFO count, the behavior of the ARRm residuals was examined for multiple different types of events. Epileptogenic events that were examined include: HFOs (Ripples), Fast Ripple, Spike and HFO, Sharp Spike and HFO, Epileptic discharge and HFO. A segment of iEEG with two HFOs is shown in Figure 84.

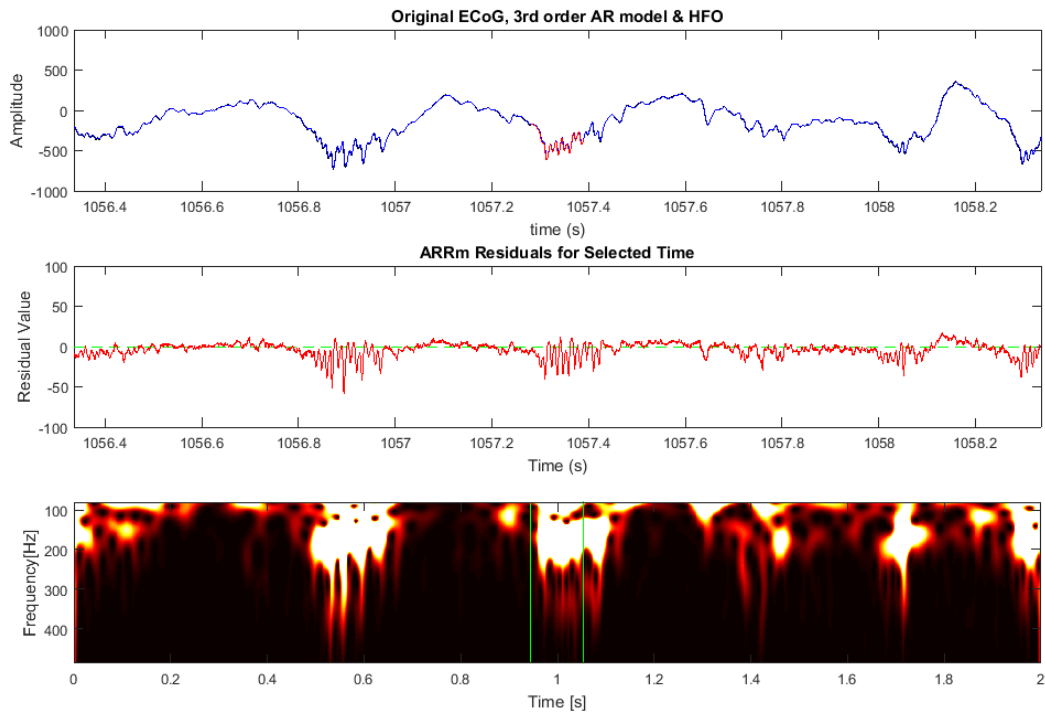


Figure 84: Multiple HFOs

It can be observed in the iEEG that these HFOs are occurring from flat background activity. The ARRm residual stays near 0 during the flat iEEG background activity, but when the HFO occurs, the ARRm residuals spike to approximately -50. This indicates that the ARRm residual does deviate from 0 when HFOs occur, therefore elevating the variation in residuals. The energies of both HFOs seem to be at approximately 200Hz. A segment of iEEG with a fast ripple is shown in Figure 85.

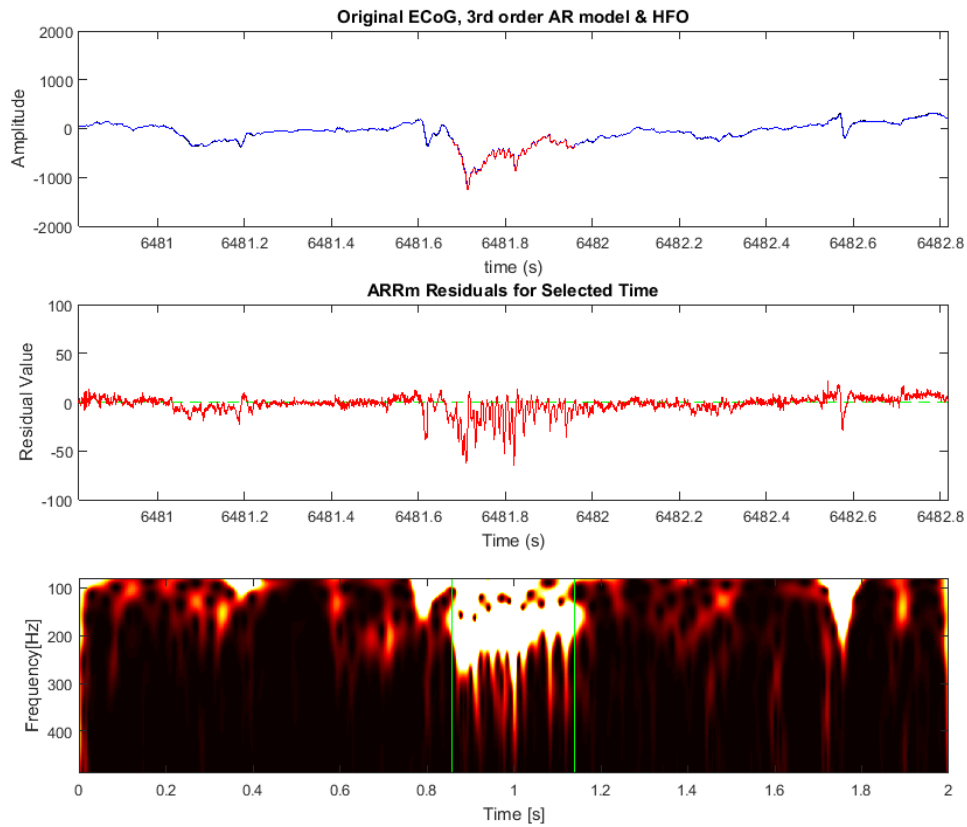


Figure 85: HFO with High Levels of High Frequency Energy > 250Hz, Possibly Classified as a Fast Ripple

It can be observed from the power spectrum that this HFO has high energy >250Hz. This may classify this HFO as a fast ripple. The ARRm residual spikes to a level of about -50 or -75 for the duration of this FR. It is interesting to note the behavior of other channels after this event in the iEEG (Figure 86).

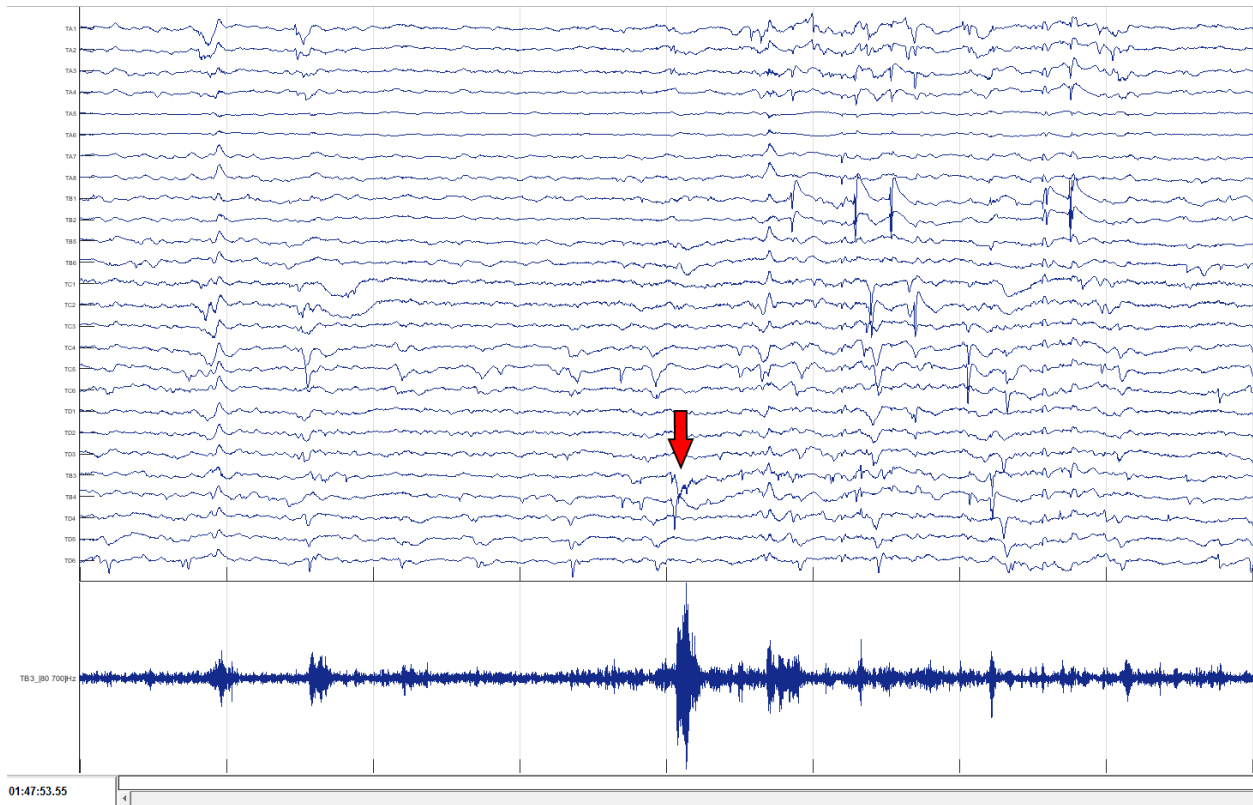


Figure 86: Location of Fast Ripple from Figure 85 in Raw iEEG and Channel TB3 filtered for >80Hz

It can be noted that some spiking occurs on TA1 and TC4-5, then an epileptic discharge follows on all channels, which then leads to sharp spiking, especially on channels TB1 and TB2. From observations in the heat plot (Figure 82), channels TB1 and TB2 had very high ARRm values over time, and in the patient notes, consistent spiking was observed on these channels. The effects of spikes on the ARRm residual is examined next. Spikes and an HFO in the iEEG is shown in Figure 87.

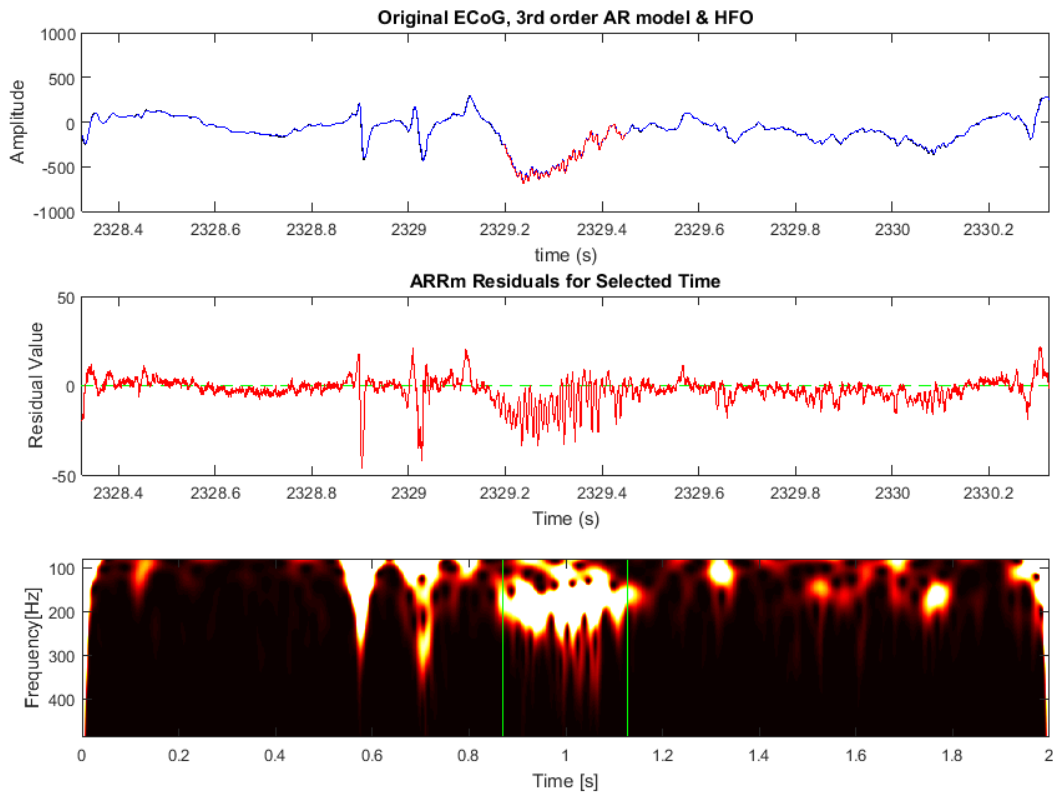


Figure 87: Epileptic Spikes and HFO

At the moment of each spike, it can be noted that the ARRm residual spikes to approximately -50, and when the HFO occurs the residual values are also similarly elevated but not quite to -50. The HFO appears to be at a frequency of approximately 200Hz. An example of a sharp spike and an HFO in the iEEG is shown in Figure 88.

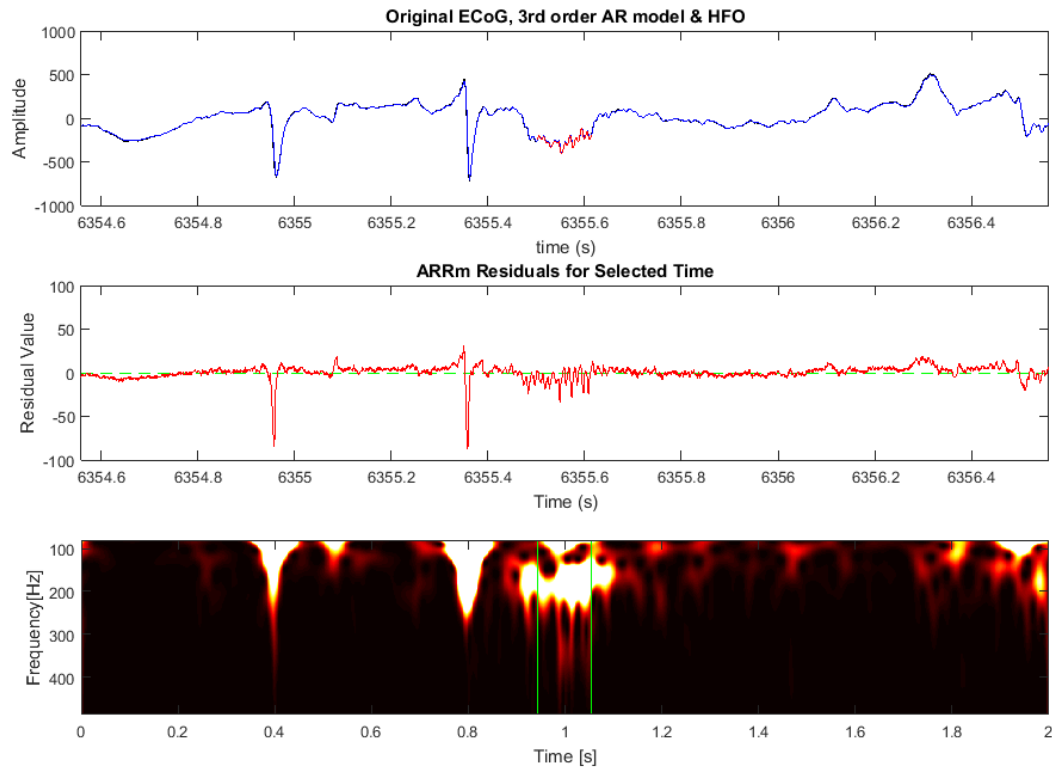


Figure 88: Sharp Spike and HFO

Observing the response of the ARRm residual to sharp spikes, it can be noted that the residual spikes to almost -100, and when the HFO occurs the residual reaches levels similar to previous observations. Therefore, when sharp spikes occur, the residual variation would increase to a very high level. An example of an epileptic discharge and a discharge with a superimposed HFO is shown in Figure 89.

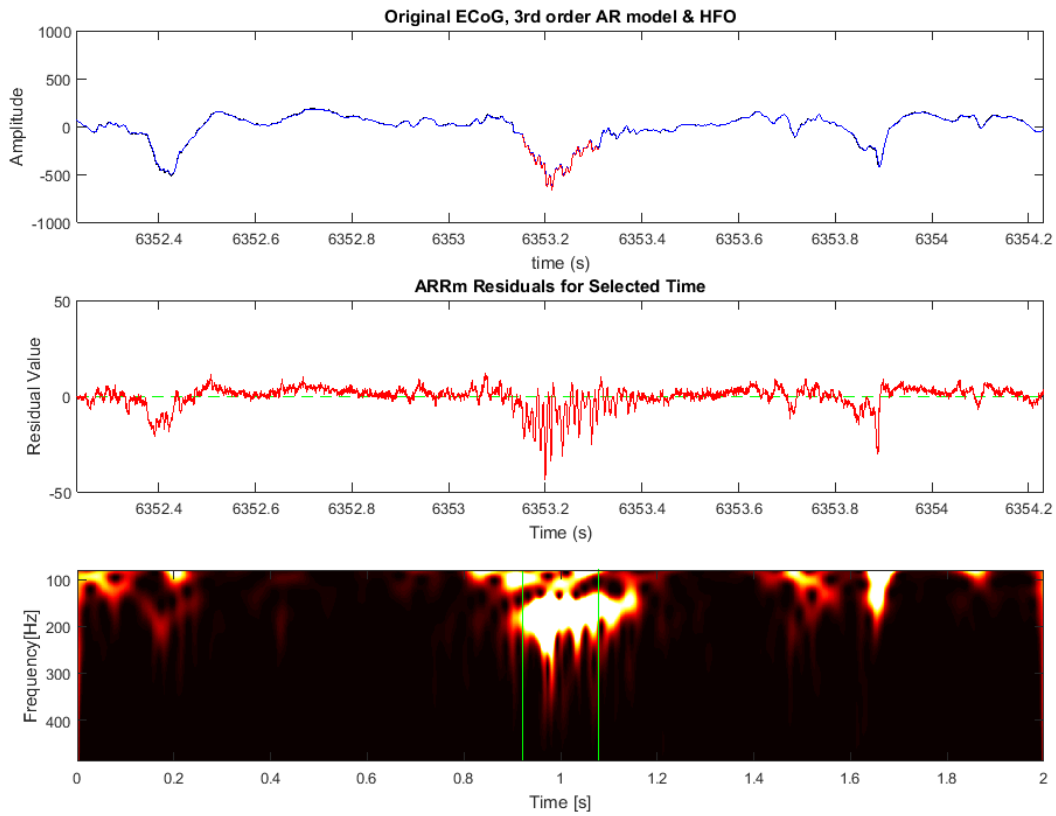


Figure 89: Epileptic Discharge and HFO

Comparing the response of the ARRm residual to the epileptic discharge and the discharge with a superimposed HFO, it can be seen that the residual does not reach as high of a level when the discharge occurs as opposed to the discharge with a superimposed HFO. From this observation, it should be noted that the addition of an HFO superimposed on a discharge or a spike may elevate the residuals more than a discharge or spike alone.

Chapter 5: Discussion and Conclusions

15. Discussion

The results of this study have yielded some interesting points of discussion around the ARRM algorithm and its relation to HFO detection in iEEG. Both of these methods had results where certain channels exhibited a high number of HFOs or high ARRM value throughout the EMU study, and these channels were quite often channels of interest identified by the epileptologist. In most results, high ARRM values show relation to a high number of HFOs detected on channels, but other results have shown an inverse relationship between the ARRM value and HFO count. In addition, it has been observed that for a channel that has a high number of total HFO detections, the adjacent channel has a high median ARRM value, and vice versa. Statistical correlations have revealed that there are significant positive correlations between the ARRM Value and the HFO count over time on some channels of interest, but other results have shown inverse relationships on channels of interest. The question is raised as to what information each method is extracting from the iEEG. The HFO detection method is known to identify high frequency oscillations in the iEEG, and the ARRM value is claimed to quantify the occurrence of nonharmonic events in the iEEG, which may include spikes, high frequency oscillations and more specifically fast ripples.

Using the previously developed Modified Burnos method of HFO detection, it was determined that the original dynamic way of calculating the EOI threshold with the standard deviation and mean of the envelope of the signal needs further development for full automation. Using this method of calculating the EOI threshold, the threshold was too low, which included too much activity in the results. Therefore, channels with the strongest activity over time in the heat plots and Total HFO Count plots were not discernable from other activity. Seizures also had strong values in the results, showing high HFO counts across all channels during the seizure. Another observation was that more activity across all channels was included in the

results for 2000Hz patients than the 1000Hz patient, which is expected since higher frequency content can be quantified with a higher sampling rate.

Using a higher optimal constant threshold for EOI detection on each patient, many of the extra results were filtered out, and the results showed clear evidence of consistently high HFO activity over time on some channels, including channels of interest. Seizure activity also was preserved when using a higher optimal threshold and could be seen as high counts of HFOs across all channels at one point in time. A lower EOI threshold enhanced seizure activity and all other high frequency content in the signal in a wide range of energies. A higher EOI threshold enhanced interictal activity with high frequency content in a higher range of energy and also preserved seizure activity. In future work, the automated calculation of the EOI threshold should be investigated, or a filter should be available in a review tool to change the heat plot based on the desired lower or higher EOI threshold.

The ARRm method showed consistent patterns of high ARRm values that stood out over time with minimal epochs where high ARRm values occurred across all channels, regardless of patient or sampling frequency. This characteristic pattern observed in multiple patients using HFO detection with an optimal constant threshold also was observed for the ARRm value. These results indicate that the ARRm value may represent channels with a high number of HFOs occurring. The strength of this relationship was measured with a linear and nonparametric correlation test. In both instances, several channels of interest showed significant positive correlations ($p < 0.01$) for each patient analyzed. Some observations were made in the heat plots that when a seizure occurred, the ARRm value decreased across all channels, and then returned to the interictal levels when the seizure stopped. For one patient file, there was a channel of interest that showed a significant negative correlation. Observing the channel on the heat plot, it was seen that the HFO rate was high and ARRm value was low before a seizure

occurred, then after the seizure ended, the ARRm value was high and HFO rate was low. In other patients and files, the ARRm value also tends to drop when a seizure occurs, which means that the ARRm method may be insensitive to seizure activity.

The investigation into the behavior of the ARRm residual during different epileptogenic events revealed some patterns that can explain the relationship between the ARRm value and the HFO count. The following events were examined: HFOs (Ripples), Fast Ripple, Spike and HFO, Sharp Spike and HFO, Epileptic discharge and HFO. Out of all these events, sharp spikes in the iEEG elevated the ARRm residual the most, which would therefore raise the residual signal variation as well, in turn increasing the ARRm value. If sharp spikes are disregarded, it was found that spikes or discharges with superimposed HFOs elevated the ARRm residual the most, and FRs generally caused a higher residual than Ripples. Assuming the HFO detection algorithm is truly only detecting HFOs, and considering these observations, a significantly positive correlation between high values of HFO counts and the ARRm value could be relevant to the presence of FRs, or the occurrence of HFOs and spikes together. Both FRs and the occurrence of HFOs with spikes have been shown in literature to be strongly related to the SOZ. A channel with only the highest ARRm values could indicate the presence of consistent sharp spiking in the signal. It would be interesting to modify the ARRm method to disregard sharp spiking, which would tailor the algorithm to only be sensitive to HFOs, therefore possibly drawing stronger correlations between the ARRm value and HFOs detected, or revealing new results.

Since the ARRm method is solely based in the time domain, it has potential for implementation for real time analysis, and is computationally less expensive and less time consuming than methods that include analysis in the frequency domain. Some experimentation on computation time was done. It is apparent that the speed of both methods is dependent on the sampling frequency and number of channels, but the speed of the HFO detection method is

also dependent on the number of EOIs detected. With each EOI that is detected, the HFO detection method must run analysis in the frequency domain to determine if it should be classified as an HFO. Running the same file with 58 channels, 2hrs in length, sampled at 2000Hz, the ARRM method took 1hr 25min., the HFO detection method with an optimized constant threshold took 2hrs, and the HFO detection method with a calculated higher threshold took 10hrs 35min. Running the ARRM method on a 1000Hz file with 58 channels took 50min. By these findings, the ARRM method has shown to be very computationally efficient compared to the HFO detection method. An embedded system implementation of the ARRM method was done on a smaller scale (lower sampling frequency, 1 channel) for a proof-of-concept real-time EEG collection/analysis. EEG was sampled, the signal was modeled with a 3rd order AR model, and the ARRM value was calculated in real time. In future work, this implementation could be scaled up to more channels and higher sampling frequencies on more sophisticated hardware equipped with a fast CPU and the appropriate amount of RAM.

The statistical analysis has revealed that the positive correlation between high ARRM values and high HFO counts is significant for some channels of interest for each patient. Due to the behavior of the ARRM residual examined during various epileptogenic events, this significant relationship may indicate the consistent presence of FRs or spikes superimposed with HFOs on that channel. In order to make a stronger statement about this correlation, more patient files should be analyzed in the future.

16. Conclusion

In reference to the points made in my thesis proposal, various approaches of examining the ARRM and HFO methods of identifying channels of interest in the iEEG data from the EMU

were performed. These approaches include heat plot comparison by time (beginning, middle, and end of EMU stay), median ARRm value and Total HFO count bar plots, and comparison side by side between the ARRm value over time and HFO count over time. The HFO method of identifying channels of interest was validated with an optimal constant threshold to detect high preponderances of HFOs over time on channels of interest to the epileptologist for multiple patients. The ARRm method was developed in MATLAB to post-process iEEG data and compare to the results of the HFO detection method. Since initial results showed promise of correlation to the HFO detection results, the ARRm method was not modified further. Instead, analysis focused on the ability of the ARRm method to quantify nonharmonicity over long-term recordings of iEEG as opposed to the analysis of short-term recordings that have been the focus of the studies published.

Both the HFO detection method and the ARRm method were applied to multiple patient iEEG recordings, one patient at sampled at 1000Hz and two at 2000Hz. Analyzing the heat plot of results from each method, it was found that consistently high values of the ARRm value and HFO count over time were related to channels of interest identified by the epileptologist for multiple patients. By statistical analysis using a linear and nonparametric correlation tests, it was found that in some channels of interest on each patient, high values of the ARRm value and HFO count over time were significantly correlated ($p < 0.01$). Investigation into the behavior of the ARRm residual during different epileptogenic events showed that a significant positive correlation between channels of interest could indicate the presence of Fast Ripples or the occurrence of HFOs and spikes together, both which have been shown in literature to be strongly related to the seizure onset zone. The ARRm method shows promise as a computationally efficient alternative or a supplement to HFO detection for automating identification of the seizure onset zone in patients with epilepsy.

Appendices

ARRm/HFO Median/Total HFO Count Plots

Patient WDH-022

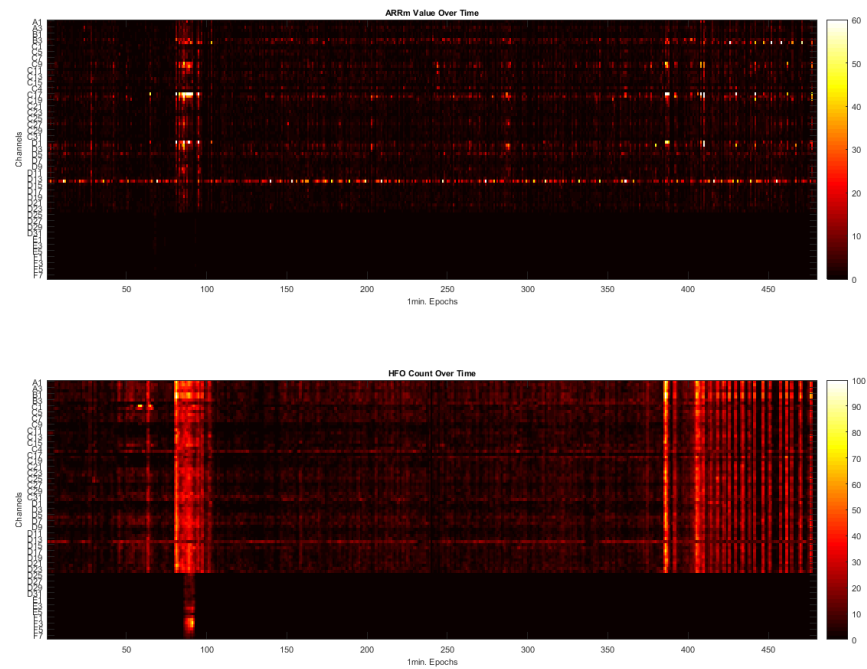


Figure 90: ARRm Value and HFO Count Over Time from the Middle of EMU Stay – Patient WDH-022

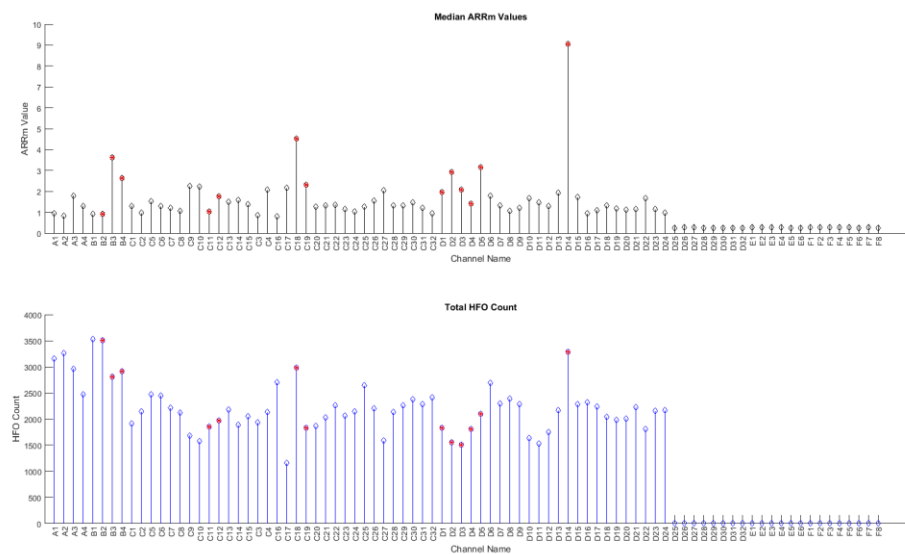


Figure 91: Median ARRm Value and Total HFO Count from the Middle of EMU Stay – Patient WDH-022

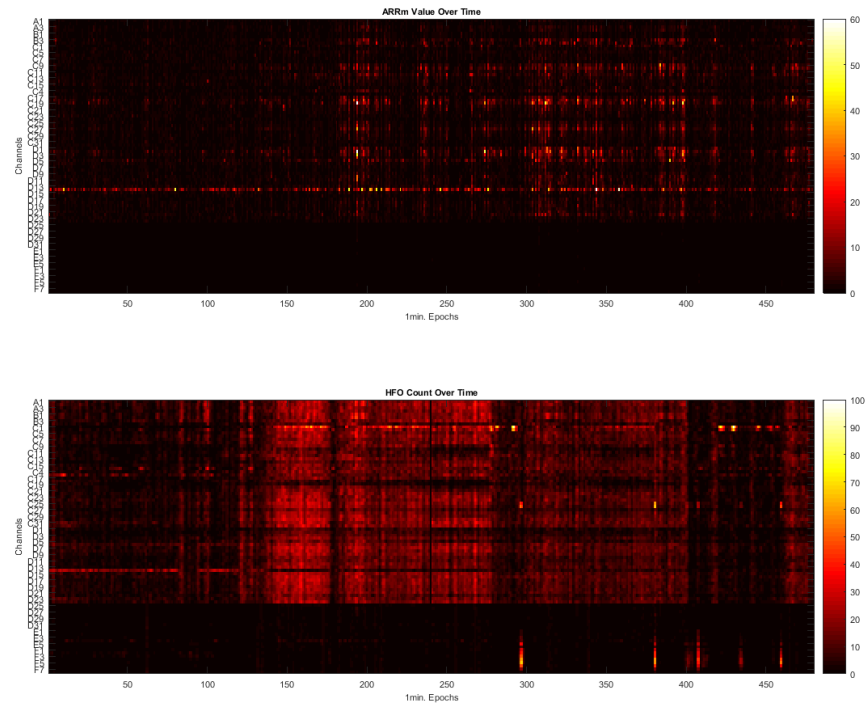


Figure 92: ARRm Value and HFO Count Over Time from the End of EMU Stay – Patient WDH-022

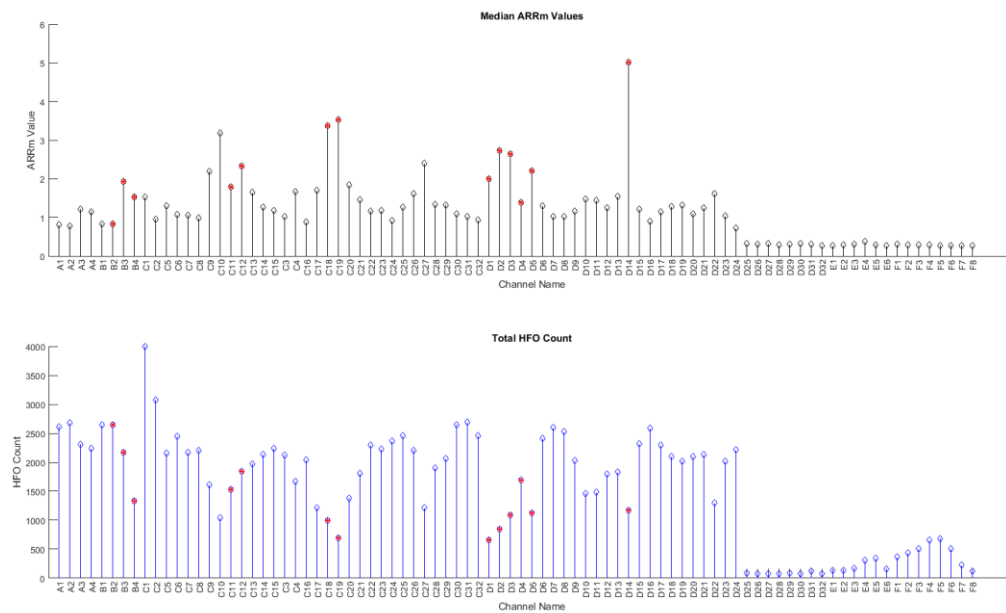


Figure 93: Median ARRm Value and Total HFO Count from the End of EMU Stay – Patient WDH-022

Patient DJA-033

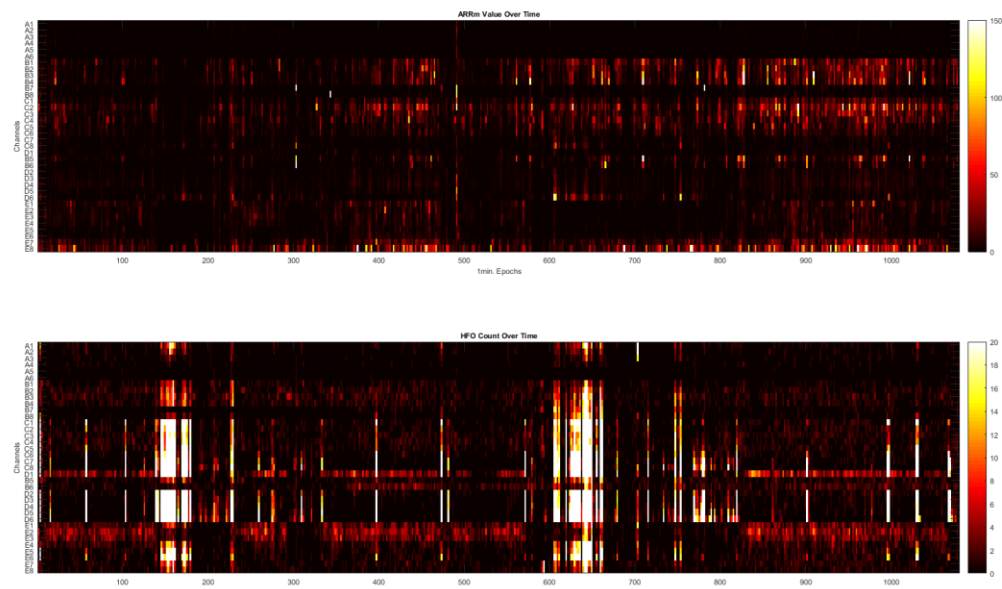


Figure 94: ARRm Value and HFO Count Over Time from the Middle of EMU Stay – Patient DJA-033

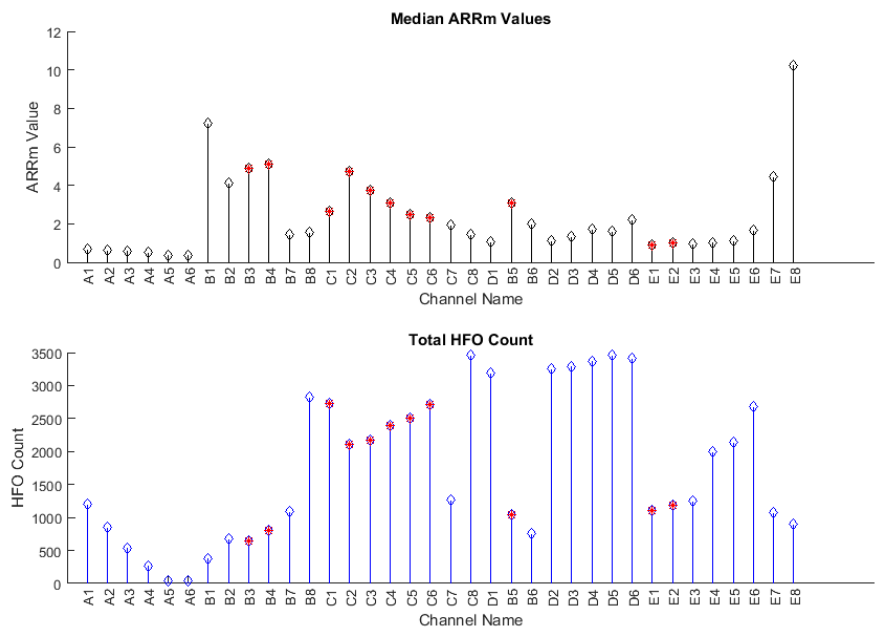


Figure 95: Median ARRm Value and Total HFO Count from the Middle of EMU Stay – Patient DJA-033

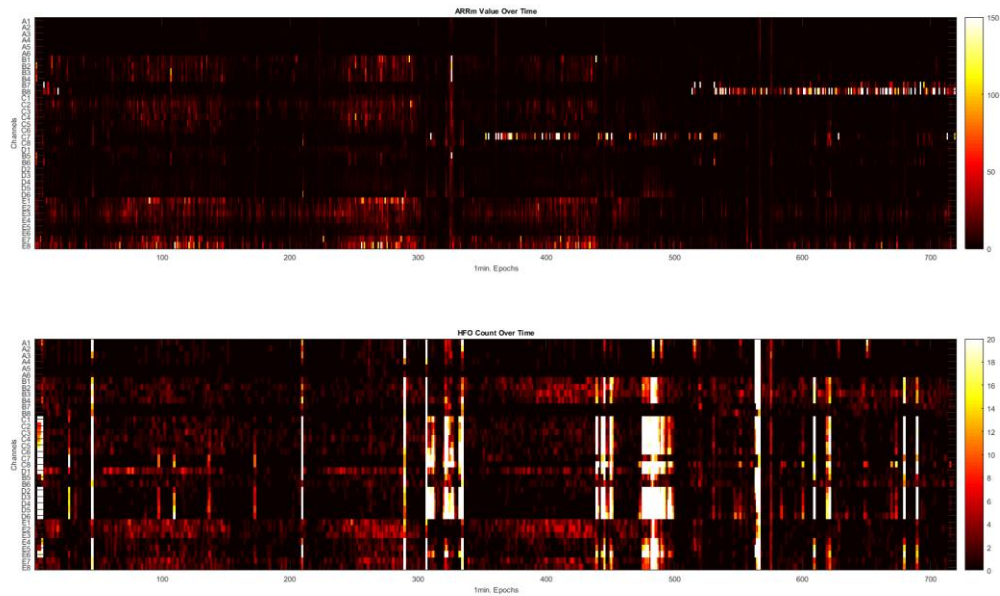


Figure 96: ARRm Value and HFO Count Over Time from the End of EMU Stay – Patient DJA-033

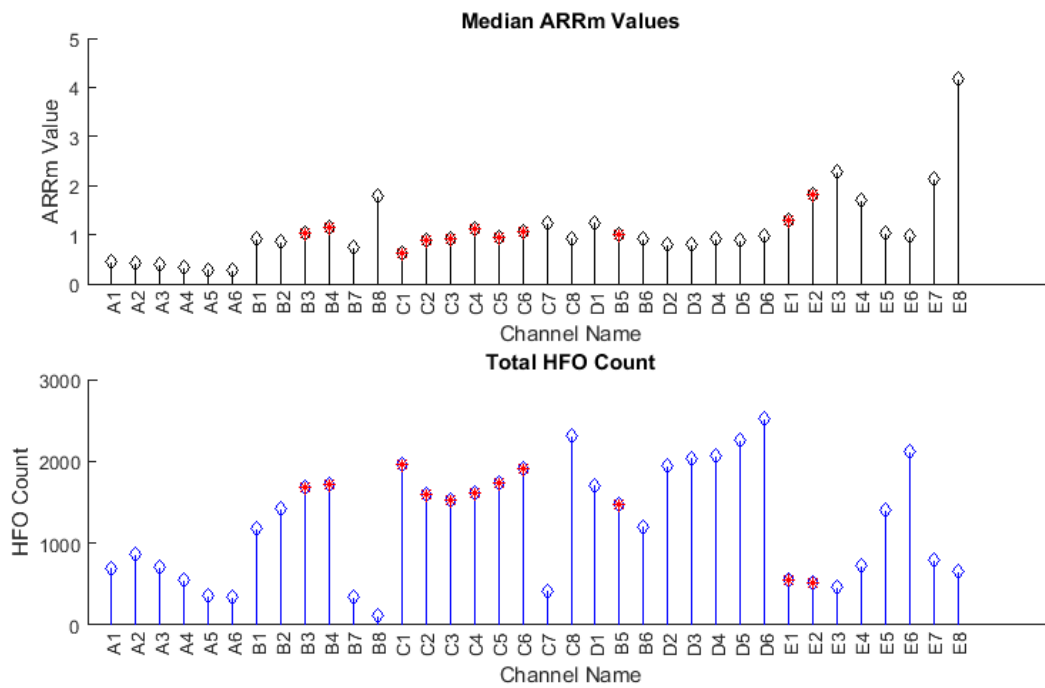


Figure 97: Median ARRm Value and Total HFO Count from the End of EMU Stay – Patient DJA-033

Patient JMM-025

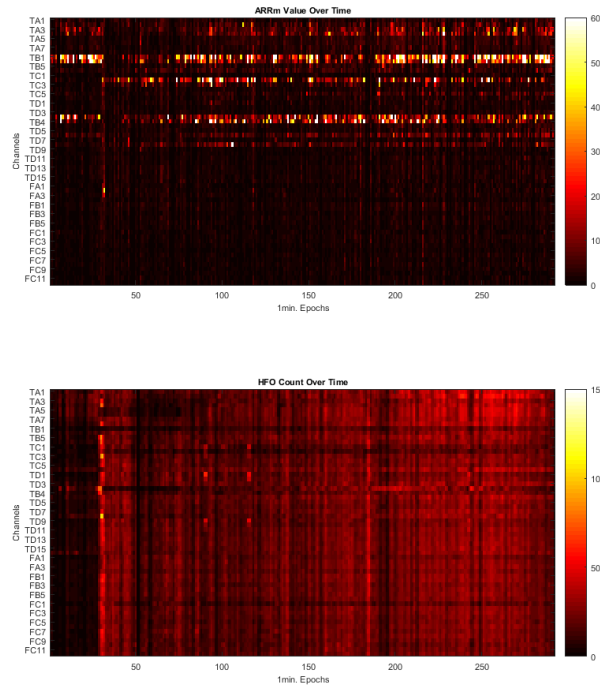


Figure 98: ARRm Value and HFO Count Over Time from Second Group of Files at the Beginning of EMU Stay – Patient JMM-025

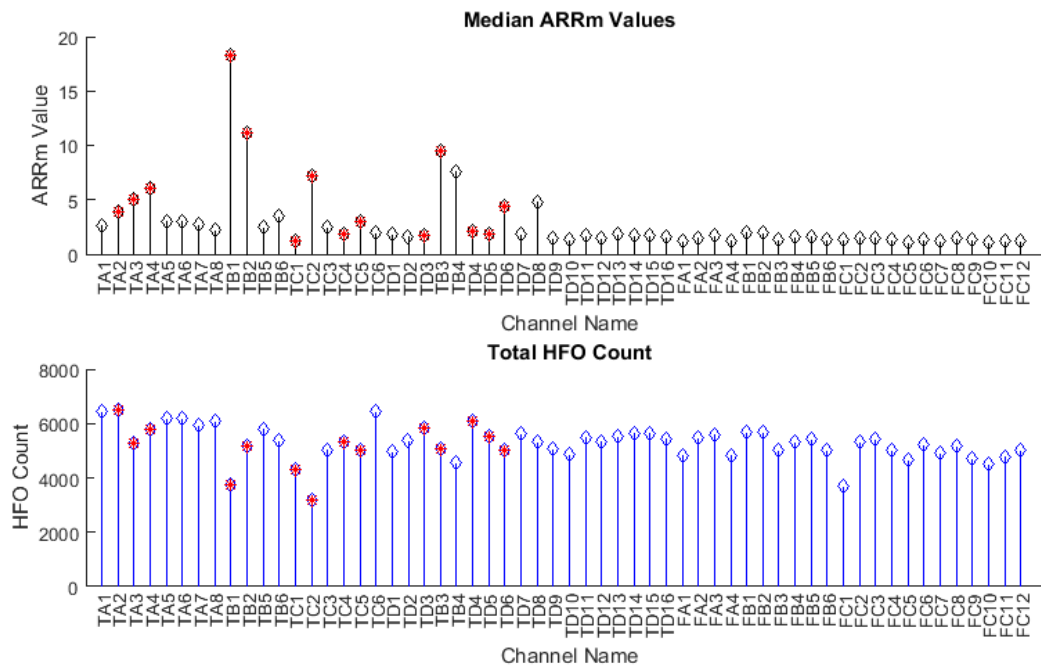


Figure 99: Median ARRm Value and Total HFO Count from the Second Group of Files at the Beginning of EMU Stay – Patient JMM-025

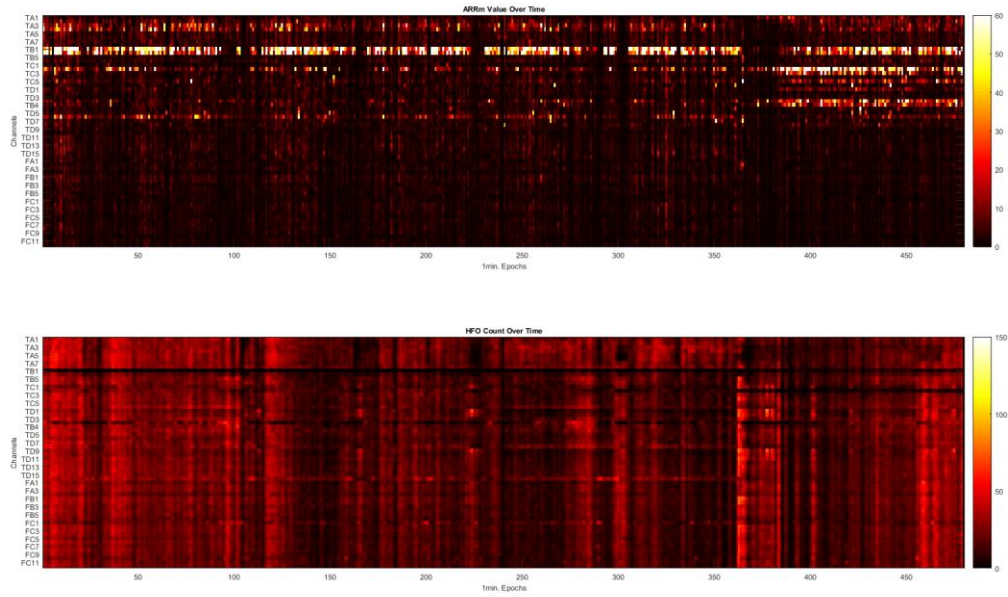


Figure 100: ARRm Value and HFO Count Over Time from First Group of Files at the Middle of EMU Stay – Patient JMM-025

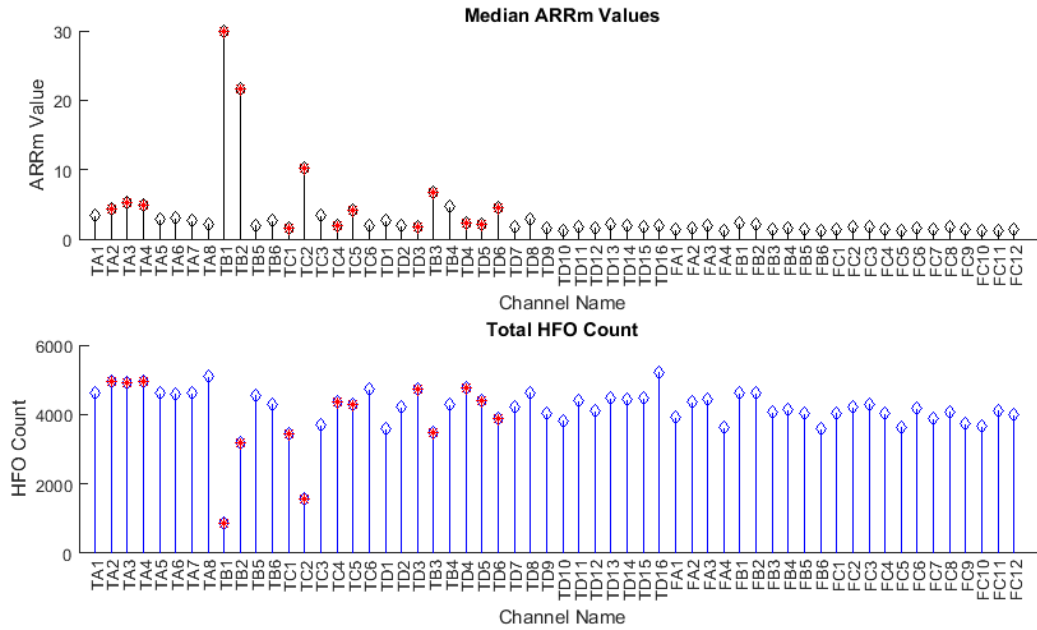


Figure 101: Median ARRm Value and Total HFO Count from the First Group of Files at the Middle of EMU Stay – Patient JMM-025

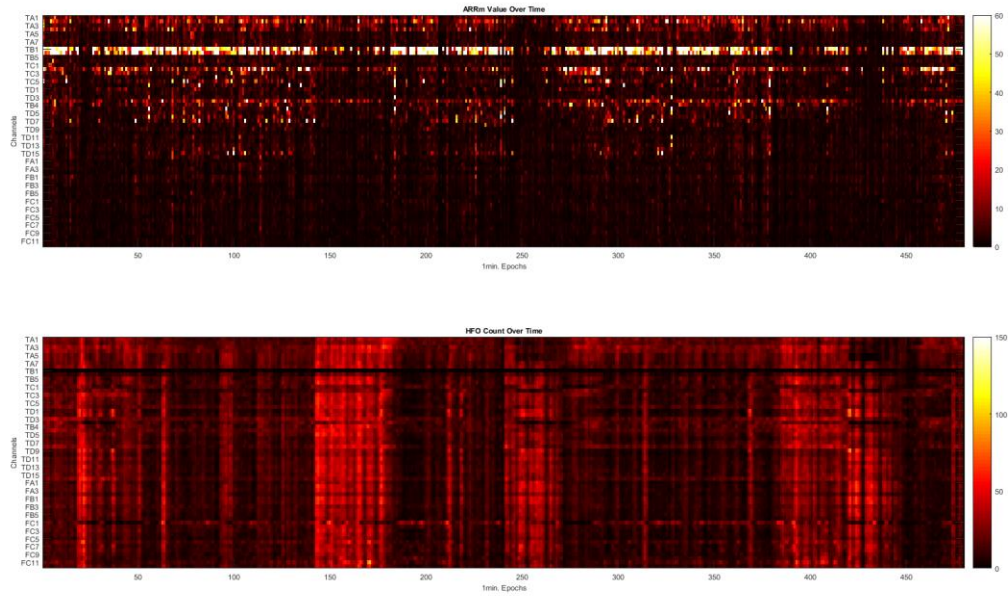


Figure 102: ARRm Value and HFO Count Over Time from Second Group of Files at the Middle of EMU Stay – Patient JMM-025

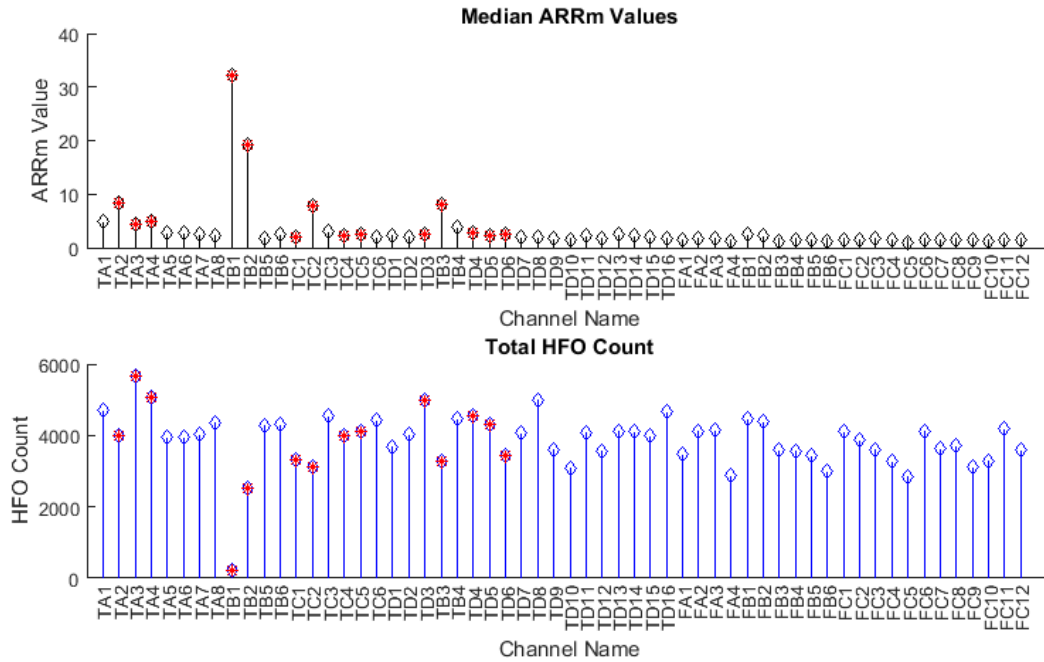


Figure 103: Median ARRm Value and Total HFO Count from the Second Group of Files at the Middle of EMU Stay – Patient JMM-025

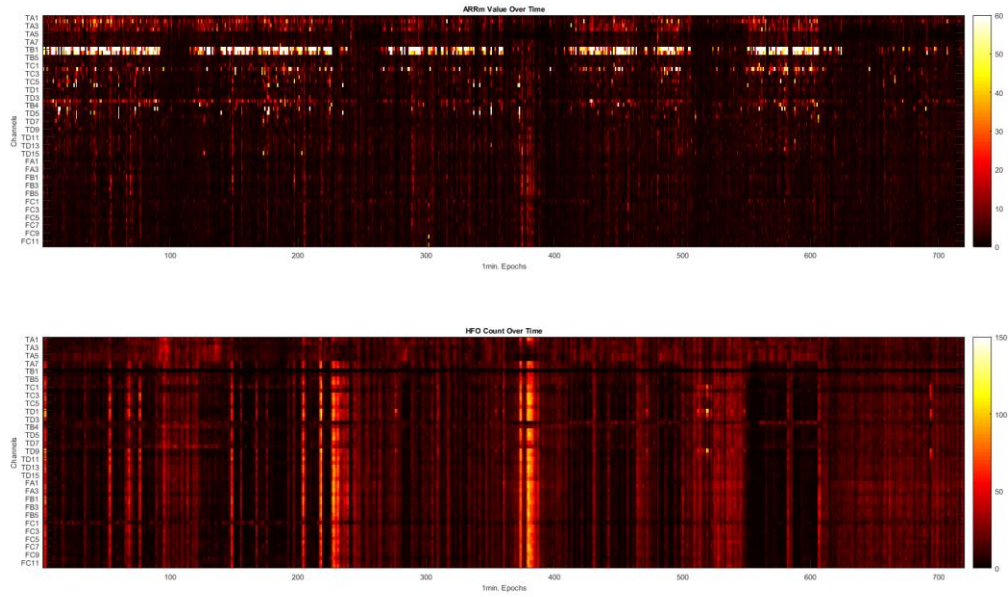


Figure 104: ARRm Value and HFO Count Over Time from the End of EMU Stay – Patient JMM-025

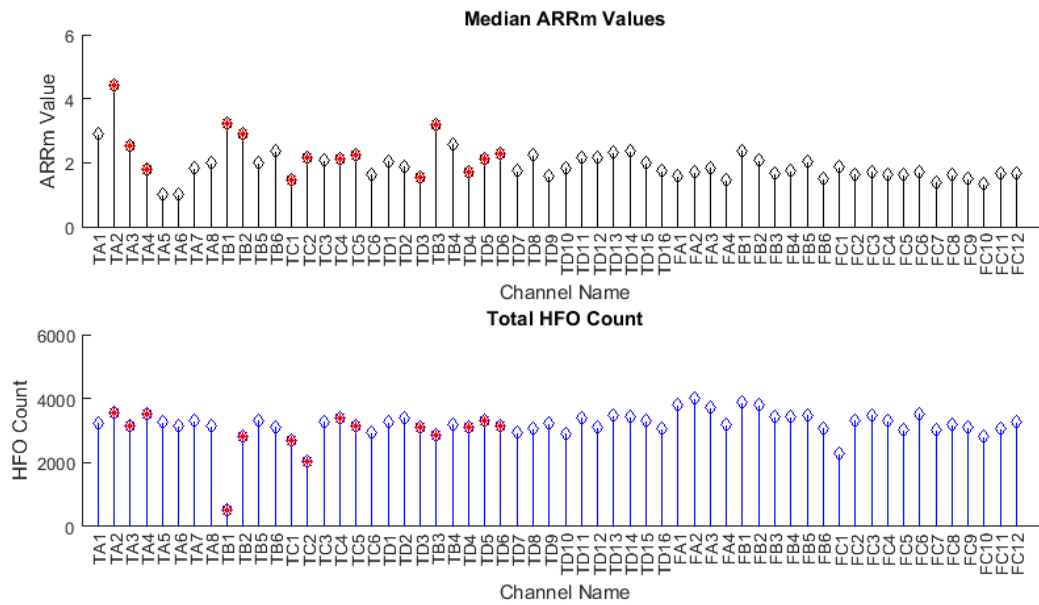


Figure 105: Median ARRm Value and Total HFO Count from the End of EMU Stay – Patient JMM-025

MATLAB Code

RIPPLELAB Implementation of Modified Burnos Method

```
function [m_HFOEvents,EOI_lims] = f_findHFOxRIAZ(pstr_SignalPath,ps_SignalIdx,...
                                                st_HFOData,s_SampleFreq,...
                                                channelLabel,st_waitOutput)
%_findHFOxRIAZ This function analyzes the EEG signal for HFOs
% The modified Burnos method is implemented for HFO Detection

if ~isempty(st_waitOutput)
    p_HFOwaitFigure(st_waitOutput,...
                    'MethPatch',0)
end

% -----
% Get eeg signal
m_Data      = [];
load(pstr_SignalPath)
Signal      = m_Data(:,ps_SignalIdx);
%*****
Signal = double(Signal); %ADDED

clear m_Data

% -----
% set parameters
fs = s_SampleFreq;
hp = st_HFOData.s_FreqIni;
lp = st_HFOData.s_FreqEnd;
channel_name = channelLabel;

clear st_HFOData
% -----

% main parameters
HFOobj.time_thr = ceil(0.006*fs); % 6 ms
HFOobj.fs = fs;
HFOobj.channel_name = channel_name;
% filtering
HFOobj.hp = hp;
HFOobj.lp = lp;
HFOobj.Fst1 = (hp-10)/(fs/2); % parameters for filtering
HFOobj.Fp1 = hp/(fs/2);
HFOobj.Fp2 = lp/(fs/2);
HFOobj.Fst2 = (lp+10)/(fs/2);
HFOobj.Ast1 = 40;
HFOobj.Ap = 0.5;
HFOobj.Ast2 = 40;
% merge IoEs
HFOobj.maxIntervalToJoin = 0.01*HFOobj.fs; % 10 ms
% reject events with less than 6 peaks
```

```

HFOobj.minNumberOscillations = 6;
HFOobj.dFactor = 2;
% Stage 2
HFOobj.bound_min_peak = 40; % Hz, minimum boundary for the lowest ("deepest") point
HFOobj.ratio_thr = 0.5; % threshold for ratio
HFOobj.min_trough = 0.2; % 20 %
HFOobj.limit_fr = 500;
HFOobj.start_fr = 60; % limits for peak frequencies

% -----
% 1.
% filtering

str_Message = 'Filtering Signal - Step 1 ....';
p_HFOwaitFigure(st_waitOutput,...
    'LogsList',str_Message)
p_HFOwaitFigure(st_waitOutput,...
    'MethPatch',0)

Signal_filtered=filtering(Signal, HFOobj);

str_Message = 'Filtering Signal - OK';
p_HFOwaitFigure(st_waitOutput,...
    'LogsList',str_Message)
p_HFOwaitFigure(st_waitOutput,...
    'MethPatch',1/12)

% -----
% 2.
% envelope

str_Message = 'Calculating Envelope - Step 2 ....';
p_HFOwaitFigure(st_waitOutput,...
    'LogsList',str_Message)

%get the envelope of the filtered signal using the Hilbert transform
env = abs(hilbert(Signal_filtered));

str_Message = 'Calculating Envelope - OK';
p_HFOwaitFigure(st_waitOutput,...
    'LogsList',str_Message)
p_HFOwaitFigure(st_waitOutput,...
    'MethPatch',1/6)

% -----
% 3.
% threshold

str_Message = 'Calculating Threshold - Step 3 ....';
p_HFOwaitFigure(st_waitOutput,...
    'LogsList',str_Message)

HFOobj.THR = 3 * std(env) + mean(env);

```

```

% ----- Original Author's comments from previous thesis -----
%HFOobj.THR = 4 * std(env) + mean(env)
% HFOobj.THR =27.19761355;
% HFOobj.THR =21.39934218;
%threshold was 18 for WDH patient
% HFOobj.THR=9.4631; % for SH-EEG files 25% increment from average

% -----Code implemented for optimized thresholding for this thesis -----
%WDH-022 Threshold
%HFOobj.THR = 18.0;

%DJA-033 Threshold
%HFOobj.THR = 4.0;

%JMM-025 Threshold
%HFOobj.THR = 30.0;

str_Message      = 'Calculating Threshold - OK';
p_HFOwaitFigure(st_waitOutput,...
                'LogsList',str_Message)
p_HFOwaitFigure(st_waitOutput,...
                'MethPatch',1/4)

% -----
% 4.
% Stage 1 - detection of EoIs
str_Message      = 'Detection of EoIs - Step 4 ....';
p_HFOwaitFigure(st_waitOutput,...
                'LogsList',str_Message)

env(1)=0; env(length(env))=0; % assign the first and last positions at 0 point

pred_env(2:length(env))=env(1:length(env)-1);
pred_env(1)=pred_env(2);
if size(pred_env,1)~=size(env,1) % check the size if it's not been transposed
    pred_env=pred_env';
end

t1=find(pred_env<(HFOobj.THR/2) & env>=(HFOobj.THR/2)); % find zero crossings rising
t2=find(pred_env>(HFOobj.THR/2) & env<=(HFOobj.THR/2)); % find zero crossings falling

trig=find(pred_env<HFOobj.THR & env>=HFOobj.THR); % check if envelope crosses the THR level
rising
trig_end=find(pred_env>=HFOobj.THR & env<HFOobj.THR); % check if envelope crosses the THR level
falling

nDetectionCounter = 0;
% initialize struct
Detections=struct('channel_name','', 'start','', 'peak','', 'stop','', ...
                'peakAmplitude', '');

```

```

% check every trigger point, where envelope crosses the threshold,
% find start and end points (t1 and t2), t2-t1 = duration of event;
% start and end points defined as the envelope crosses half of the
% threshold for each EOIs

for i=1: numel(trig)

    % check for time threshold duration, all times are in pt
    if trig_end(i)-trig(i) >= HFOobj.time_thr

        nDetectionCounter = nDetectionCounter + 1;
        k=find(t1<=trig(i) & t2>=trig(i)); % find the starting and end points of envelope
        Detections(nDetectionCounter).channel_name = HFOobj.channel_name;

        % check if it does not start before 0 moment
        if t1(k)>0
            Detections(nDetectionCounter).start = t1(k);
        else
            Detections(nDetectionCounter).start = 1;
        end

        % check if it does not end after last moment
        if t2(k) <= length(env)
            Detections(nDetectionCounter).stop = t2(k);
        else
            Detections(nDetectionCounter).stop = length(env);
        end

        [ peakAmplitude , ind_peak ] = max(env(t1(k):t2(k)));
        Detections(nDetectionCounter).peak = (ind_peak + t1(k));
        Detections(nDetectionCounter).peakAmplitude = peakAmplitude;

    end
end

% saving EOI
%%*****

EOIstart = zeros(length(Detections),1);
EOIstop = zeros(length(Detections),1);

% if length(Detections) > 1
%     for i = 1:length(Detections)
%         EOIstart(i) = Detections(i).start;
%         EOIstop(i) = Detections(i).stop;
%     end
% end

if length(Detections) < 1
    str_Message = 'NO EOIs DETECTED';
    p_HFOwaitFigure(st_waitOutput,...
        'LogsList',str_Message)
    m_HFOEvents = horzcat(EOIstart,EOIstop);
    return;
end

```

```

end

%%*****

str_Message      = 'Detection of EoIs - OK';
p_HFOWaitFigure(st_waitOutput,...
                'LogsList',str_Message)
p_HFOWaitFigure(st_waitOutput,...
                'MethPatch',1/2)

str_Message      = 'Validating HFOs - Step 5 ....';
p_HFOWaitFigure(st_waitOutput,...
                'LogsList',str_Message)

if (nDetectionCounter > 0)

    % -----
    % 5.
    % Merge EoIs
    joinedDetections = joinDetections(Detections, HFOobj);

    % -----
    % 6.
    % Check for sufficient number of oscillations
    results = checkOscillations(joinedDetections, signal_filtered, ...
                                mean(abs(signal_filtered)), std(abs(signal_filtered)), HFOobj);

    % -----
    % 7.
    % Stage 2 - recognition of HFOs among EoIs
    results = PS_validation_all(results, signal, env, HFOobj);

    m_HFOEvents = zeros(length(results),2);
    for i = 1:length(results)
        m_HFOEvents(i,1) = results(i).start*HFOobj.fs;
        m_HFOEvents(i,2) = results(i).stop*HFOobj.fs;
    end

    if isempty(results)
        m_HFOEvents = [0 0];
    end

    %m_HFOEvents = [results(:).start results(:).stop]

else
    % initialize struct
    results(1).channel_name = channel_name;
    results(1).start = -1;
    results(1).stop = -1;
    results(1).peak = -1;
    results(1).peakHFOFrequency = 0;
    results(1).troughFrequency = 0;
    results(1).peakLowFrequency = 0;
    results(1).peakAmplitude = 0;

```

```

        m_HFOEvents = [0 0];
    end

    str_Message      = 'validating HFOs - OK';
    p_HFOwaitFigure(st_waitOutput,...
                    'LogsList',str_Message)
    p_HFOwaitFigure(st_waitOutput,...
                    'MethPatch',1)

end

% =====
function result=filtering(Signal, p)

% Filter Signal in the range [hp lp]
% -----
% filtering by using IIR Cauer filter
size_check=size(Signal,1);

fil_design=fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2',...
    p.Fst1,p.Fp1,p.Fp2,p.Fst2,p.Ast1,p.Ap,p.Ast2);
filter_design=design(fil_design,'ellip'); % band pass IIR filter
[B,A]= sos2tf(filter_design.sosMatrix,filter_design.ScaleValues);
Signal=filtfilt(B,A,Signal); %zero-phase filtering

% check the size if it's not been transposed
if size(Signal,1)~=size_check
    Signal=Signal';
end

result=Signal;
end

% =====
function joinedDetections = joinDetections(Detections, p)

% Merge EoIs with inter-event-interval less than 10 ms into one EoI
nOrigDetections = length(Detections);

% fill result with first detection
joinedDetections = struct('channel_name','','start','','peak','','stop','');
joinedDetections(1).channel_name = Detections(1).channel_name;
joinedDetections(1).start = Detections(1).start;
joinedDetections(1).stop = Detections(1).stop;
joinedDetections(1).peak = Detections(1).peak;
joinedDetections(1).peakAmplitude = Detections(1).peakAmplitude;
nDetectionCounter = 1;

for n = 2 : nOrigDetections

    % join detection
    if Detections(n).start > joinedDetections(nDetectionCounter).start

```

```

nDiff = Detections(n).start - joinedDetections(nDetectionCounter).stop;

if nDiff < p.maxIntervalToJoin

    joinedDetections(nDetectionCounter).stop = Detections(n).stop;

    if joinedDetections(nDetectionCounter).peakAmplitude < ...
        Detections(n).peakAmplitude

        joinedDetections(nDetectionCounter).peakAmplitude = ...
            Detections(n).peakAmplitude;
        joinedDetections(nDetectionCounter).peak=Detections(n).peak;

    end

else
    % initialize struct
    nDetectionCounter = nDetectionCounter + 1;
    joinedDetections(nDetectionCounter).channel_name = Detections(n).channel_name;
    joinedDetections(nDetectionCounter).start = Detections(n).start;
    joinedDetections(nDetectionCounter).stop = Detections(n).stop;
    joinedDetections(nDetectionCounter).peak = Detections(n).peak;
    joinedDetections(nDetectionCounter).peakAmplitude = Detections(n).peakAmplitude;

end
end
end

end

% =====
function checkedOscillations = checkOscillations(Detections, Signal, ...
    AbsoluteMean, AbsoluteStd, p)

% Reject events not having a minimum of 6 peaks above 2 SD
% -----
% set parameters
nDetectionCounter = 0;

for n = 1 : length(Detections)

    % get EEG for interval
    intervalEEG = Signal(Detections(n).start : Detections(n).stop);

    % compute abs values for oscillation interval
    absEEG = abs(intervalEEG);

    % look for zeros
    zeroVec=find(intervalEEG(1:end-1).*intervalEEG(2:end)<0);
    nZeros=numel(zeroVec);

    nMaxCounter = 0;

    if nZeros > 0

```



```

% look for maxima with sufficient amplitude between zeros
for ii = 1 : nZeros-1

    lStart = zeroVec(ii);
    lEnd   = zeroVec(ii+1);
    dMax = max(absEEG(lStart:lEnd));

    if dMax > AbsoluteMean + p.dFactor * AbsoluteStd;

        nMaxCounter = nMaxCounter + 1;

    end
end

if nMaxCounter >= p.minNumberOscillations

    nDetectionCounter = nDetectionCounter + 1;

    checkedOscillations(nDetectionCounter).channel_name = ...
        Detections(n).channel_name; %#ok<AGROW>
    checkedOscillations(nDetectionCounter).start = ...
        Detections(n).start; %#ok<AGROW>
    checkedOscillations(nDetectionCounter).stop = ...
        Detections(n).stop; %#ok<AGROW>
    checkedOscillations(nDetectionCounter).peak = ...
        Detections(n).peak; %#ok<AGROW>
    checkedOscillations(nDetectionCounter).peakHFOFrequency = 0; %#ok<AGROW>
    checkedOscillations(nDetectionCounter).troughFrequency = 0; %#ok<AGROW>
    checkedOscillations(nDetectionCounter).peakLowFrequency = 0; %#ok<AGROW>
    checkedOscillations(nDetectionCounter).peakAmplitude = ...
        Detections(n).peakAmplitude; %#ok<AGROW>

end
end

if nDetectionCounter < 1

    % initialize struct
    checkedOscillations(1).channel_name = p.channel_name;
    checkedOscillations(1).start = -1;
    checkedOscillations(1).stop = -1;
    checkedOscillations(1).peak = -1;
    checkedOscillations(1).peakHFOFrequency = 0;
    checkedOscillations(1).troughFrequency = 0;
    checkedOscillations(1).peakLowFrequency = 0;
    checkedOscillations(1).peakAmplitude = 0;

end
end

% =====
function PSvalidated = PS_validation_all(Detections, Signal, env, p)

```

```

% Stage 2 - recognition of HFOs among EOIs
% -----
% set parameters
nDetectionCounter = 0;

for n = 1 : length(Detections)
    % line below limits EOIs to time length .5 seconds, i.e. the
    % difference is less than p.fs*1.
    if (Detections(n).peak ~= -1) && (Detections(n).stop-Detections(n).start< p.fs*1)

        % find the sec interval where the peak occurs
        det_start = Detections(n).peak-Detections(n).start;
        det_stop = Detections(n).stop-Detections(n).peak;

        % define 0.5 sec interval where HFOs occur and take for
        % analysis 0.1 sec before + interval (0.5 sec) + 0.4 sec after
        % in total 1 sec around an HFO is analyzed

        if floor(Detections(n).peak/(p.fs/2)) == 0 % First 0.5 sec

            det_peak = Detections(n).peak;
            intervalST=Signal(1: p.fs);
            interval_env=env(1: p.fs);

        elseif (floor(Detections(n).peak/(p.fs/2)) == ...
            length(Signal)/(p.fs/2) -1) % last 0.5 sec

            det_peak = mod(Detections(n).peak, p.fs);
            intervalST=Signal(length(Signal)-p.fs+1: length(Signal));
            interval_env=env(length(Signal)-p.fs+1: length(Signal));

        else

            det_peak = mod(Detections(n).peak, (p.fs/2))+floor(0.1*p.fs);
            t_peak_interval = floor(Detections(n).peak/(p.fs/2));

            %*****
            % CHECKING INDICES
            %*****

            % [t_peak_interval*p.fs/2+1-floor(0.1*p.fs),...
            %     t_peak_interval*p.fs/2+ceil(0.9*p.fs),length(Signal)]
            intervalST = Signal(t_peak_interval*p.fs/2+1-floor(0.1*p.fs):...
                min(size(Signal), t_peak_interval*p.fs/2+ceil(0.9*p.fs)));
            interval_env=env(t_peak_interval*p.fs/2+1-floor(0.1*p.fs):...
                min(size(Signal), t_peak_interval*p.fs/2+ceil(0.9*p.fs)));

        end

        %-----
        % calculate S-transform frequency from 0 to limit_fr
        [STSignal, ~, ~] = st(intervalST, 0, p.limit_fr, 1/p.fs, 1); % S-transform

```

```

%*****
%****ADDED AN UPPER LIMIT TO INDICES SO WE WEREN'T ACCESSING THE
%ST TRANSFORM OR ENVELOPE OUTSIDE ITS LENGTH
%*****

upper_index=length(interval_env);

% -----
% analyze instantaneous power spectra
true_HFO=0; % counter for recognized HFOs

for tcheck = ...
    max(det_peak-det_start,1):1:min(det_peak+det_stop,upper_index)

    % check if the envelope is above half of the peak+threshold

    if interval_env(tcheck)>1/2*(Detections(n).peakAmplitude+p.THR)

        % for maximum upper start_f frequenc
        [maxV, maxF] = max(abs(STSignal(p.start_fr:end, tcheck)).^2); % HFO peak
        maxF = maxF + p.start_fr-1;

        % search for minimum before found maximum
        [ minV , minF] = min(abs(STSignal(p.bound_min_peak:maxF, tcheck)).^2); %the
trough
        minF=minF+p.bound_min_peak-1;

        % check for sufficient difference
        if isempty(abs(STSignal(1:minF, tcheck)).^2)
            peaks=[];
        else
            [peaks, ~]=findpeaks(abs(STSignal(1:minF, tcheck)).^2); % Low frequcny peak
            end

            if isempty(peaks)
                fpeaks=floor(minF/2);
                peaks=abs(STSignal(fpeaks, tcheck)).^2;
            end

            if isempty(peaks)
                ratio_HFO=0;
                ratio_LowFr=0;
            else
                ratio_HFO = 10*log10(maxV) - 10*log10(minV); % ratio between HFO peak and the
trough
                ratio_LowFr = 10*log10(peaks(end)) - 10*log10(minV); % ratio between Low
Frequency peak and the trough
            end

            % check the difference and check for sufficient trough
            if (upper_index>0) && (ratio_HFO > p.ratio_thr * ratio_LowFr) &&

```

```

(ratio_HFO>p.min_trough*10*log10(maxV)) && (maxF<500)

    true_HFO=true_HFO+0;

else

    true_HFO=true_HFO+1;

end

end

end

if (upper_index>0)&&(true_HFO==0) % all conditions are satisfied
    disp('HFO detected');
    % search for peak
    tcheck=det_peak;
    [~, maxF] = max(abs(STSignal(p.start_fr:end, tcheck)).^2);
    maxF = maxF + p.start_fr-1;

    % search for minimum before found maximum
    [ ~ , minF] = min(abs(STSignal(p.bound_min_peak:maxF, tcheck)).^2); %the trough
    minF=minF+p.bound_min_peak-1;

    % check for sufficient difference
    [~, fpeaks]=findpeaks(abs(STSignal(1:minF, tcheck)).^2);

    nDetectionCounter = nDetectionCounter + 1;

    % times are translates to seconds
    PSvalidated(nDetectionCounter).channel_name = Detections(n).channel_name;
    %ok<AGROW>
    PSvalidated(nDetectionCounter).start = Detections(n).start/p.fs;
    %ok<AGROW>
    PSvalidated(nDetectionCounter).stop = Detections(n).stop/p.fs;
    %ok<AGROW>
    PSvalidated(nDetectionCounter).peak = Detections(n).peak/p.fs;
    %ok<AGROW>
    PSvalidated(nDetectionCounter).peakHFOFrequency = maxF; %ok<AGROW>
    PSvalidated(nDetectionCounter).troughFrequency = minF; %ok<AGROW>
    if size(fpeaks)~=0;
    PSvalidated(nDetectionCounter).peakLowFrequency = fpeaks(end); %ok<AGROW>
    else
        PSvalidated(nDetectionCounter).peakLowFrequency=0;
    end
    PSvalidated(nDetectionCounter).peakAmplitude = Detections(n).peakAmplitude;
    %ok<AGROW>

end

end

end

```

```

if nDetectionCounter < 1

    % initialize struct
    PSvalidated(1).channel_name = p.channel_name;
    PSvalidated(1).start = -1;
    PSvalidated(1).stop = -1;
    PSvalidated(1).peak = -1;
    PSvalidated(1).peakHFOFrequency = 0;
    PSvalidated(1).troughFrequency = 0;
    PSvalidated(1).peakLowFrequency = 0;
    PSvalidated(1).peakAmplitude = 0;

end

end

```

```

function [st,t,f] = st(timeseries,minfreq,maxfreq,samplingrate,freqsamplingrate)
% Returns the Stockwell Transform of the timeseries.
% Code by Robert Glenn Stockwell.
% DO NOT DISTRIBUTE
% BETA TEST ONLY
% Reference is "Localization of the Complex Spectrum: The S Transform"
% from IEEE Transactions on Signal Processing, vol. 44., number 4, April 1996, pages 998-1001.
%
%-----Inputs Needed-----
%
% *****All frequencies in (cycles/(time unit))!*****
%     "timeseries" - vector of data to be transformed
%-----Optional Inputs -----
%
% "minfreq" is the minimum frequency in the ST result(Default=0)
% "maxfreq" is the maximum frequency in the ST result (Default=Nyquist)
% "samplingrate" is the time interval between samples (Default=1)
% "freqsamplingrate" is the frequency-sampling interval you desire in the ST result (Default=1)
% Passing a negative number will give the default ex. [s,t,f] = st(data,-1,-1,2,2)
%-----Outputs Returned-----
%
% st      -a complex matrix containing the Stockwell transform.
%           The rows of STOutput are the frequencies and the
%           columns are the time values ie each column is
%           the "local spectrum" for that point in time
% t       - a vector containing the sampled times
% f       - a vector containing the sampled frequencies
%-----Additional details-----
% % There are several parameters immediately below that
% the user may change. They are:
% [verbose]   if true prints out informational messages throughout the function.
% [removeedge] if true, removes a least squares fit parabola
%              and puts a 5% hanning taper on the edges of the time series.
%              This is usually a good idea.

```

```

%[analytic_signal] if the timeseries is real-valued
%                  this takes the analytic signal and STs it.
%                  This is almost always a good idea.
%[factor]         the width factor of the localizing gaussian
%                  ie, a sinusoid of period 10 seconds has a
%                  gaussian window of width factor*10 seconds.
%                  I usually use factor=1, but sometimes factor = 3
%                  to get better frequency resolution.
% Copyright (c) by Bob Stockwell

% This is the S transform wrapper that holds default values for the function.
TRUE = 1;
FALSE = 0;

```

```

verbose = FALSE;
removeedge= FALSE;
analytic_signal = FALSE;
factor = 1;END of DEFAULT PARAMETERS

```

```

%%%START OF INPUT VARIABLE CHECK
% First: make sure it is a valid time_series
%      If not, return the help message

if verbose disp(' '),end % i like a line left blank

if nargin == 0
    if verbose disp('No parameters inputted. '),end
    st_help
    t=0;;st=-1;;f=0;
    return
end

% Change to column vector
if size(timeseries,2) > size(timeseries,1)
    timeseries=timeseries';
end

% Make sure it is a 1-dimensional array
if size(timeseries,2) > 1
    error('Please enter a *vector* of data, not matrix')
    return
elseif (size(timeseries)==[1 1]) == 1
    error('Please enter a *vector* of data, not a scalar')
    return
end

% use defaults for input variables

if nargin == 1
    minfreq = 0;
    maxfreq = fix(length(timeseries)/2);
    samplingrate=1;

```

```

    freqsamplingrate=1;
elseif nargin==2
    maxfreq = fix(length(timeseries)/2);
    samplingrate=1;
    freqsamplingrate=1;
    [ minfreq,maxfreq,samplingrate,freqsamplingrate] =
check_input(minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,timeseries);
elseif nargin==3
    samplingrate=1;
    freqsamplingrate=1;
    [ minfreq,maxfreq,samplingrate,freqsamplingrate] =
check_input(minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,timeseries);
elseif nargin==4
    freqsamplingrate=1;
    [ minfreq,maxfreq,samplingrate,freqsamplingrate] =
check_input(minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,timeseries);
elseif nargin == 5
    [ minfreq,maxfreq,samplingrate,freqsamplingrate] =
check_input(minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,timeseries);
else
    if verbose disp('Error in input arguments: using defaults'),end
    minfreq = 0;
    maxfreq = fix(length(timeseries)/2);
    samplingrate=1;
    freqsamplingrate=1;
end
if verbose
    disp(sprintf('Minfreq = %d',minfreq))
    disp(sprintf('Maxfreq = %d',maxfreq))
    disp(sprintf('Sampling Rate (time domain) = %d',samplingrate))
    disp(sprintf('Sampling Rate (freq. domain) = %d',freqsamplingrate))
    disp(sprintf('The length of the timeseries is %d points',length(timeseries)))

    disp(' ')
end
%END OF INPUT VARIABLE CHECK

% If you want to "hardwire" minfreq & maxfreq & samplingrate & freqsamplingrate do it here

% calculate the sampled time and frequency values from the two sampling rates
t = (0:length(timeseries)-1)*samplingrate;
spe_nelements =ceil((maxfreq - minfreq+1)/freqsamplingrate);
f = (minfreq + [0:spe_nelements-1]*freqsamplingrate)/(samplingrate*length(timeseries));
if verbose disp(sprintf('The number of frequency voices is %d',spe_nelements)),end

% The actual S Transform function is here:
st =
strans(timeseries,minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,removeedge,analytic_signal,factor);
% this function is below, thus nicely encapsulated

%WRITE switch statement on nargout
% if 0 then plot amplitude spectrum

```

```

if nargin==0
    if verbose disp('Plotting pseudocolor image'),end
    pcolor(t,f,abs(st))
end

return

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function st =
strans(timeseries,minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,removeedge,analytic_signal,factor);
% Returns the Stockwell Transform, STOutput, of the time-series
% Code by R.G. Stockwell.
% Reference is "Localization of the Complex Spectrum: The S Transform"
% from IEEE Transactions on Signal Processing, vol. 44., number 4,
% April 1996, pages 998-1001.
%
%-----Inputs Returned-----
%       - are all taken care of in the wrapper function above
%
%-----Outputs Returned-----
%
%       ST    -a complex matrix containing the Stockwell transform.
%               The rows of STOutput are the frequencies and the
%               columns are the time values
%
%-----

% Compute the length of the data.
n=length(timeseries);
original = timeseries;
if removeedge
    if verbose disp('Removing trend with polynomial fit'),end
    ind = [0:n-1]';
    r = polyfit(ind,timeseries,2);
    fit = polyval(r,ind) ;
    timeseries = timeseries - fit;
    if verbose disp('Removing edges with 5% hanning taper'),end
    sh_len = floor(length(timeseries)/10);
    wn = hanning(sh_len);
    if(sh_len==0)
        sh_len=length(timeseries);
        wn = 1&[1:sh_len];
    end
    % make sure wn is a column vector, because timeseries is

```



```

if size(wn,2) > size(wn,1)
    wn=wn';
end

timeseries(1:floor(sh_len/2),1) = timeseries(1:floor(sh_len/2),1).*wn(1:floor(sh_len/2),1);
timeseries(length(timeseries)-floor(sh_len/2):n,1) = timeseries(length(timeseries)-
floor(sh_len/2):n,1).*wn(sh_len-floor(sh_len/2):sh_len,1);

end

% If vector is real, do the analytic signal

if analytic_signal
    if verbose disp('Calculating analytic signal (using Hilbert transform)'),end
    % this version of the hilbert transform is different than hilbert.m
    % This is correct!
    ts_spe = fft(real(timeseries));
    h = [1; 2*ones(fix((n-1)/2),1); ones(1-rem(n,2),1); zeros(fix((n-1)/2),1)];
    ts_spe(:) = ts_spe.*h(:);
    timeseries = ifft(ts_spe);
end

% Compute FFT's
tic;vector_fft=fft(timeseries);tim_est=toc;
vector_fft=[vector_fft,vector_fft];
tim_est = tim_est*ceil((maxfreq - minfreq+1)/freqsamplingrate) ;
if verbose disp(sprintf('Estimated time is %f',tim_est)),end

% Preallocate the SOutput matrix
st=zeros(ceil((maxfreq - minfreq+1)/freqsamplingrate),n);
% Compute the mean
% Compute S-transform value for 1 ... ceil(n/2+1)-1 frequency points
if verbose disp('Calculating S transform...'),end
if minfreq == 0
    st(1,:) = mean(timeseries)*(1&[1:1:n]);
else
    st(1,:)=ifft(vector_fft(minfreq+1:minfreq+n).*g_window(n,minfreq,factor));
end

%the actual calculation of the ST
% Start loop to increment the frequency point
for banana=freqsamplingrate:freqsamplingrate:(maxfreq-minfreq)

st(banana/freqsamplingrate+1,:)=ifft(vector_fft(minfreq+banana+1:minfreq+banana+n).*g_window(n,minfreq+banana,factor));
end % a fruit loop!  aaaaa ha ha ha ha ha ha ha ha ha ha
% End loop to increment the frequency point
if verbose disp('Finished Calculation'),end

%-----
function gauss=g_window(length, freq, factor)

% Function to compute the Gaussian window for
% function Stransform. g_window is used by function

```

```

% Stransform. Programmed by Eric Tittley
%
%-----Inputs Needed-----
%
%     length-the length of the Gaussian window
%
%     freq-the frequency at which to evaluate
%           the window.
%
%     factor- the window-width factor
%
%-----Outputs Returned-----
%
%     gauss-The Gaussian window
%

vector(1,:)=[0:length-1];
vector(2,:)=[-length:-1];
vector=vector.^2;
vector=vector*(-factor*2*pi^2/freq^2);
% Compute the Gaussion window
gauss=sum(exp(vector));

%-----

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ minfreq,maxfreq,samplingrate,freqsamplingrate] =
check_input(minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,timeseries)
% this checks numbers, and replaces them with defaults if invalid

% if the parameters are passed as an array, put them into the appropriate variables
s = size(minfreq);
l = max(s);
if l > 1
    if verbose disp('Array of inputs accepted. '),end
    temp=minfreq;
    minfreq = temp(1);;
    if l > 1 maxfreq = temp(2);,end;
    if l > 2 samplingrate = temp(3);,end;
    if l > 3 freqsamplingrate = temp(4);,end;
    if l > 4
        if verbose disp('Ignoring extra input parameters. '),end
    end;
end

if minfreq < 0 | minfreq > fix(length(timeseries)/2);
    minfreq = 0;
    if verbose disp('Minfreq < 0 or > Nyquist. Setting minfreq = 0. '),end
end
if maxfreq > length(timeseries)/2 | maxfreq < 0
    maxfreq = fix(length(timeseries)/2);
    if verbose disp(sprintf('Maxfreq < 0 or > Nyquist. Setting maxfreq = %d',maxfreq)),end
end
if minfreq > maxfreq

```

```

    temporary = minfreq;
    minfreq = maxfreq;
    maxfreq = temporary;
    clear temporary;
    if verbose disp('Swapping maxfreq <=> minfreq. '),end
end
if samplingrate <0
    samplingrate = abs(samplingrate);
    if verbose disp('Samplingrate <0. Setting samplingrate to its absolute value. '),end
end
if freqsamplingrate < 0 % check 'what if freqsamplingrate > maxfreq - minfreq' case
    freqsamplingrate = abs(freqsamplingrate);
    if verbose disp('Frequency Samplingrate negative, taking absolute value '),end
end

% bloody odd how you don't end a function

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function st_help
    disp(' ')
    disp('st()  HELP COMMAND')
    disp('st() returns - 1 or an error message if it fails')
    disp('USAGE::  [localspectra,timevector,freqvector] = st(timeseries)')
    disp('NOTE::  The function st() sets default parameters then calls the function
strans()')
    disp(' ')
    disp('You can call strans() directly and pass the following parameters')
    disp('**** warning! These inputs are not checked if strans() is called directly!! ****')
    disp('USAGE::  localspectra =
strans(timeseries,minfreq,maxfreq,samplingrate,freqsamplingrate,verbose,removeedge,analytic_signal,factor) ')

    disp(' ')
    disp('Default parameters (available in st.m)')
    disp('VERBOSE          - prints out informational messages throughout the function.')
    disp('REMOVEEDGE         - removes the edge with a 5% taper, and takes')
    disp('FACTOR              - the width factor of the localizing gaussian')
    disp('                    - ie, a sinusoid of period 10 seconds has a ')
    disp('                    - gaussian window of width factor*10 seconds.')
    disp('                    - I usually use factor=1, but sometimes factor = 3')
    disp('                    - to get better frequency resolution.')
    disp(' ')
    disp('Default input variables')
    disp('MINFREQ             - the lowest frequency in the ST result(Default=0)')
    disp('MAXFREQ             - the highest frequency in the ST result (Default=nyquist)')
    disp('SAMPLINGRATE         - the time interval between successive data points (Default = 1)')
    disp('FREQSAMPLINGRATE     - the number of frequencies between samples in the ST results')

% end of st_help procedure

```

ARRm Method

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Identification of the Seizure Onset Zone in iEEG
% of Patients with Refractory Epilepsy Using Autoregressive Model Residual
% Modulation
% Filename: FinalProject_ARRm.m
% Author: Allison Rogutich
% Date: 07/30/18
% Description:
% An iEEG file is loaded, channels are selected, DC offset is removed, then
% the signals are windowed. The ARRm values are then calculated every 1
% min. epoch of iEEG for each channel. The matrix of this result is
% plotted in a heat plot image, and patterns of ARRm values can be
% observed with these plots.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% import files *****
disp('Importing EEG file...');

%select an iEEG file
[filename,pathname] = uigetfile('\\\\fileserv\biomed\D Drive\Spectrum ECOG data\*.edf');
%read in iEEG file with helper function edfread()
[EEGfile, data] = edfread(strcat(pathname,filename));

% Extract EEG data *****
disp('Selecting Channels');

%get sampling frequency
fs = EEGfile.frequency(1);

%select channels
channel_labels = EEGfile.label;
[s,v] = listdlg('PromptString','Select Channels:',...
               'ListString',channel_labels);

chosen_channel_labels = EEGfile.label(s);
n_channels = length(chosen_channel_labels);

%get EEG signal
eegSignal = data(s,:);
N = length(eegSignal(1,:));

%Get time vector
t = 0:1/fs:N/fs;
t = t(1:end-1);

% remove DC offset *****
disp('Removing DC offset...');
mean_eeg = mean(eegSignal);
for i = 1:n_channels
    eegSignal(i,:) = eegSignal(i,:) - mean_eeg(i);
end
```

```

% window the EEG signal *****
disp('windowing EEG signal...');

timeDiff = 1/fs; %time between each sample
winlen = 40; %window of 40 samples,
overlap = winlen/2; %50% overlap

windowed_eegSignal = zeros(winlen,N/overlap,n_channels);
for i = 1:n_channels
    k = 1;
    for j = 1:overlap:N-winlen
        windowed_eegSignal(:,k,i) = eegSignal(i,j:(j+winlen)-1);
        k = k+1;
    end
end

n_windows = length(windowed_eegSignal);

% get ARrm for all channels selected

nOrder = 3; %AR model order
n_epoch = ceil(n_windows/((fs*60)/winlen)); %number of 1 min epochs in file length

% ARrm, 3rd order AR model
% each row corresponds to channels selected in order of the list
ARRmVals = zeros(n_epoch,n_channels);
order3_ARmodel = zeros(n_channels,N);

% calculate ARrm value over each epoch for each channel
for i = 1:n_channels
    disp(strcat('**** Channel ',chosen_channel_labels(i),' ****'));
    [ARRmVals(:,i),order3_ARmodel(i,:)] =
    getARRm_revised_fast(windowed_eegSignal(:, :, i),nOrder,n_windows,N,winlen,overlap,t,fs);
end

% rotate matrix to have Channels on yaxis time on xaxis
ARRmVals_rotate = ARRmVals';

% Heat plot of ARrm values over time
figure;
h = imagesc(ARRmVals_rotate);
title({'ARRm in ECoG Channels per 1min Epoch',''});
colorbar();
colormap('hot');
caxis([0,60]);
xticks(1:20:n_epoch);
xticklabels(num2cell((0:20:n_epoch)));
yticks(1:n_channels);
yticklabels(chosen_channel_labels(1:n_channels));

%Save all ARrm info in struct
foldername = uigetdir();

```

```

save(strcat(foldername,'\ ',strcat(strtok(filename,'_')),'_ARRm_all.mat'),...

'eegSignal','order3_ARmodel','windowed_eegSignal','ARRmVals_rotate','chosen_channel_labels',...
'-v7.3','-nocompression');

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: get ARRm function
% Filename: getARRm_revised_fast.m
% Author: Allison Rogutich
% Date: 07/30/18
% Description:
% Inputs:
%   selected_channel_wEEG - windowed iEEG signal for 1 channel
%   nOrder - number of AR model orders to calculate
%   n_windows - number of windows
%   N - length of iEEG signal
%   winlen - window length in samples
%   overlap - window overlap in samples
%   t - time vector of eeg signal
%   fs - sampling frequency
% Outputs:
%   ARRm - vector of ARRm values per 1 min epoch for this channel
%   order3_ARmodel - 3rd order AR model for this channel
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ARRm,order3_ARmodel] =
getARRm_revised(selected_channel_wEEG,nOrder,n_windows,N,winlen,overlap,t,fs)
% get A values for each order *****

disp('Calculating AR coefficients per window...');
[A] = get_a_values_ar_fast(nOrder,n_windows,selected_channel_wEEG);

% get AR models, plot vs. original signal *****

AR_models = zeros(1,N);
AR_residuals = zeros(winlen,n_windows);

disp('Calculating AR models per window...');
for n = 1:n_windows

    %set the size of the windowed AR model vector
    if n>1
        AR_model_N = zeros(1,overlap);
    else
        AR_model_N = zeros(1,winlen);
    end

    %pad the beginning of the AR model with first nOrder original iEEG
    %signal values. These values cannot be predicted.

```

```

pad = selected_channel_wEEG(1:nOrder,n);

%initialize vector of AR model
AR_model = zeros(1,winlen);
AR_model(1:nOrder) = pad;

%Predict every point for the AR model starting with the first value
%after nOrder original values
for j = nOrder+1:winlen
    for k = 1:nOrder
        a = A(k,n);
        AR_model(j) = AR_model(j) + a*selected_channel_wEEG(j-k,n);
    end
end

%      plotting model and residuals
%      tw = 1:40;
%      figure;
%      subplot(2,1,1);
%      plot(tw,AR_model,'b',tw, selected_channel_wEEG(:,n),'k');
%      title('ARRM Modeled Signal and Original iEEG');
%      xlabel('time (s)');
%      subplot(2,1,2);
%      zeroLine = zeros(1,40);
%      plot(tw,zeroLine,'k--',tw,selected_channel_wEEG(:,n)-AR_model,'r');
%      title('Residuals');
%      xlabel('time (s)');

%save AR model in AR model vector for this window
if n>1
    AR_model_N(1,:) = AR_model(overlap+1:end);

else
    AR_model_N(1,:) = AR_model;
end

% calculate the residuals in this window
AR_residuals(:,n) = selected_channel_wEEG(:,n) - AR_model';

% save modeled AR signal from AR vector. 50% overlap after first window
if n>1
    index = (n*overlap)+1;
    AR_models(1,index:index+overlap-1) = AR_model_N;
else
    AR_models(1,1:winlen) = AR_model_N;
end
end

%save the 3rd order AR model of the iEEG
order3_ARmodel = AR_models(1,1:end-overlap);

% Calculate Residual Signal Variances *****

```

```

disp('Calculating Residual Signal Variances and D values ...');

var_r = zeros(1,n_windows); %vector for variances of window residuals
%calculate variance of AR residuals
for j = 1:n_windows
    var_r(1,j) = var(AR_residuals(:,j));
end

% ARRM value *****

disp('Calculating ARRM ...');

%number of 1 min epochs in iEEG time length
n_epoch = ceil(n_windows/((fs*60)/winlen));

%initialize channel ARRM results vector
ARRm = zeros(n_epoch,1);

%number of samples in 1 min epoch
epoch_len = (fs*60)/40;

%change any NaN values to 0 in in the r matrix
var_r(isnan(var_r)) = 0;

%calculate ARRM value (coefficient of variation)
index = 1;
for i = 1:epoch_len:n_windows-epoch_len
    ARRM(index,1) = std(var_r(1,i:i+epoch_len-1))/mean(var_r(1,i+epoch_len-1));
    index = index+1;
end

end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Get a coefficients for AR model
% Filename: get_a_values_ar_fast.m
% Author: Allison Rogutich
% Date: 07/30/18
% Description: This function determined the a coefficients used for the AR
% models using yule walker equations.
% Inputs:
%   nOrder - number of AR orders
%   n_windows - number of windows in windowed iEEG
%   windowed_eegSignal - windowed iEEG signal
% Outputs:
%   a - a values in matrix form for each order
%   i - vector of starting indices for a values for each order in a matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

function [a] = get_a_values_ar_fast(nOrder,n_windows,windowed_eegSignal)

%initialize vector for A coefficients
a = zeros(nOrder,n_windows);

%for each window of iEEG data, calculate the A coefficients
for k = 1:n_windows
    %coefficients of EEG signal window from Yule-Walker equations
    coeffs = aryule(windowed_eegSignal(:,k)',nOrder);
    %save coefficients for this window
    a(:,k) = -coeffs(2:end)';
end

end

```

ARRm/HFO Analysis Tool

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: ARRm/HFO Analysis Review Tool
% Filename: HFO_ARRm_review_tool_fast.m
% Author: Allison Rogutich
% Date: 07/30/18
% Description:
%   An ARRm analysis struct file and an HFO analysis struct file is loaded
%   into the workspace. Then a GUI application is launched to examine the
%   results of each analysis method.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% load files *****

%load ARRm file
[filename_ARRm,pathname_ARRm] = uigetfile('*.mat','Select an ARRm file ...');
ARRmstruct = load(strcat(pathname_ARRm,filename_ARRm));

%load HFO file
[filename_HFO,pathname_HFO] = uigetfile('*.rhfe','Select HFO file ...');
HFOstruct = load(strcat(pathname_HFO,filename_HFO),'-mat');

% Get Results and info of both methods *****

%get original iEEG signal and 3rd order model
eegSignal = ARRmstruct.eegSignal;
order3_ARmodel = ARRmstruct.order3_ARmodel;

%get channel labels
chosen_channel_labels = ARRmstruct.chosen_channel_labels;
fields = chosen_channel_labels';

%get sampling frequency
fs = HFOstruct.st_FileData.v_SampleRate(1);

% get HFO Counts over time*****
N = length(eegSignal(1,:));

t = 0:1/fs:N/fs;
t = t(1:end-1);

tseg = round(N/fs)*fs;
winlength = 60*fs;
hop = 0.5*60*fs;
numwin = floor(tseg/hop)+1-ceil(winlength/hop);
HFO_Channel_Time_Matrix = zeros(size(chosen_channel_labels,1),numwin);

for i = 1: numel(fields)

    if isfield(HFOstruct,fields{i})
        %get interval and event limits
```

```

chStruct = HFOstruct.(fields{i});

%get HFO event interval limits
temp = chStruct.st_HFOInfo.m_IntervLims;
hfo_intervals.(fields{i}) = temp;

%get HFO event limits
temp = chStruct.st_HFOInfo.m_EvtLims;
hfo_events.(fields{i}) = temp;

temp = hfo_events.(fields{i});

if size(temp,1) == 1 && temp(1) <= 0 || temp(2) <= 0
    hfo_times.(fields{i}) = '';
else

    temp = int32(temp(:,1));

    %get hours minutes seconds
    strs = cell(length(temp),1);
    for j = 1:length(temp)
        index = temp(j);
        secs = t(index);

        hours = floor(secs/3600);
        next = mod(secs,3600);

        minutes = floor(next/60);
        next = mod(next,60);

        seconds = floor(next);

        strs{j} = sprintf('%02d:%02d:%02d',hours,minutes,seconds);
    end
    hfo_times.(fields{i}) = strs;
end
elseif ismember(chosen_channel_labels,fields{i})
    hfo_events.(fields{i}) = [0,0];
    hfo_intervals.(fields{i}) = [0,0];
    hfo_times.(fields{i}) = '';
end
disp(i);
end
% *****
for i = 1:numel(fields)

    if isfield(HFOstruct,fields{i})
        chStruct = HFOstruct.(fields{i});

        %get HFO heat plot
        HFO_Total_Matrix = zeros(length(chosen_channel_labels),1);

        d = fields{i};

```

```

idx = find(ismember(chosen_channel_labels,d),1,'first');

HFO_windows = hfo_events.(fields{i});

HFO_Total_Matrix(i) = size(HFO_windows,1);

for n = 1:numwin
    nstart = round(1+(n-1)*hop);
    nstop = round(nstart+winlength);
    num_events = find(HFO_windows(:,1)>=nstart & HFO_windows(:,1)<=nstop);
    HFO_Channel_Time_Matrix(idx,n) = size(HFO_windows(num_events,:),1);
end
else
    idx = find(ismember(chosen_channel_labels,fields{i}),1,'first');
    HFO_Channel_Time_Matrix(idx,:) = 0;
end

disp(i);

end

% save HFO Counts over time and ARRM values over time matrices
ARRM_vals = ARRMstruct.ARRMVals_rotate;

HFO_Channel_Time_Matrix = horzcat(HFO_Channel_Time_Matrix,...
    zeros(length(chosen_channel_labels),...
    abs(size(ARRM_vals,2)-size(HFO_Channel_Time_Matrix,2))));

clear chStruct;

% Select patient and channels of interest *****

%SH-EEG BA2680
COI_patients.SH_EEG = {'ch37D','ch38D','ch39D','ch40D','ch41D','ch42D','ch34C'};

%MA-EEG MA7670
COI_patients.MA_EEG = {'51J','52J','53J','54J','55J','36F','26D'};

%ALM-016 FA7670
COI_patients.ALM_016 = {'C1','C2','C3','C4','D1','D2'};

%HSH-021 FA7671
COI_patients.HSH_021 = {'LA1','LA2','RB4','RB5','RB6','RC6','LA4','LA5','LA6','LA7','LB1','LC1'};

%ANR-024
COI_patients.ANR_024 = {'E5','E6','E7','F6','F7'};

%MAS-029
COI_patients.MAS_029 = {'LG1','LG2','LG3','LG4','RG1','RG2'};

%DJA-033
COI_patients.DJA_033 = ({'C2','C3','C4','C5','C1','C6','E2','E1','B3','B4','B5'});

```

```

%JMM-025
COI_patients.JMM_025 =
({'TB1','TB2','TB3','TA2','TA3','TA4','TC1','TC2','TC4','TC5','TD3','TD4','TD5','TD6'});

%WDH-022
COI_patients.WDH_022 = {'B2','B3','B4','D1','D3','D4','C11','C12','C18','C19','D14'};

%choose the COI labels for appropriate patient
COI = COI_patients.JMM_025;

for i = 1:length(COI)
    COI_indexes(i) = find(strcmp(chosen_channel_labels,COI(i)));
end

%get values for summary plots
medianARRm = median(ARRm_vals,2);
totalHFO = sum(HFO_Channel_Time_Matrix,2);
medianHFO = median(HFO_Channel_Time_Matrix,2);

% Launch GUI *****
%
scrollplot_MODEL_tool_fast(t,eegSignal,order3_ARmodel,chosen_channel_labels, ...
    hfo_intervals,hfo_events,hfo_times,fields,fs,...
    ARRm_vals,HFO_Channel_Time_Matrix,medianARRm,totalHFO,medianHFO,COI_indexes);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: ARRm HFO analysis Tool
% Filename: scrollplot_MODEL_tool_fast.m
% Author: Allison Rogutich
% Date: 07/30/18
% Description: This is the GUI application that provides multiple types of
% functionality to investigate the ARRm and HFO analysis results.
% Inputs:
%   t - time vector
%   x_eeg - original iEEG signal
%   x_ARmodel - 3rd order AR model
%   channel_labels - channel name labels for plotting
%   hfo_intervals - matrix of HFO event interval indexes
%   hfo_events - matrix of HFO event indexes
%   hfo_times - matrix of HFO event times
%   fields - channel labels for accessing HFO struct info
%   fs - sampling frequency
%   ARRm_vals - matrix of ARRm values over time for each channel
%   HFO_Channel_Time_Matrix - matrix of HFO counts over time for each
%       channel
%   medianARRm - vector of median ARRm values for each channel
%   totalHFO - vector of total HFO counts for each channel
%   medianHFO - vector of median HFO counts for each channel
%   COI_indexes - vector of channel of interest indexes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function scrollplot_MODEL_tool_fast(t,x_eeg,x_ARmodel,channel_labels,...
    hfo_intervals,hfo_events,hfo_times,fields,fs,...
    ARRM_vals,HFO_Channel_Time_Matrix,medianARRM,totalHFO,medianHFO,COI_indexes)

s_row = 3;
s_col = 2;
n_channels = length(channel_labels);
half_min_samples = fs*30;

wn1 = 80;
wn2 = 488;

%Plot original iEEG *****
fig = figure('units','normalized','outerposition',[0 0 1 1]);
subplot(s_row,s_col,1);
%%%%% Generate and plot data

t1 = single(hfo_intervals.(fields{1})(1,1));
t2 = single(hfo_intervals.(fields{1})(1,2));

te1 = single(hfo_events.(fields{1})(1,1));
te2 = single(hfo_events.(fields{1})(1,2));

%*****
plot(t(t1:t2),x_eeg(1,t1:t2),'k',t(t1:t2),x_ARmodel(1,t1:t2),'b',t(te1:te2),x_eeg(1,te1:te2),'r')
;

title('Original ECoG, 3rd order AR model & HFO');
xlabel('time (s)');
ylabel('Amplitude');
lim = axis();
axis([t(t1),t(t2),lim(3),lim(4)]);

%Plot HP filtered signal *****
subplot(s_row,s_col,3);
% band pass filter original EEG signal between 80 and 488Hz
filtered_eeg = filtering(x_eeg(1,t1:t2),wn1,wn2,fs);
diff = (te2-te1);
start = 1 + (te1-t1);
stop = start + diff;

plot(t(te1:te2),filtered_eeg(start:stop),'k');

title('Filtered (80-500Hz)');
xlabel('time (s)');
ylabel('Amplitude');
lim = axis();
axis([t(te1),t(te2),lim(3),lim(4)]);

%Plot S transform PSD *****
subplot(s_row,s_col,5);

```

```

%*****
[s_trans,st_t,st_f] = st(x_eeg(1,t1:t2), 0,wn2, 1/fs, 1);
xlim([0 80]);

hfo_x1 = (te1-t1);
hfo_x1 = st_t(hfo_x1);
hfo_x2 = (te2-t1);
hfo_x2 = st_t(hfo_x2);

imagesc(st_t,st_f*length(st_t)/fs,abs(s_trans).^2,zlim);
hold on;
line([hfo_x1,hfo_x1],[1,wn2], 'Color','green');
line([hfo_x2,hfo_x2],[1,wn2], 'Color','green');
hold off;
axis([st_t(1),st_t(length(st_t)),1,wn2]);
xlabel('Time [s]');
ylabel('Frequency [Hz]');
colorbar;

%Plot ARRM *****
subplot(s_row,s_col,[2 4 6]);
imagesc(ARRm_vals);
colormap('hot');
colorbar;

caxis([0 round(max(max(ARRm_vals))/2)]);
title('ARRm Values');
yticks(1:n_channels);
yticklabels(channel_labels(1:n_channels));
set(gca, 'fontsize',7);

hold on;
current_index = floor(t1/half_min_samples);
if current_index == 0
    current_index = 1;
end
plot(current_index,1,'g+', 'MarkerEdgeColor','g', 'MarkerSize',10);
hold off;

saved_ARRm_scroll = round(max(max(ARRm_vals))/2);
saved_HFO_scroll = round(max(max(HFO_Channel_Time_Matrix))/2);

if round(max(max(ARRm_vals))) > round(max(max(HFO_Channel_Time_Matrix)))
    scale_max = round(max(max(ARRm_vals)))*2;
else
    scale_max = round(max(max(HFO_Channel_Time_Matrix)))*2;
end

%%%% Creating Uicontrol

hfosel = uicontrol('Style','listbox',...
    'String',hfo_times.(fields{1}),...

```

```

'Position',[20 400 100 200],...
'Callback',@settime);

% Create pop-up menu
popup = uicontrol('Style','popup',...
    'String', channel_labels,...
    'Position', [20 600 100 50],...
    'Callback', @setmap);

color_slider = uicontrol('Style','slider',...
    'Position', [645 575 100 20],...
    'Min',0.1,'Max',...
    scale_max,'Value',...
    round(max(max(ARRm_vals))/2),...
    'Callback', @colorlim);

color_slider_txt = uicontrol('Style','text',...
    'Position',[640 600 110 20],...
    'String','Heat Plot Color Scale');

heat_toggle = uicontrol('Style','togglebutton',...
    'String','Toggle Heat Plot',...
    'Value',0,...
    'Position',[645 540 100 30],...
    'Callback',@heat_togglebutton);

plot_total_graphs = uicontrol('Style','pushbutton',...
    'Position',[20,200,100,30],...
    'String','Total/Median Plots',...
    'Callback',@plot_t_graphs);

plot_median_graphs = uicontrol('Style','pushbutton',...
    'Position',[20,150,100,30],...
    'String','Median/Median Plots',...
    'Callback',@plot_m_graphs);

plot_median_graphs = uicontrol('Style','pushbutton',...
    'Position',[20,100,100,30],...
    'String','ARRm Residual',...
    'Callback',@plot_arm_residual);

plot_stats = uicontrol('Style','pushbutton',...
    'Position',[20,300,100,30],...
    'String','Plot Stats',...
    'Callback',@plot_stat_graphs);

plot_scatter = uicontrol('Style','pushbutton',...
    'Position',[20,250,100,30],...
    'String','Scatterplot',...
    'Callback',@plot_scatter_graphs);

plot_arm_hfo_time = uicontrol('Style','pushbutton',...
    'Position',[20,350,100,30],...

```



```

    'String','Time Plot',...
    'Callback',@plot_arm_hfo_time_graphs);

value_panel = uipanel('Title','Current values','FontSize',8,...
    'BackgroundColor','white',...
    'Units','Pixels',...
    'Position',[645,430,100,100],...
    'Parent',fig);

txtInfo =
sprintf('ARRM:\n%f\n#HFOs:\n%d',ARRM_vals(current_index,1),HFO_Channel_Time_Matrix(current_index,
1));
value_panel_text = uicontrol('Style','text',...
    'Parent',value_panel,...
    'String',txtInfo,...
    'Position',[8,5,80,80]);

function settime(source,event)
    val = source.Value;
    c_index = get(popup,'value');

    %Plot original iEEG *****

    t1 = single(hfo_intervals.(fields{c_index}))(val,1));
    t2 = single(hfo_intervals.(fields{c_index}))(val,2));

    te1 = single(hfo_events.(fields{c_index}))(val,1));
    te2 = single(hfo_events.(fields{c_index}))(val,2));

    subplot(s_row,s_col,1);

    %*****

plot(t(t1:t2),x_eeg(c_index,t1:t2),'k',t(t1:t2),x_ARmodel(c_index,t1:t2),'b',t(te1:te2),x_eeg(c_i
ndex,te1:te2),'r');
    title('Original ECoG, 3rd order AR model & HFO');
    xlabel('time (s)');
    ylabel('Amplitude');
    lim = axis();
    axis([t(t1),t(t2),lim(3),lim(4)]);

    %Plot HP filtered signal *****

    subplot(s_row,s_col,3);
    % band pass filter original EEG signal between 80 and 488Hz

    %*****

    filtered_eeg = filtering(x_eeg(1,t1:t2),wn1,wn2,fs);

    diff = (te2-te1);
    start = 1 + (te1-t1);
    stop = start + diff;

    plot(t(te1:te2),filtered_eeg(start:stop),'k');

```

```

title('Filtered (80-500Hz)');
xlabel('time (s)');
ylabel('Amplitude');
lim = axis();
axis([t(te1),t(te2),lim(3),lim(4)]);

%Plot S transform PSD *****
subplot(s_row,s_col,5);

%*****
[s_trans,st_t,st_f] = st(x_eeg(c_index,t1:t2), 0, wn2, 1/fs, 1);
zlim([0 80]);

hfo_x1 = (te1-t1);
hfo_x1 = st_t(hfo_x1);
hfo_x2 = (te2-t1);
hfo_x2 = st_t(hfo_x2);

imagesc(st_t,st_f*length(st_t)/fs,abs(s_trans).^2,zlim);
hold on;
line([hfo_x1,hfo_x1],[1,wn2],'color','green');
line([hfo_x2,hfo_x2],[1,wn2],'color','green');
hold off;
axis([st_t(1),st_t(length(st_t)),1,wn2]);
xlabel('Time [s]');
ylabel('Frequency[Hz]');
colorbar;

%Plot ARRM *****
colormax = get(color_slider,'value');
buttonVal = heat_toggle.Value;

if buttonVal == 1
    subplot(s_row,s_col,[2 4 6]);
    imagesc(HFO_Channel_Time_Matrix);
    colormap('hot');
    colorbar;
    title('HFO Rate');
    caxis([0 colormax]);
else
    subplot(s_row,s_col,[2 4 6]);
    imagesc(ARRM_vals);
    colormap('hot');
    colorbar;
    caxis([0 colormax]);
    title('ARRM values');
end
yticks(1:n_channels);
yticklabels(channel_labels(1:n_channels));
set(gca,'fontsize',7);

hold on;
current_index = floor(t1/half_min_samples);

```

```

if current_index == 0
    current_index = 1;
end
plot(current_index,c_index,'g+', 'MarkerEdgeColor','g','MarkerSize',10);
hold off;

txtInfo =
sprintf('ARRm:\n%f\n#HFOs:\n%d',ARRm_vals(c_index,current_index),HFO_Channel_Time_Matrix(c_index,
current_index));
set(value_panel_text, 'String',txtInfo);
end

function setmap(source,event)
    val = source.Value;

    set(hfose1, 'String',hfo_times.(fields{val}));

    set(hfose1, 'value',1);

    %Plot original iEEG *****
    t1 = single(hfo_intervals.(fields{val}))(1,1));
    t2 = single(hfo_intervals.(fields{val}))(1,2));

    te1 = single(hfo_events.(fields{val}))(1,1));
    te2 = single(hfo_events.(fields{val}))(1,2));

    subplot(s_row,s_col,1);

    %*****

plot(t(t1:t2),x_eeg(val,t1:t2), 'k',t(t1:t2),x_ARmodel(val,t1:t2), 'b',t(te1:te2),x_eeg(val,te1:te2
), 'r');

    title('Original ECoG, 3rd order AR model & HFO');
    xlabel('time (s)');
    ylabel('Amplitude');
    lim = axis();
    axis([t(t1),t(t2),lim(3),lim(4)]);

    %Plot HP filtered signal *****
    subplot(s_row,s_col,3);
    % band pass filter original EEG signal between 80 and 488Hz

    %*****
    filtered_eeg = filtering(x_eeg(1,t1:t2),wn1,wn2,fs);
    diff = (te2-te1);
    start = 1 + (te1-t1);
    stop = start + diff;

    plot(t(te1:te2), filtered_eeg(start:stop), 'k');
    title('Filtered (80-500Hz)');
    xlabel('time (s)');
    ylabel('Amplitude');
    lim = axis();

```

```

axis([t(te1),t(te2),lim(3),lim(4)]);

%Plot S transform PSD *****
subplot(s_row,s_col,5);

%*****
[s_trans,st_t,st_f] = st(x_eeg(val,t1:t2), 80, wn2, 1/fs, 1);
xlim([0 80]);
hfo_x1 = (te1-t1);
hfo_x1 = st_t(hfo_x1);
hfo_x2 = (te2-t1);
hfo_x2 = st_t(hfo_x2);

imagesc(st_t,st_f*length(st_t)/fs,abs(s_trans).^2,zlim);
hold on;
line([hfo_x1,hfo_x1],[1,wn2],'color','green');
line([hfo_x2,hfo_x2],[1,wn2],'color','green');
hold off;
axis([st_t(1),st_t(length(st_t)),1,wn2]);
xlabel('Time [s]');
ylabel('Frequency[Hz]');
colorbar;

%Plot ARRM *****
colormax = get(color_slider,'value');
buttonVal = heat_toggle.Value;

if buttonVal == 1
    subplot(s_row,s_col,[2 4 6]);
    imagesc(HFO_Channel_Time_Matrix);
    colormap('hot');
    colorbar;
    title('HFO Rate');
    caxis([0 colormax]);
else
    subplot(s_row,s_col,[2 4 6]);
    imagesc(ARRM_vals);
    colormap('hot');
    colorbar;
    caxis([0 colormax]);
    title('ARRM Values');
end
yticks(1:n_channels);
yticklabels(channel_labels(1:n_channels));
set(gca,'fontsize',7);

hold on;
current_index = floor(t1/half_min_samples);
if current_index == 0
    current_index = 1;
end
plot(current_index,val,'g+','MarkerEdgeColor','g','MarkerSize',10);
hold off;

```

```

        txtInfo =
sprintf('ARRm:\n%f\n#HFOs:\n%d',ARRm_vals(val,current_index),HFO_Channel_Time_Matrix(val,current_
index));
        set(value_panel_text,'String',txtInfo);

    end

function colorlim(source,event)
    buttonVal = heat_toggle.Value;
    colormax = source.Value;

    if buttonVal == 1
        subplot(s_row,s_col,[2 4 6]);
        imagesc(HFO_Channel_Time_Matrix);
        colormap('hot');
        colorbar;
        title('HFO Rate');
        caxis([0 colormax]);
        saved_HFO_scroll = colormax;
    else
        subplot(s_row,s_col,[2 4 6]);
        imagesc(ARRm_vals);
        colormap('hot');
        colorbar;
        caxis([0 colormax]);
        title('ARRm Values');
        saved_ARRm_scroll = colormax;
    end
    yticks(1:n_channels);
    yticklabels(channel_labels(1:n_channels));
    set(gca,'fontsize',7);

    hold on;
    c_index = get(popup,'value');
    t_index = get(hfosl,'value');

    t1 = hfo_intervals.(fields{c_index})(t_index,1);
    current_index = floor(t1/half_min_samples);
    if current_index == 0
        current_index = 1;
    end
    plot(current_index,c_index,'g+','MarkerEdgeColor','g','MarkerSize',10);
    hold off;

end
function heat_togglebutton(source,event)
    buttonVal = source.Value;
    c_index = get(popup,'value');
    t_index = get(hfosl,'value');
    t1 = hfo_intervals.(fields{c_index})(t_index,1);

    current_index = floor(t1/half_min_samples);

```

```

if current_index == 0
    current_index = 1;
end

if buttonVal == 1
    set(color_slider, 'Max', max(max(HFO_Channel_Time_Matrix))*2);
    set(color_slider, 'Value', saved_HFO_scroll);

    subplot(s_row, s_col, [2 4 6]);
    imagesc(HFO_Channel_Time_Matrix);
    colormap('hot');
    colorbar;
    title('HFO Rate');
    caxis([0 saved_HFO_scroll]);
else
    set(color_slider, 'Max', max(max(ARRm_vals))*2);
    set(color_slider, 'Value', saved_ARRm_scroll);

    subplot(s_row, s_col, [2 4 6]);
    imagesc(ARRm_vals);
    colormap('hot');
    colorbar;
    caxis([0 saved_ARRm_scroll]);
    title('ARRm Values');
end
yticks(1:n_channels);
yticklabels(channel_labels(1:n_channels));
set(gca, 'fontsize', 7);

hold on;
plot(current_index, c_index, 'g+', 'MarkerEdgeColor', 'g', 'MarkerSize', 10);
hold off;

txtInfo =
sprintf('ARRm:\n%f\n#HFOs:\n%d', ARRm_vals(c_index, current_index), HFO_Channel_Time_Matrix(c_index,
current_index));
set(value_panel_text, 'String', txtInfo);
end

function plot_t_graphs(source, event)
figure;
set(gcf, 'Position', [10, 10, 1500, 700]);

subplot(2,1,1);

hold on;
stem(1:n_channels, totalHFO, 'bd-');
m = totalHFO;
x = 1:n_channels;
plot(x(COI_indexes), m(COI_indexes), 'r*');
hold off;

xticks(1:n_channels);
xticklabels(channel_labels(1:n_channels));

```

```

xtickangle(90);
title('Total HFO Count');
xlabel('Channel Name');
ylabel('HFO Count');

subplot(2,1,2);

hold on;
stem(1:n_channels, medianARRm, 'kd-');
m = medianARRm;
x = 1:n_channels;
plot(x(COI_indexes),m(COI_indexes), 'r*');
hold off;

xticks(1:n_channels);
xticklabels(channel_labels(1:n_channels));
xtickangle(90);
title('Median ARRm Values');
xlabel('Channel Name');
ylabel('ARRm Value');

end

function plot_m_graphs(source,event)
figure;
set(gcf, 'Position', [10, 10, 1500, 700]);

subplot(2,1,1);

hold on;
stem(1:n_channels, medianHFO, 'bd-');
m = medianHFO;
x = 1:n_channels;
plot(x(COI_indexes),m(COI_indexes), 'r*');
hold off;

xticks(1:n_channels);
xticklabels(channel_labels(1:n_channels));
xtickangle(90);
title('Median HFO Count');
xlabel('Channel Name');
ylabel('HFO Count');

subplot(2,1,2);

hold on;
stem(1:n_channels, medianARRm, 'kd-');
m = medianARRm;
x = 1:n_channels;
plot(x(COI_indexes),m(COI_indexes), 'r*');
hold off;

xticks(1:n_channels);
xticklabels(channel_labels(1:n_channels));

```

```

        xtickangle(90);
        title('Median ARRM Values');
        xlabel('Channel Name');
        ylabel('ARRM Value');

    end

function plot_stat_graphs(source,event)
    figure;
    subplot(2,1,1);

    boxplot(ARRM_vals');
    title('ARRM Values');
    xticks(1:n_channels);
    xticklabels(channel_labels(1:n_channels));
    xtickangle(90);

    subplot(2,1,2);
    boxplot(HFO_Channel_Time_Matrix');
    title('HFO Values');
    xticks(1:n_channels);
    xticklabels(channel_labels(1:n_channels));
    xtickangle(90);
end

function plot_scatter_graphs(source,event)
    c_index = get(popup,'value');

    figure;
    hold on;
    scatter(ARRM_vals(c_index,:),HFO_Channel_Time_Matrix(c_index,:));

    ch_name = char(channel_labels(c_index));

    if isempty(find(COI_indexes == c_index)) == 0
        title(['Scatterplot #HFOs vs. ARRM Value',ch_name,'CHANNEL OF INTEREST']);
    else
        title(['Scatterplot #HFOs vs. ARRM Value',ch_name]);
    end

    hold off;
    xlabel('ARRM Values');
    ylabel('#HFOs');
end

function plot_arm_hfo_time_graphs(source,event)
    c_index = get(popup,'value');
    t_time = 1:size(ARRM_vals,2);
    ch_name = char(channel_labels(c_index));

    figure;
    subplot(2,1,1);
    plot(t_time,ARRM_vals(c_index,:));
    if isempty(find(COI_indexes == c_index)) == 0

```



```

        title(['ARRm Value Over 1 min. Epochs',ch_name,'CHANNEL OF INTEREST']);
    else
        title(['ARRm Value Over 1 min. Epochs',ch_name]);
    end
    xlabel('1 min. epoch #');
    ylabel('ARRm Value');

    subplot(2,1,2);
    plot(t_time,HFO_Channel_Time_Matrix(c_index,:));
    if isempty(find(COI_indexes == c_index)) == 0
        title(['HFO Count Over 1 min. Epochs',ch_name,'CHANNEL OF INTEREST']);
    else
        title(['HFO Count Over 1 min. Epochs',ch_name]);
    end
    xlabel('1 min. epoch #');
    ylabel('HFO Count');

end

function plot_arm_residual(source,event)

    val = get(popup,'value');
    t_val = get(hfose1,'value');
    t1 = single(hfo_intervals.(fields{val})(t_val,1));
    t2 = single(hfo_intervals.(fields{val})(t_val,2));

    te1 = single(hfo_events.(fields{val})(t_val,1));
    te2 = single(hfo_events.(fields{val})(t_val,2));

    arm_t = t(t1:t2);
    arm_r = x_eeg(val,t1:t2) - x_ARmodel(val,t1:t2)
    z_line = zeros(1,length(arm_r));

    figure;
    subplot(3,1,1);

    plot(t(t1:t2),x_eeg(val,t1:t2),'k',t(t1:t2),x_ARmodel(val,t1:t2),'b',t(te1:te2),x_eeg(val,te1:te2),
    'r');
    title('Original ECoG, 3rd order AR model & HFO');
    xlabel('time (s)');
    ylabel('Amplitude');
    lim = axis();
    axis([t(t1),t(t2),lim(3),lim(4)]);

    subplot(3,1,2);
    plot(arm_t,z_line,'g--',arm_t,arm_r,'r');
    title('ARRm Residuals for Selected Time');
    ylabel('Residual value');
    xlabel('Time (s)');
    lim = axis();
    axis([t(t1),t(t2),lim(3),lim(4)]);

    %*****

```

```

subplot(3,1,3);

%*****

[s_trans,st_t,st_f] = st(x_eeg(val,t1:t2), 80, wn2, 1/fs, 1);
zlim([0 80]);

hfo_x1 = (te1-t1);
hfo_x1 = st_t(hfo_x1);
hfo_x2 = (te2-t1);
hfo_x2 = st_t(hfo_x2);

imagesc(st_t,st_f*length(st_t)/fs,abs(s_trans).^2,zlim);
hold on;
line([hfo_x1,hfo_x1],[1,wn2],'color','green');
line([hfo_x2,hfo_x2],[1,wn2],'color','green');
hold off;
axis([st_t(1),st_t(length(st_t)),80,wn2]);
xlabel('Time [s]');
ylabel('Frequency[Hz]');
colormap hot;

end

end

```

Statistical Calculations

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Get Correlation
% Filename: getcorrelation.m
% Author: Allison Rogutich
% Date: 07/30/18
% Description:
% An ARRM Value over time figure and HFO count over time figure are
% loaded, and the data is extracted, then a linear and nonparametric
% correlation is run on each file.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% import figure files *****
%ARRm directory
[ARRmfiles,ARRmfoldername] = uigetfile('*.fig');
ARRmfiles = strcat(ARRmfoldername,ARRmfiles);

%choose HFO directory
[HFOfiles,HFOfoldername] = uigetfile('*.fig');
HFOfiles = strcat(HFOfoldername,HFOfiles);

% extract data *****

h = openfig(ARRmfiles);
h1 = openfig(HFOfiles);

ARRmData = getimage(h);
HFOData = getimage(h1);

%get channel labels next, store in chosen_channel_labels
allaxes = findall(h,'type','axes');
ylab = get(allaxes(1),'YTickLabel');
n_channels = length(ylab);

close all;

% Normalize matrices so values are from 0-1 *****
ARRmData = (ARRmData - min(min(ARRmData)))./(max(max(ARRmData))-min(min(ARRmData)));
HFOData = (HFOData - min(min(HFOData)))./(max(max(HFOData))-min(min(HFOData)));

% linear correlation *****

%Transpose ARRM and HFO matrices so that rows = observations
%and columns = variables(channels)
%channels go from column1-columnx
[rho,pval] = corr(ARRmData',HFOData','type','pearson');

%the diagonal of the rho matrix will be correlations between ARRM and HFO
%of the same channel. Likewise with the pvalues
correlations = diag(rho);
```

```

pvals = diag(pval);

%plot correlations and note significant correlations with red bars
x = 1:n_channels;
figure;
subplot(1,2,1);
hold on;
barh(x,correlations);
B = find(pvals<0.01);
barh(x(B),correlations(B),'r');
title(['Pearson Linear Correlation Value Between Channnels'],{'Significant Channels p<0.01 (red)'});
yticks(1:n_channels);
yticklabels(ylab(1:n_channels));
ylabel('Channels');
set( gca, 'YDir', 'reverse','fontsize',6 );

% rank correlation *****

[rho,pval] = corr(ARRmData',HFOData','type','spearman');

%the diagonal of the rho matrix will be correlations between ARRm and HFO
%of the same channel. Likewise with the pvalues
correlations = diag(rho);
pvals = diag(pval);

%plot correlations and note significant correlations with red bars
x = 1:n_channels;
subplot(1,2,2);
hold on;
barh(x,correlations);
B = find(pvals<0.01);
barh(x(B),correlations(B),'r');
title(['Spearman Rank Correlation Value Between Channnels'],{'Significant Channels p<0.01 (red)'});
yticks(1:n_channels);
yticklabels(ylab(1:n_channels));
ylabel('Channels');
set( gca, 'YDir', 'reverse','fontsize',6 );

```

Published with MATLAB® R2017a

Real-Time EEG Acquirement/Analysis System Software Design Document

Software Design Document Real-Time EEG Acquirement/Analysis System Author: Allison Rogutich

I. Introduction

This document outlines the software and hardware design of a real-time EEG acquiring/analyzing system. The application of this system is to serve as a proof-of-concept embedded implementation of the ARRm method in real time. Future versions of this system can be scaled up to multiple channels in a sophisticated medical-grade device with potential for use in the Epilepsy Monitoring Unit or intraoperatively.

II. Hardware Modules

An overview of the system includes a direct interface to the test subject through EEG electrodes, biosignal ADC/amplifier evaluation board, microcontroller, and display (**Figure 1**).

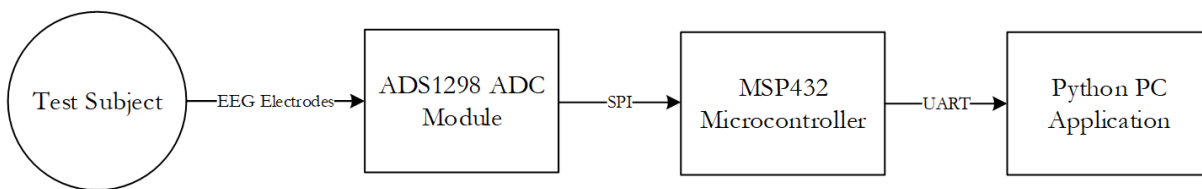


Figure 1: Hardware Block Diagram

A. EEG electrodes

1. ADINSTRUMENTS MLA2504 Shielded Lead Wires

These lead wires are shielded, which protect the delicate EEG signals from other interfering fields between the electrode site and the microcontroller. The end of the lead wire has a snap meant for use with snap-on electrodes. The pin connected to the shield should be grounded, and the other pin is connected to snap.



Figure 2: Shielded Electrode Lead Wires

The minimal setup for EEG electrodes is to have at least one active electrode, a reference electrode, and a ground (GND) electrode. The GND electrode eliminates common-mode interference. The potential voltage difference between the active electrodes and reference electrode is the EEG measurement. A common place for the reference electrode is at the vertex, or center/top of the head. Since there is no ideal reference for all EEG measurement cases, it is usually best practice to select an uninvolved and quiet electrode as reference.

2. ADINSTRUMENTS MLA1010 Disposable Electrodes

These electrodes are disposable Ag/AgCl for use with snap connect lead wires.



Figure 3: EEG Electrodes

Table 1: Pin Connections Between ADS1298 and MSP432

ADS1298	MSP432	Function
J3-7 (install JP21 2-3)	P2.0 (brown)	SPI_CS (active low)
J3-3	P2.1 (yellow)	SPI_CLK
J3-11	P2.3 (green)	SPI_MOSI
J3-13	P2.2 (blue)	SPI_MISO
J3-15	P4.5 (white)	SPI_DRDY
J3-14 (install JP22 2-3)	P4.3 (grey)	SPI_START
J3-8	P4.4 (orange)	RESET (active low)
J4-9 (connect JP24 to DVDD and TP10)	3.3V (red)	VCC_3.3V
J4-5	GND (black)	GND
J5-5 (J5 open)		PWDN (active-low)

Additional characteristics of the board relevant to biopotential measurements include analog input DC impedance of at least $1000\text{M}\Omega$, and the ability to remove 60Hz common mode interference. The analog input is fully differential, and the net signal INP-INN can span between $\pm V_{\text{REF}}/\text{Gain}$ where V_{REF} is reference voltage 2.4V. The common mode rejection ratio is shown in **Figure 5** for all gain values.

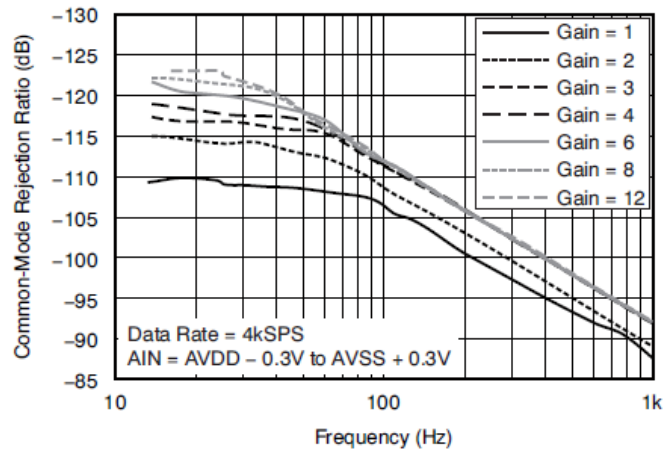


Figure 5: CMRR vs. Frequency

The ADS1298 is used in differential mode for the analog input signals (**Figure 6**).

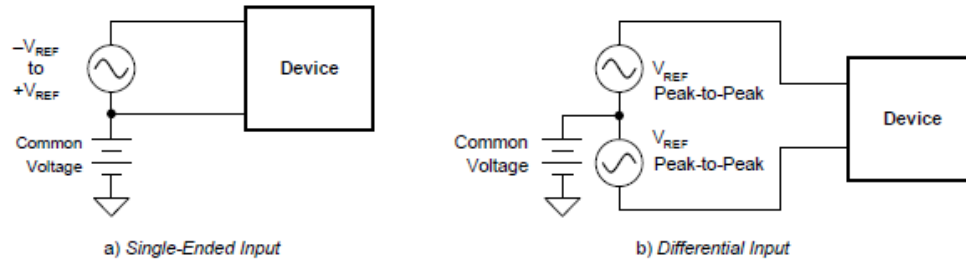
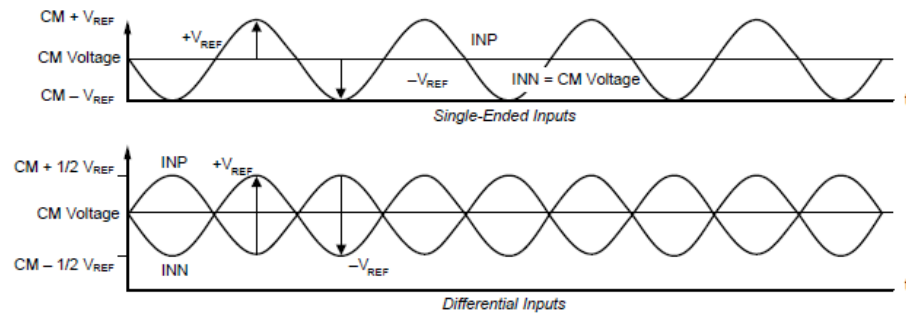


Figure 6: Methods of Driving the ADS1298

In differential mode, the signal is the difference between INP and INN. In this application, this is the difference between the active EEG electrode and the reference EEG electrode (**Figure 7**).



$$\text{Common-Mode Voltage (Differential Mode)} = \frac{(\text{INP}) + (\text{INN})}{2}, \text{ Common-Mode Voltage (Single-Ended Mode)} = \text{INN}$$

$$\text{Input Range (Differential Mode)} = (\text{AINP} - \text{AINN}) = 2 V_{\text{REF}}$$

Figure 7: Using the ADS1298 in Single-Ended or Differential Mode

For this application with 1 input channel, JP33 pins 2 and 4 were used to connect the reference and active electrodes respectively to IN1N and IN1P (**Figure 8**).

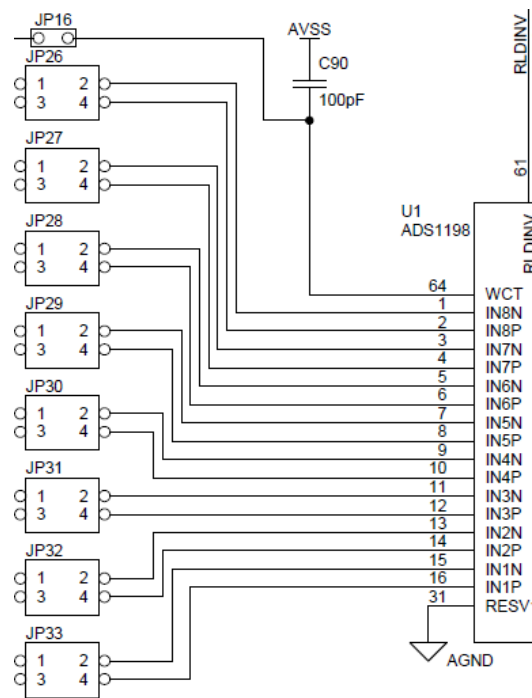


Figure 8: A/D Module Schematic for Analog Inputs

C. Microcontroller

1. MSP432P401R

The MSP432P401R LaunchPad development kit was used for this system to communicate with the ADS1298 evaluation board, process signals with the ARRM method, and output data to a display (**Figure 9**). This board also has an on-board debugger.

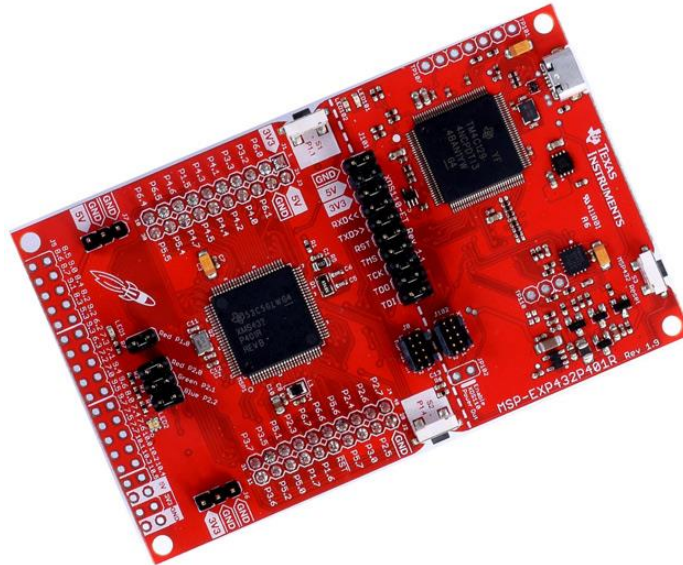


Figure 9: MSP432P401R LaunchPad Development Kit

The MSP432P401R microcontroller includes a 48MHz ARM Cortex-M4F with 256kB Flash, and 64kB RAM. This microcontroller has all features necessary for this system such as Timers, GPIOs, SPI, and UART.

D. Display

1. Serial Interface to PC Python Application

To display the signals collected from the ADC module, the microcontroller sends the data in UART messages to a Python Application on a PC. The Python Application receives data, converts the bytes into voltages, and plots the data. The calculated ARRM value is also sent to the application and is printed.

III. Software Modules

Software for the system was implemented in C++. The main software components of the system are a class that handles all communication with the ADS1298, a class that handles computation for the ARRM method, and a file that implements delay functions (**Figure 10**).

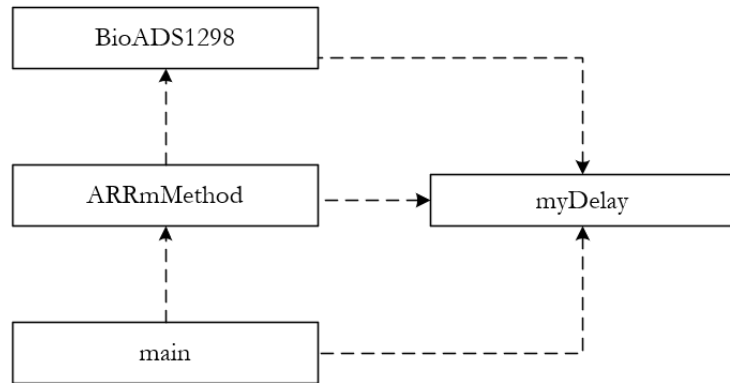


Figure 10: System Software Class Diagram

Current design of the system only collects data up to a defined number of samples based on the sampling frequency, then saves the data, the AR model, residual signal variation values, and ARRM value, then outputs that data to the PC application. The purpose of this current state of the system was to visualize and verify the data collected and analyzed by the system. Due to memory limitations, the lowest sampling frequency (250Hz) had to be used, and large amounts of data could not be saved in RAM. The main flow of the current state of the software is shown in **Figure 11**.

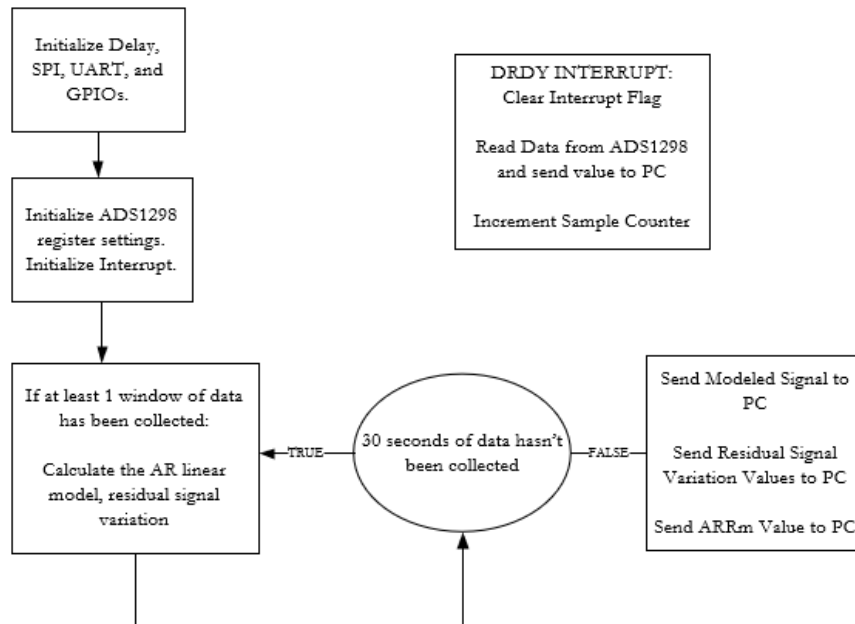


Figure 11: System Software State Diagram

A. ADS1298 Communication and Register Settings

1. Communication

Reading and writing to the ADS1298 is done via commands and opcodes. Registers can be read from/written to and also data from the analog inputs can be read with these commands (**Table 2**). The ADS1298 is by default configured for continuous data conversion (RDATAAC). The SDATAAC command must be sent before any reading or writing operations can occur.

a) Read/Write Registers

ADS1298 registers can be read from and written to using the RREG and WREG commands. Two byte opcodes must be sent for these commands. The first byte contains the command opcode and the register address. The second byte specifies the number of registers to be read-1. Since these commands are multi byte, this poses a restriction on the SCLK. A 4 tclk period must separate the end of one byte from another. Internal clock frequency is 2.048MHz, so the time to decode one instruction is 1.96μs. In this application, SCLK is set to 2MHz, so one byte is transferred in 4μs, which satisfies enough time to decode.

b) Read Data

The RDATA command is used for reading conversion data on a sample by sample basis. The START command must be issued before using the RDATA opcodes. When DRDY is low, data is ready to be read. When reading data with the RDATA command, the read operation can overlap the occurrence of the next DRDY without data corruption. In RDATAAC mode, DRDY is pulled low when data is ready to read, then the microcontroller can read data over SPI without using the RDATA command.

Table 2: ADS1298 Commands and Opcodes

Command	Description	First Byte	Second Byte
WAKEUP	Wakeup from standby mode	0x02	-
STANDBY	Enter standby mode	0x04	-
RESET	Reset the device	0x06	-
START	Start/restart (synchronize) conversions	0x08	-
STOP	Stop conversion	0x0A	-
RDATAAC	Enable Read Data Continuous mode	0x10	-
SDATAAC	Stop Read Data Continuous mode	0x11	-
RDATA	Read data by command	0x12	-
RREG	Read n nnnn registers starting at address r rrrr	001r rrrr	000n nnnn
WREG	Write n nnnn registers starting at address r rrrr	010r rrrr	000n

Reading data from the ADS1298 requires 216 clocks for each sample of data (DRDY toggle). Readings from each channel are in 24-bit Two's Complement format (Figure 12, Table 3).

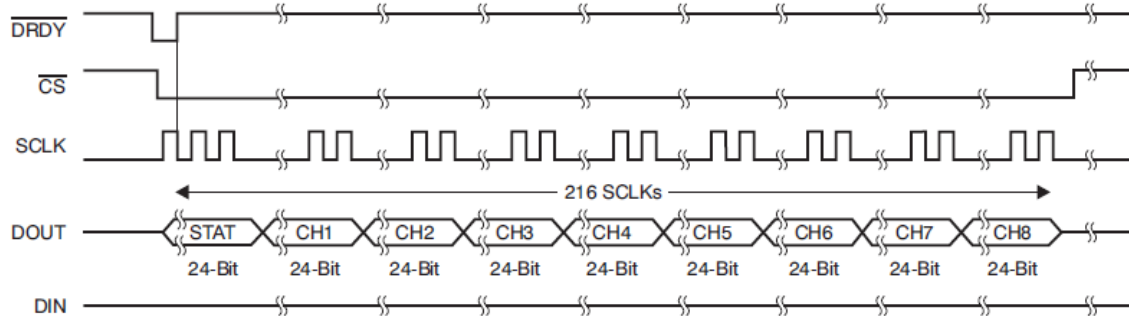


Figure 12: Timing Diagram for RDATA SPI Communication

The 24-bit data is first converted into an integer value, then the integer value is converted into the analog input signal value based on the table below, where the weight on the LSB is $V_{REF} / (2^{23} - 1)$. V_{REF} is 2.4V.

Table 3: Input Signal vs. Ideal Output Code

INPUT SIGNAL, V_{IN} ($INxP - INxN$)	IDEAL OUTPUT CODE ⁽²⁾
$\geq V_{REF}$	7FFFFFFh
$V_{REF} / (2^{23} - 1)$	000001h
0	000000h
$-V_{REF} / (2^{23} - 1)$	FFFFFFFh
$\leq -V_{REF} / (2^{23} - 1)$	800000h

2. Register Settings

To initialize the ADS1298 for this application, the initialization procedure is as follows after power-up:

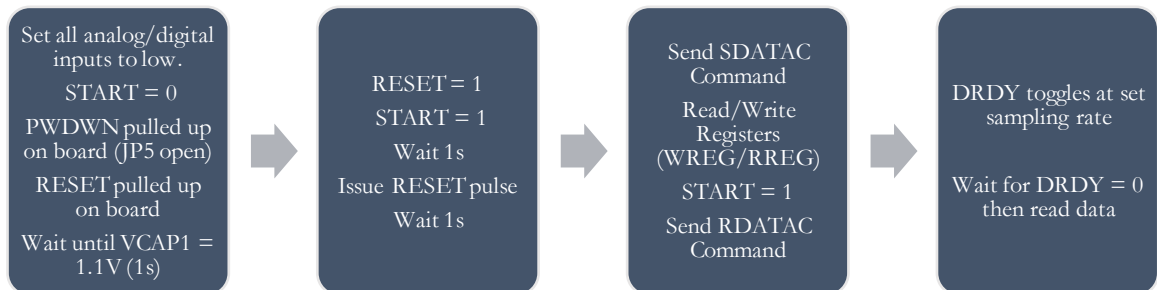


Figure 13: Software Initialization Procedure for ADS1298

The CONFIG and CHnSET registers must be initialized during the “Read/Write Registers” phase of the initialization procedure. In this step, the test signal can be set up, or the registers can be custom programmed. For this application, CONFIG1 was

programmed for Low Power mode, and an Output Data Rate of 250Hz. The CHnSET registers were also programmed for normal electrode input and a PGA gain of 6. Relevant programming options for this application or initializing the test signal are shown in **Table 4**.

Table 4: Relevant ADS1298 Register Settings

CONFIG1 Register	
Bit	Description
7	High-resolution or Low-Power mode 0 = LP mode 1 = HR mode
2:0	Output Data Rate 000 = (HR: 32kHz, LP: 16kHz) 001 = (HR: 16Hz, LP: 8Hz) 010 = (HR: 8kHz, LP: 4kHz) 011 = (HR: 4kHz, LP: 2kHz) 100 = (HR: 2kHz, LP: 1kHz) 101 = (HR: 1kHz, LP: 500Hz) 110 = (HR: 500Hz, LP: 250Hz) 111 = Reserved
CONFIG2 Register	
Bit	Description
4	Test Signal Source 0 = driven externally 1 = generated internally
2	Test Signal Amplitude 0 = $1 \times -(V_{REFP}-V_{REFN})/2400V$ 1 = $2 \times -(V_{REFP}-V_{REFN})/2400V$
1:0	Test Signal Frequency 00 = $f_{CLK}/2^{21}$ 01 = $f_{CLK}/2^{20}$ 10 = Not Used 11 = at DC
CHnSET Registers	
Bit	Description
6:4	PGA gain 000 = 6 001 = 1 010 = 2 011 = 3 100 = 4 101 = 8 110 = 12
2:0	Channel input 000 = Normal electrode input

	001 = Input shorted (for offset or noise measurements) 010 = Used in conjunction with RLD 011 = MVDD for supply measurement 100 = Temperature sensor 101 = Test Signal 110 = RLD_DRP (positive electrode is driver) 111 = RLD_DRN (negative electrode is driver)
--	--

B. BioADS1298 Class

This class includes all information necessary to initialize peripherals and communicate to the ADS1298 on the biosignal amplifier evaluation board. An interrupt is set up on the GPIO pin connected to the DRDY signal to trigger reading the data every time a conversion is completed on the ADC. Opcodes, commands, and register values are defined, and functions are implemented to carry out these commands over SPI. The number of data samples is also defined based on a definition of the sampling frequency. Private helper functions are implemented that utilize the SPI driver library to send and receive data. The most critical function in this class reads the signal on channel 1 sent from the ADS1298, sends the value to the Python PC application, converts the raw value to voltage, and stores the converted value in a buffer [1], [2].

C. ARRMMethod Class

This class has a BioADS1298 object, and uses some of its functions and variables that read/store data from the analog inputs of the ADS1298. In this class, window sample length and overlap is defined, and functions are implemented to calculate the Auto-regressive model of each window. Burg's recursive algorithm is utilized to calculate the Auto-regressive model coefficients [3]. Helper functions are implemented to compute the mean, variance, and standard deviation of vectors of data. The most critical function in this class calculates the linear model for the current window of data, calculates the residual signal variation and saves the values, then saves the model values. Based on the values saved for the residual signal variation, another function calculates the ARRM value [4], [5].

D. Delay Module

The delay module utilizes the Timer 32 on the MSP432 to create delay functions. Functions for delaying in seconds, milliseconds, and microseconds are defined [6], [7].

IV. Project Timeline and Milestones

The design of the system included sequential implementation/testing steps to ensure each level of software/hardware interface worked properly. The major system milestones are shown below in **Figure 14**.

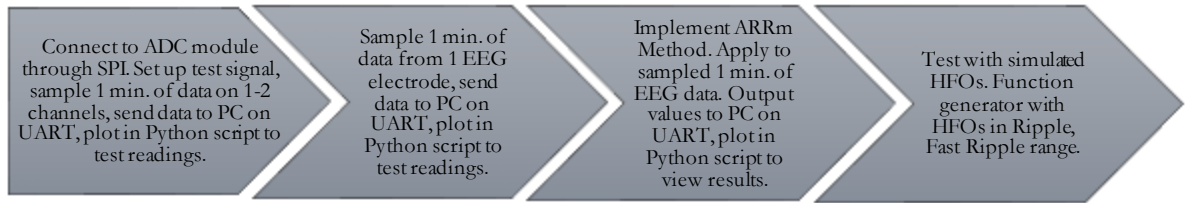


Figure 14: System Design Timeline

A. Milestone 1

The first milestone was completed with 2 channels of readings. The test signal is a square wave configured to be generated internally, amplitude of $1 \times (\text{VREFP} - \text{VREFN})/2400\text{V}$, frequency pulsed at $\text{fCLK}/2^{21}$, and gain of 6. The signals collected from channels 1 & 2, plotted with a Python script, can be observed in (Figure 15).

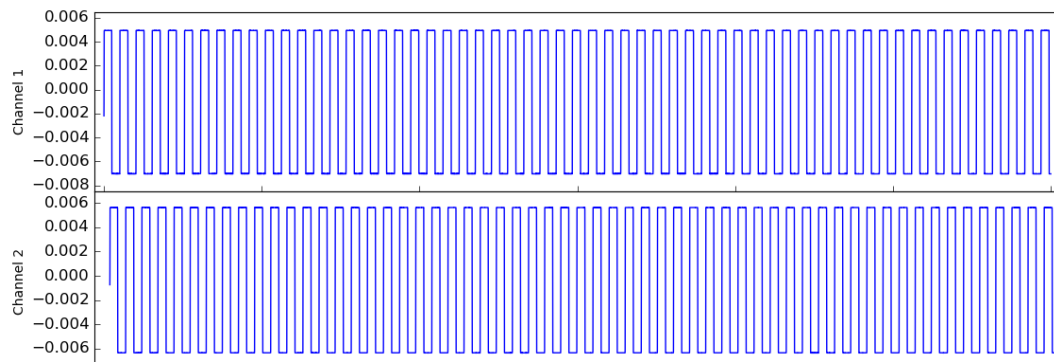


Figure 15: Test Signal Sampled from Channels 1 & 2

VREFP is 2.4V and VREFN is Analog GND, so the amplitude is expected to be $\pm 1\text{mV}$ multiplied by a gain of 6, $\pm 6\text{mV}$. The frequency should be approximately 1Hz. The plot is 60 seconds in length, and by observation, the frequency is verified to be approximately 1Hz. The values on the y-axis represent Voltage, and the peak-to-peak on both signals is $\pm 6\text{mV}$. There is an observable DC offset on channel 1.

B. Milestone 2

The test setup for EEG recording with 1 channel on the ADC module can be referenced Table 5.

Table 5: Electrode Connections

Location	Electrode	Connection
Oz	Channel 1 Active	IN1P
AFz	Reference	IN1N
Right Leg	Ground	GND

Electrode location and nomenclature was referenced from the combinatorial 10-10 system proposed by the American Electroencephalographic Society and the International Federation of Clinical Neurophysiology (**Figure 16**).

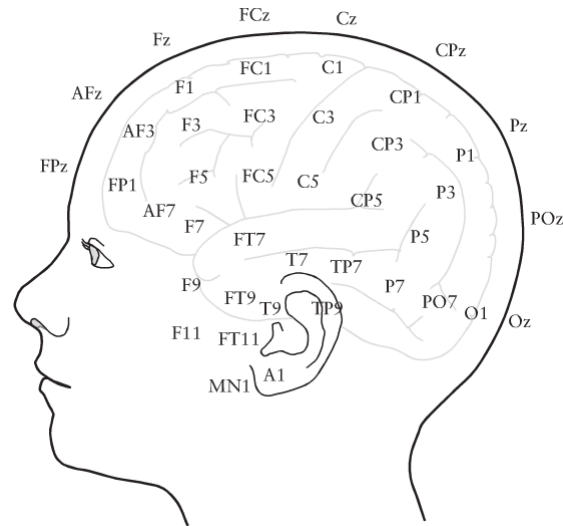


Figure 16: Electrode Positions and Nomenclature of 10-10 System

The Channel 1 Active electrode was placed at the Oz position, the Reference electrode was placed at the AFz position, and the Ground electrode was placed on the inner ankle of the right leg. The initial signal was recorded with a gain of 6 (**Figure 17, Figure 18**).

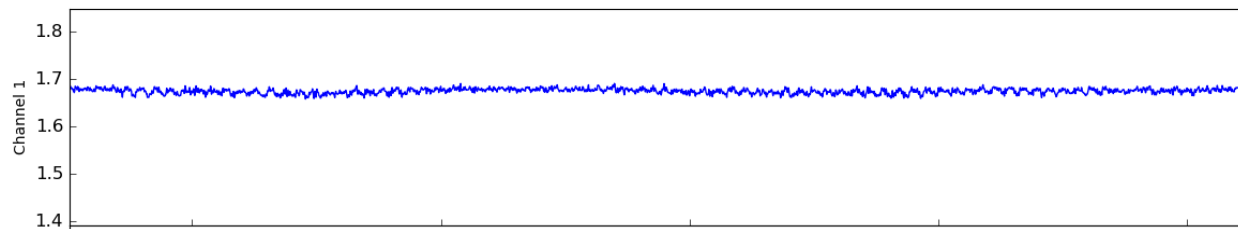


Figure 17: Initial EEG Recording from Oz (Larger Scale)

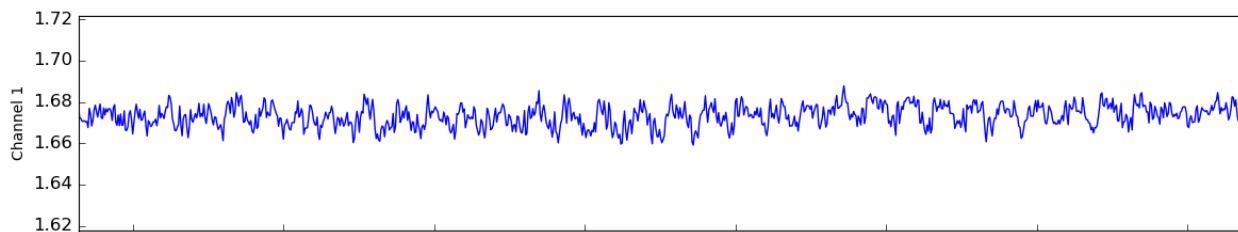


Figure 18: Initial EEG Recording from Oz (Smaller Scale)

It can be observed from these two plots that the ADS1298 can record EEG alpha waves (10Hz), but noisy interference still seems to be present in the readings.

C. Milestone 3

The first step to this milestone was testing the accuracy of the Auto-Regressive Modeling. Since the model uses up just as much RAM as the original signal, only 30 seconds of the signal was recorded. First, it was applied to the test signal, and plotted (**Figure 19, Figure 20**).

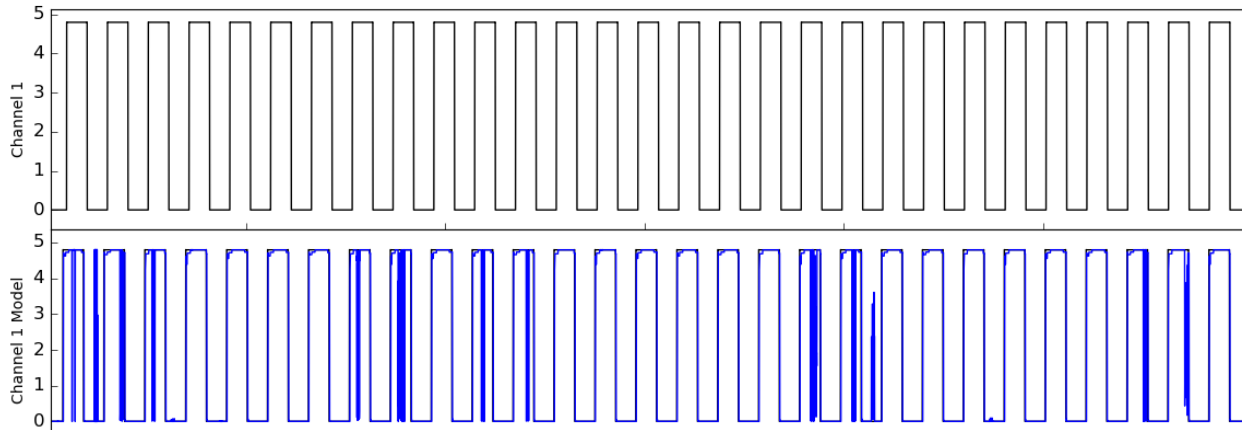


Figure 19: Test Signal and Modeled Signal (Large Scale)

It can be observed that there are some large spikes in the modeled signal from the large scale and small scale view.

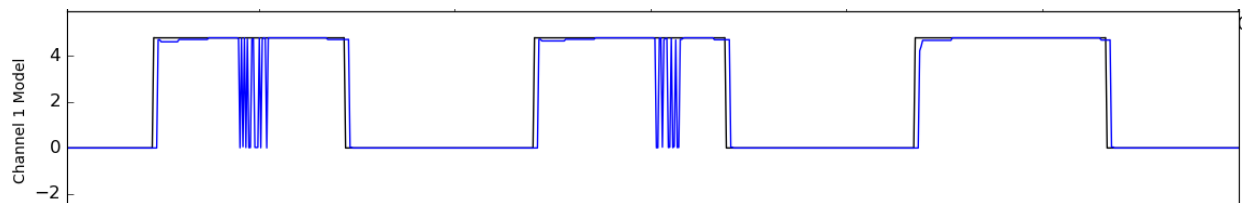


Figure 20: Test Signal and Modeled Signal (Small Scale)

It is not surprising that the AR model cannot predict a square wave very accurately, since a square wave has many sharp changes and sometimes the model window might include the signal's transition from low to high or high to low. Next, the AR modeling was applied to EEG collected from the test subject, and the model was plotted (**Figure 21**). (**Note:** at this point in development, there was an issue with two's complement conversion, which is why values are largely negative and there is some clipping)

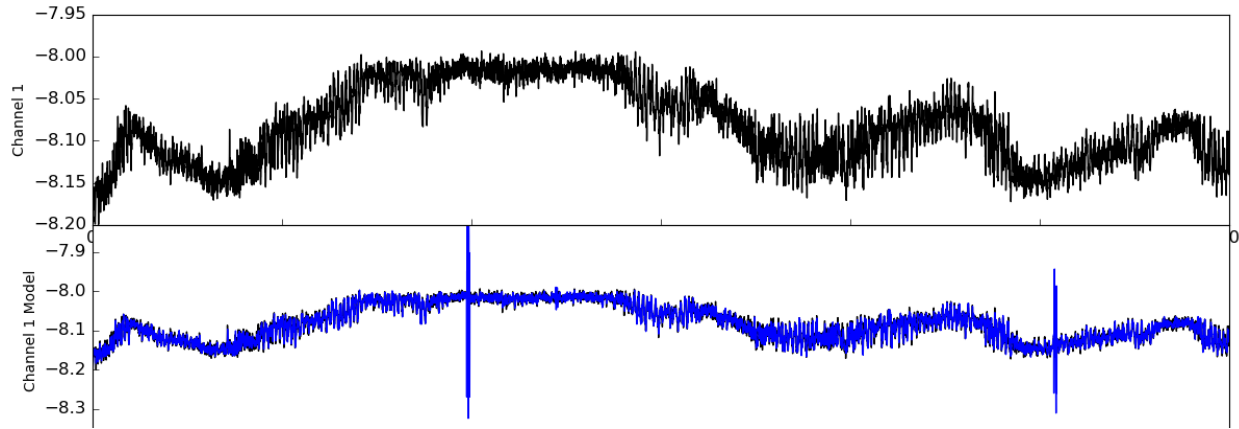


Figure 21: EEG Signal and Modeled Signal (Large Scale)

From the smaller and smallest scales zoomed in on the signal, 10Hz Alpha waves can be observed, and on the smallest scale, the modeled and original signal can be clearly differentiated (Figure 22, Figure 23).

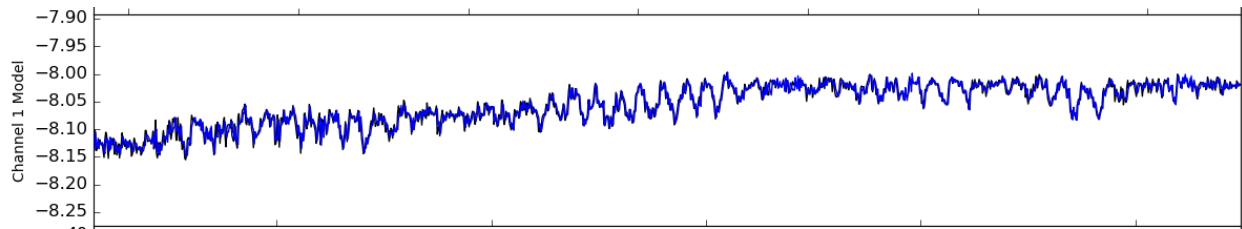


Figure 22: Modeled EEG Signal (Blue) and Original EEG Signal (Black) (Small Scale)

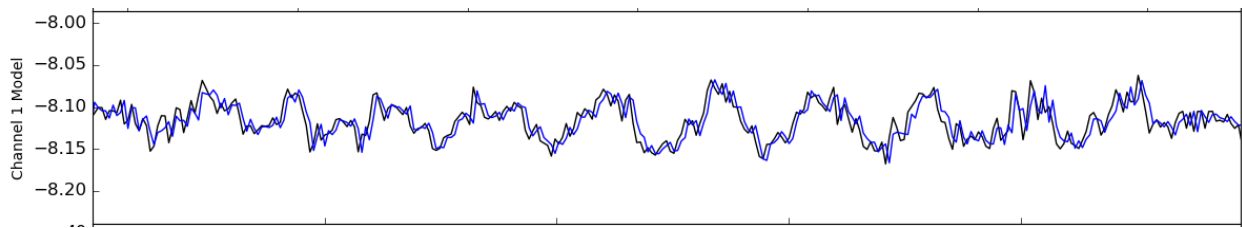


Figure 23: Modeled EEG Signal (Blue) and Original EEG Signal (Smallest Scale)

The residual signal variation was examined by recording ECG, since the QRS spikes similarly mimic epileptic spikes. The ECG signal recording and residual signal variation can be observed in Figure 24.

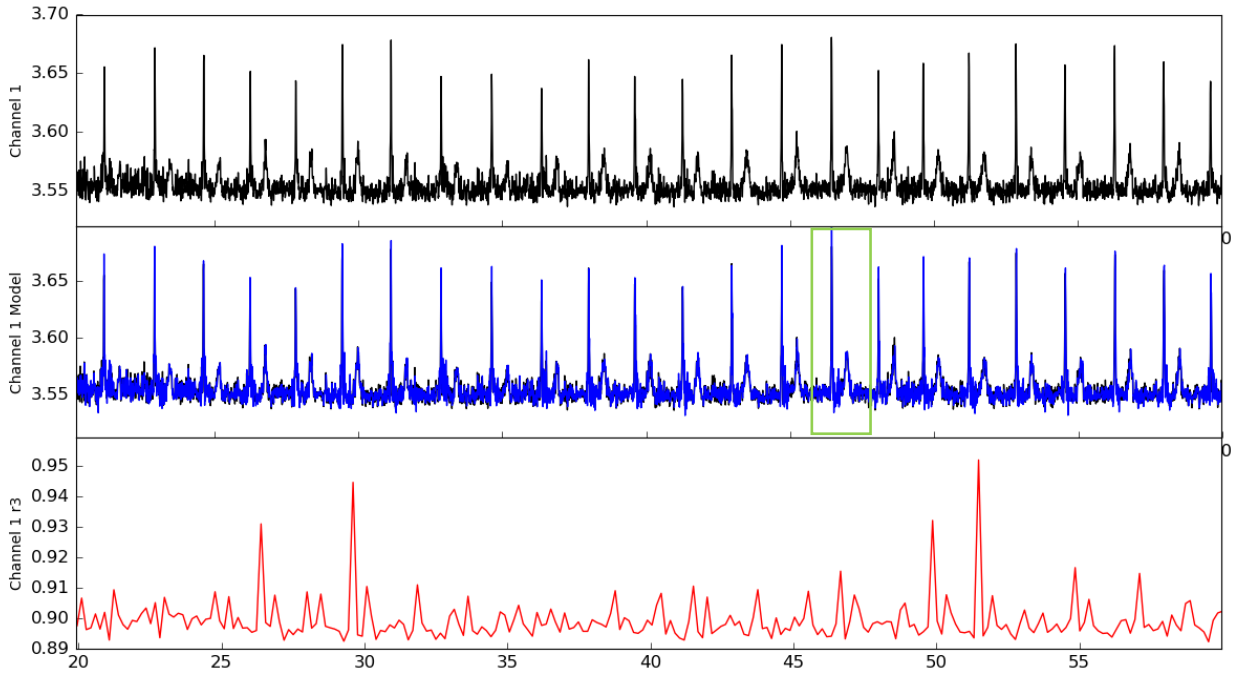


Figure 24: ECG Signal, Modeled Signal, and Residual Signal Variation

The residual signal variation tends to peak when there are spikes from the QRS complex or T wave. The green box selects one heartbeat to examine on a small scale (**Figure 25**).

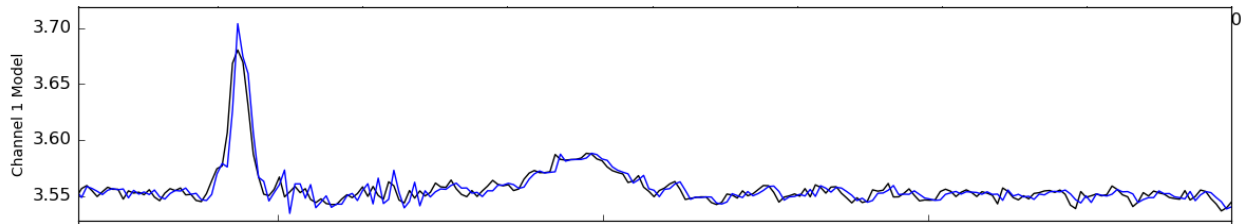


Figure 25: Original ECG Signal (black) and Modeled ECG Signal (blue) - One Heartbeat

The ARR_m value was calculated using 30 seconds of residual signal variation values:
0.00717022.

D. Milestone 4

This milestone is pending completion. For future studies on this system, a simulated EEG signal with HFOs injected should be generated using an arbitrary function generator as the signal input. Software should be modified/extended to use the proper sampling frequency (1000-2000Hz), and perform continuous collection/analysis while only outputting the ARR_m values for display over time.

References and Attachments

- [1] BioADS1298.h
- [2] BioADS1298.cpp
- [3] Collomb, Cedrick. “Burg’s Method, Algorithm and Recursion”, 2009.
- [4] ARRmMethod.h
- [5] ARRmMethod.cpp
- [6] myDelay.h
- [7] myDelay.cpp

References

- [1] M. Van't Klooster, "High Frequency Oscillations in Epilepsy Surgery," Universiteit Utrecht: Brain Center Rudolf Magnus , 2016.
- [2] Encyclopedia Britannica, Inc., The Encyclopedia Britannica Guide to The Brain, UK: Robinson, 2008.
- [3] A. D. Kaye and S. F. Davis, Principles of Neurophysiological Assessment, Mapping, and Monitoring, New York, NY: Springer, 2014.
- [4] J. W. Clark, M. R. Neuman, W. H. Olson and et. al, Medical Instrumentation: Application and Design, Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- [5] H. Blume, "Implanted EEG Electrodes," Epilepsy Foundation, 2018. [Online]. Available: <https://www.epilepsy.com/learn/professionals/diagnosis-treatment/surgery/implanted-eegelectrodes>. [Accessed 2018].
- [6] E. Wyllie, Wyllie's Treatment of Epilepsy: Principles and Practice, Philadelphia, PA: Lippincott Williams & Wilkins, 2011.
- [7] E. S. Connolly, G. M. McKhann II, J. Huang and et.al, Fundamentals of Operative Techniques in Neurosurgery, New York, NY: Thieme Medical Publishers Inc., 2002.
- [8] M. Zijlmans, P. Jiruska, R. Zelman and et. al, "High-Frequency Oscillations as a New Biomarker in Epilepsy," *Ann Neurol*, vol. 71, no. 2, pp. 169-178, 2012.
- [9] M. Z. Koubeissi and N. J. Azar, Epilepsy Board Review: A Comprehensive Guide, New York, NY: Springer, 2017.
- [10] M. Zijlmans, G. Z. Worrell, M. Dumpelmann and et.al, "How to record high-frequency oscillations in epilepsy: A practical guideline," *Epilepsia*, vol. 58, no. 8, pp. 1305-1315, 2017.
- [11] E. E. Geertsema, M. A. van't Klooster and N. E. van Klink, "Non-harmonicity in high-frequency components of the intra-operative corticogram to delineate epileptogenic tissue during surgery," *Clinical Neurophysiology*, vol. 128, no. 1, pp. 153-164, 2017.
- [12] P. Jiruska, C. Alvarado-Rojas, C. A. Schevon and et. al, "Update on the mechanisms and roles of high-frequency oscillations in seizures and epileptic disorders," *Epilepsia*, vol. 58, no. 8, pp. 1330-1339, 2017.
- [13] M. Zijlmans, G. M. Huiskamp, O. L. Cremer and et.al, "Epileptic high-frequency oscillations in intraoperative electrocorticography: The effect of propofol," *Epilepsia*, vol. 53, no. 10, pp. 1799-1809, 2012.
- [14] J. G. Jeffereys, "Nonsynaptic modulation of neuronal activity in the brain: Electric currents and extracellular ions," *Physiological Reviews*, vol. 75, no. 4, p. 689, 1995.

- [15] R. M. Helling, M. M. Koppert, G. H. Visser and et. al, "Gap Junctions as Common Cause of High-Frequency Oscillations and Epileptic Seizures in a Computational Cascade of Neuronal Mass and Compartmental Modeling," *International Journal of Neural Systems*, vol. 25, no. 6, 2015.
- [16] S. Burnos, B. Frauscher, R. Zelmann and et. al, "The morphology of high frequency oscillations (HFO) does not improve delineating the epileptogenic zone," *Clinical Neurophysiology*, vol. 126, pp. 2140-2148, 2016.
- [17] B. Frauscher, F. Bartolomei, K. Kobayashi and et.al, "High-frequency oscillations: The state of clinical research," *Epilepsia*, vol. 58, no. 8, pp. 1316-1329, 2017.
- [18] M. A. van't Klooster, N. E. van Klink, F. S. Leijten and et.al, "Residual fast ripples in the intraoperative corticogram predict epilepsy surgery outcome," *Neurology*, vol. 85, no. 2, pp. 120-128, 2015.
- [19] S. A. Hussain, G. W. Mathern, R. Sankar and J. Y. Wu, "Prospective and "live" fast ripple detection and localization in the operating room: Impact on epilepsy surgery outcomes in children," *Epilepsy Research*, vol. 127, pp. 344-351, 2016.
- [20] K. Kerber, M. Dumpelmann, B. Schelter and et. al, "Differentiation of specific ripple patterns helps to identify epileptogenic areas for surgical procedures," *Clinical Neurophysiology*, vol. 125, no. 7, pp. 1339-1345, 2014.
- [21] B. Lee, M. N. Zubair, Y. D. Marquez and et. al, "A Single-Center Experience with the NeuroPace RNS System: A Review of Techniques and Potential Problems," *World Neurosurgery*, vol. 84, no. 3, pp. 719-726, 2015.
- [22] S. Burnos, P. Hilfiker, O. Sürücü and et. al, "Human intracranial high frequency oscillations (HFOs) detected by automatic time-frequency analysis," *PLoS One*, vol. 9, no. 4, 2014.
- [23] R. Islam, "Human Intracranial High Frequency Oscillation Detection Using Time Frequency Analysis and Its Relation to the Seizure Onset Zone," *Masters Theses*, no. 780, 2015.
- [24] R. G. Stockwell, L. Mansinha and R. P. Lowe, "Localization of the Complex Spectrum: The S Transform," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 998-1001, 1996.
- [25] R. Zelmann, F. Mari, M. Zijlmans and et. al, "A comparison between detectors of high frequency oscillations," *Clinical Neurophysiology*, vol. 123, pp. 106-116, 2012.
- [26] C. G. Benar, L. Chauviere, F. Bartolomei and et.al, "Pitfalls of high-pass filtering for detecting epileptic oscillations: A technical note on "false" ripples," *Clinical Neurophysiology*, vol. 121, no. 3, pp. 301-310, 2010.
- [27] E. E. Geertsema, G. H. Visser, D. N. Velis and et. al, "Automated Seizure Onset Zone Approximation Based on Nonharmonic High-Frequency Oscillations in Human Interictal Intracranial EEGs," *Int. J. of Neural Systems*, vol. 25, no. 5, 2015.

- [28] P. Franaszczuk and K. Blinowska, "Linear model of brain electrical activity - EEG as a superposition of damped oscillatory modes," *Journal of Biomedical Cybernetics*, vol. 53, no. 1, pp. 19-25, 1985.
- [29] M. Navarrete, C. Alvarado-Rojas, M. Van Quyen and et. al, "RIPPLELAB: A Comprehensive Application for the Detection, Analysis and Classification of High Frequency Oscillations in EEG Signals," *PLoS ONE*, vol. 11, no. 6, 2016.
- [30] M. Navarrete, "RIPPLELAB's User Manual," ResearchGate, 2016.
- [31] Mathworks, "Documentation: corr," 2018. [Online]. Available: <https://www.mathworks.com/help/stats/corr.html>. [Accessed 2018].