NTNU
Norwegian University of
Science and Technology

Multivariate Analysis on Preprocessed Time-Frequency Representations of Electrode Voltage Signals from Microelectrode Array Experiments on an in-vitro Dopaminergic Neuronal Culture

Ivar Thokle Hovden

Master of Science in Cybernetics and Robotics

Submission date:	June 2018
Supervisor:	Frank Westad, ITK
Co-supervisor:	Martinius Knudsen, ITK Beatriz Galindo-Prieto, ITK Peter Aaser, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Summary

The Master's thesis presents custom developed Fourier based preprocessing methods as well as results from principal component analysis (PCA) and partial least squares projection to latent structures (PLS) regression on biological neural data from a microelectrode array (MEA) in-vitro culture.

The mixed neuronal cell culture was grown by PhD students at Department of Neuromedicine and Movement Science (INB), Faculty of Medicine and Health Sciences, NTNU and contained a specific type of dissociated human midbrain dopamine neurons known to have a selective vulnerability in Parkinson's disease (PD). The differentiation of the culture was part of PD related research.

The goal of the analysis has been to look for neuronal signaling properties preferably related to motor tasks that can be used to control a robot in the NTNU Cyborg project. 13 MEA experiments from the time span 2017-03-20 to 2018-01-22 were analyzed in the thesis. Each MEA experiment contained approximately 10 minute electrical pV recordings from 60 electrodes.

The combination of the preprocessing and PCA and PLS regression resulted in a study of spatiotemporal variation of detected action potentials (APs) across frequency components in multiple combinations of electrode signals.

PCA was used as exploratory analysis of spatiotemporal variation of detected action potentials in individual MEA experiments, while PLS together with a variable influence on projection (VIP) method was used to compare sets of two MEA experiments based on spatiotemporal variation of detected action potentials.

It was discovered that synchronized (coherent) oscillations occurring in bursts on multiple electrodes gradually develop into shorter synchronized bursts with shorter pauses as the age increases, until synchronization of APs is not apparent in the most adult culture. Age increase also leads to power increase especially in larger frequencies in the investigated frequency range $300 - 3000$ Hz.

In properly preprocessed MEA experiments of the adult culture, certain frequency components show distinct variational patterns (for example the component $2540 - 2550$ Hz (a frequency resolution of 10 Hz was used). In such a MEA experiment, the patterns are observable on most electrodes, so the observed patterns are highly independent of physical electrode location. Which components that show distinct variational patterns differ across the relevant MEA experiments.

Moreover, the lower the frequency component lays in the range $300 - 3000$ Hz, the more important it is to describe the difference between two experiments when each experiment represents a different age of the culture. The age discrimination is shown to be based on difference in electrical power in these frequency components, and certain electrodes (31, 32, 33, etc.) are more influential than others in the age discrimination.

Conversely, the higher the frequency component lays in the range $300 - 3000$ Hz, the more important it is to describe the difference between two experiments when each experiment represents the same age of the culture. Two adult culture MEA experiments were selected for this analysis. All electrodes seem to be equally influential in discrimination using these frequency components. Only frequency components of electrode 35 show some unique features in the discrimination.

This is the same electrode to have picked up the most spiking variation in many of the analyses of younger culture.

Sammendrag

Denne Masteroppgaven inneholder egenutviklede Fourier-baserte metoder for preprossesering av data til prinsipalkomponentanalyse (PCA) og minste kvadraters projeksjon til latente strukturer (PLS) regresjon. Disse metodene ble brukt på biologiske nevraldata tatt opp fra en in-vitro mikroelektrodearray (MEA) cellekultur.

Differensieringen av nervecellene var på forhånd gjort av doktorgradsstudenter fra institutt for nevromedisin og bevegelsesvitenskap (INB) på fakultet for medisin og helsevitenskap, NTNU og var en type dopaminergiske nevroner eksisterende i den menneskelige hjernen som er kjent for å være spesielt utsatt for nedbrytning i diagnosen Parkinson's (PD). Differensieringsprosedyren var del av forskning på PD.

Mål med analysen har vært å se etter mønstre i nervecellesignaler fra elektrodeopptak som gjerne er relatert til kognitive bevegelsesoppgaver og som kan brukes til å styre bevegelsene til roboten i Kyborg prosjektet på NTNU. Totalt 13 MEA-eksperimenter fra perioden 20-03-2017 til 22-01-2018 ble analysert. Hvert av disse eksperimentene var omtrentlig 10 minutter lange opptak av elektrisk pV (pikovolt) fra 60 elektroder.

Kombinert sett var preprosseseringen og multivariatanalysen en studie i spatiotemporal variasjon av detekterte aksjonspotensialer over frekvenskomponenter fra kombinasjoner av elektrodesignaler.

PCA utgjorte en eksplorativ del av analysen hvor individuelle MEA-eksperimenter ble analysert. PLS sammen med en metode for beregning av enkeltvariablers påvirkning i projeksjon (VIP) ble brukt til å sammenligne sett på to MEA-eksperimenter.

Det første vesentlige resultatet av analysen var observasjonen at samvariasjon av aksjonspotensialer (fasesynkroniserte oscillasjoner over frekvenskomponenter / koherens) ble gradvis fravikende med økning av alder på cellekulturen. Disse oscillasjonene var typisk tilstedeværende i den unge kulturen og forekomm i repetative utspring ("bursts") med pause på flere sekunder. Pausene mellom utspringene ble også kortere med alderen, frem til ingen samvariasjon av aksjonspotensialer var detektert i voksen cellekultur. Elektrisk effekt, spesielt i de høyere frekvenskomponentene i området $300 - 3000$ Hz, økte samtidig vesentlig med alderen.

Med korrekt preprossesering og analyse av voksen kultur kunne bestemte variansbaserte mønstre knyttes til frekvenskomponenter (for eksempel $2540 - 2550$ Hz (frekvensoppløsning var her 10 Hz)). Like mønstre var synlig i mange elektrodeopptak (for et gitt MEA-eksperiment), så denne oppførselen ser ut til å være uavhengig av elektrodeplassering. Hvilke frekvenskomponenter som skilte seg ut var imidlertid forskjellig for de relevante MEA-eksperimentene.

Når to MEA-eksperimenter representerer forskjellig alder på kulturen er lavere frekvenskomponenter viktigere enn høyere frekvenskomponenter (begrenset i frekvensbåndet analysert, $300 - 3000$ Hz) når de spatiotemporale forskjellene mellom eksperimentene i detekterte aktsjonspotensialer skal undersøkes. Forskjellene i disse frekvenskomponentene ble fastsatt til å være effektforskjeller som øker når aldersforskjellen øker. Bestemte elektroder var spesielt utslagsgivende i denne effektforskjellen, med andre ord viktige i å beskrive alder på kulturen (31, 31, 33 med flere).

På den annen side, dersom to MEA-eksperimenter representerer lik alder på kulturen er høyere frekvenskomponenter viktigere enn lavere frekvenskomponenter (begrenset i frekvensbåndet analysert, $300-3000$ Hz) i bestemmelse av forskjellene mellom eksperimentene. To MEA-eksperimenter av voksen cellekultur ble brukt i denne analysen. Høyere frekvenskomponenter i omtrent alle elektrodemålinger var like viktige i bestemmelse av forskjellene mellom eksperimentene. Elektrode 35 viste også en del unik oppførsel i sammenligning med de andre elektrodene. 35 var også elektroden som registrerte mest variasjon i detekterte aksjonspotensialer i ung cellekultur.

Preface

The basis of the presented works were proposals in the NTNU Cyborg project to further investigate multivariate analysis on neural data as part of a larger goal of making a closed-loop bio-mechanic robot. An overview of long-term goals and associated works in the Cyborg project is given in Chapter 1.3.

Some multivariate analysis were already conducted on neural data in specialization projects by MSc student Helge-André Langåker (Langåker, 2017) and the author of this work (Hovden, 2017), as well as in work by PhD candidate Martinius Knudsen. The author has also previously investigated the same neural data in the course TTK7 ”Adaptive Data Analysis: Theory and Applications” (Hovden and Gulbrandsen, 2017a) and in TTK19 ”Structures and Relationships in Complex Systems” (Hovden and Gulbrandsen, 2017b). Description of the previous relevant work are described in Chapter 1 and the important results are summarized in Chapter 2.5.

The analyzed data are microelectrode-array (MEA) experiments on a single culture of dopaminergic neurons. All MEA data were obtained and made available by Ioanna and Axel Sandvig´s laboratory, Department of Neuromedicine and Movement Science (INB), Faculty of Medicine and Health Sciences, NTNU, based on dopaminergic neuron networks engineered in the lab. Overview of differentiation of the neurons as well as laboratory equipment and software are given in Chapter 2.1.1 and in the Appendix Chapter 8.5. Chapter 8.5 also lists the software used for multivariate analysis (The Unscrambler®version 10.5) as well as for preprocessing and other practical tasks.

I want to thank Postdoc Beatriz Galindo-Prieto and Professor II Frank Ove Westad at Department of Engineering Cybernetics for great supervision on the multivariate analysis throughout the past semester. I also thank PhD candidate Peter Aaser at Department of Computer Science for helping me to learn the Scala functional programming language for use in real-time preprocessing. Thanks also to PhD candidate Martinius Knudsen at Department of Engineering Cybernetics for organizing collaborative meetings and Professor Gunnar Tufte at Department of Computer Science for providing interesting research goals of the Cyborg project. Thanks to Ioanna and Axel Sandvig´s laboratory, Department of Neuromedicine and Movement Science (INB), Faculty of Medicine and Health Sciences, for providing access to the data. Thanks to PhD students Vibeke Devold Valderhaug, INB, Ola Huse Ramstad, INB, and Rosanne van de Wijdeven, IKOM, for all relevant experimental work and useful discussions. Lastly, a thank to Department of Engineering Cybernetics, NTNU for providing several years of education in the highly sought-after multidisciplinary field of Engineering Cybernetics.

The original problem description for the Master’s thesis is shown in Chapter 1.1 and a status report conforming to the original description is given in Chapter 1.2.

Table of Contents

List of Tables

List of Figures

Abbreviations

PD	=	Parkinson’s disease
PCA	=	Principal Component Analysis
PC	=	Principal Component
PSD	=	Power Spectral Density
ISI	=	Interspike interval
SVD	=	Singular Value Decomposition
DFT	=	Discrete Fourier Transform
FFT	=	Fast Fourier Transform
fs2	=	Functional Streams 2 (Scala library)
STFT	=	Short-Time Fourier Transform
IIR	=	Infinite Impulse Response
FIR	=	Finite Impulse Response
SNN	=	Spiking Neural Network
in-vitro	=	”in the glass”
in-vivo	=	”within the living”
DA	=	Dopamine
DAergic	=	Dopaminergic
mDA	=	Midbrain dopaminergic
iPSC	=	Induced pluripotent stem cell
hiPSC or hiPSC	=	Human pluripotent stem cell
MEA	=	Microelectrode Array
EEG	=	Electroencephalography
MEG	=	Magnetoencephalography
ECoG	=	Electrocorticography
LFP	=	Local Field Potential
SUA	=	Single-unit activity
MUA	=	Multi-unit avtivity
EAP	=	Extracellular action potential
IAP	=	Intracellular action potential
SNc or A9	=	Substantia nigra pars compacta
SNR	=	Signal-to-noise ratio
HHT	=	Hilbert–Huang transform
IMF	=	Intrinsic mode function
EMD	=	Empirical Mode Decomposition
MEMD	=	Multivariate Empirical Mode Decomposition
EEMD	=	Ensemble Empirical Mode Decomposition
CWT	=	Continuous wavelet transform
PLS or PLSR	=	Partial Least Squares Regression
PLS-DA	=	Partial Least Squares Discriminant Analysis
MVA	=	Multivariate analysis
ANN	=	Artificial Neural Network
VIP for OPLS	=	Variable Influence on Projection (VIP) for Orthogonal Projections to Latent Structures
RT	=	real-time
RTP	=	real-time preprocessed
OP	=	offline preprocessed
JVM	=	Java Virtual Machine

Chapter 1

Introduction

Consider the following:

- Linear interpretive models from multivariate analysis (MVA).
- Living dopaminergic neuronal cell culture in Microelectrode-array (MEA) for measuring electrical activity.
- MEA culture and Robot in control feedback loop over internet.
- Digital signal processing in frequency domain.
- Concurrent hardware independent functional programming by Java virtual machines (JVMs).

The Master's thesis deals with a combination of those aspects. It is a continuation of earlier works on various analyses of biological dopaminergic neural data in project work (Hovden, 2017) and in the courses TTK7 ”Adaptive Data Analysis: Theory and Applications” (Hovden and Gulbrandsen, 2017a) and TTK19 ”Structures and Relationships in Complex Systems” (Hovden and Gulbrandsen, 2017b). The common denominator is that data from a single culture (the same physical living neural cell culture) have been analyzed, with the aim of finding distinct features in the data, although with varying quality on the data. They are various voltage recordings from experiments on a specific in-vitro culture containing dopaminergic neurons dating from a timespan over several months. The experiments were conducted by PhD Students at the Sandvig laboratory, Department of Neuromedicine and Movement Science, NTNU.

The goal of the thesis has been to look for properties in the recordings that may eventually be exposed and controlled in a feedback loop for robotic control (Aaser et al., 2017; Aaser, 2017). In order to get proper results from multivate analysis on neural data, sufficient preprocessing is crucial. Building upon different approaches in earlier works, the main preprocessing in this work outputs from a raw MEA neural recording an aggregated time-frequency representation of oscillatory activity in each electrode signal, based on Fourier analysis and PCA for extraction of noise thresholds in the frequency domain. Chapter 3 describes a simplified version of the procedure (using existing software for Fourier-based noise reduction), which works on data of an entire MEA experiment, while Chapter 4 introduces the complete online version requiring PCA-based calculation of noise thresholds, which works on streamed data from a selected electrode.

1.1 Original problem description

This section is the thematic description of the Master's thesis agreement (December 11th 2017):

At Axel and Ioanna Sandvig's lab of translative and regenerative neuroscience at Department of Neuromedicine and Movement Science at St. Olav University Hospital, neuronal cultures related to motor functions are grown and experimented on in in-vitro microelectrode arrays.

From research, we know that frequencies from electrophysiological signals in the alpha ($8 - 13$ Hz) and beta/mu ($13 - 30$ Hz) band and/or lower, can be related to imaginary and real movement read from the brain, where imaginary movement can be the signals from a-priory planning of the movement, leading to another observed signal during the actual movement (Nicolas-Alonso and Gomez-Gil, 2012).

From the project work that looked at signalling properties of dopaminergic neurons, observed lower frequencies ($2 - 13$ Hz) of discrete spiking activity were proposed to relate to the low frequencies of envelopes from a wide band of higher frequencies (center approx. 1000 Hz). Additionally, significant energy increase in the high frequencies, as well as a de-synchronization of the low envelope frequencies (decrease in their energies) was observed from four experiments on the same culture over a five month period.

Based on these results, the Master's thesis would look further on detecting such time and frequency properties of neuronal signals and analyze them for interpretation of a robot action. Reading patterns in time and frequency such as resonance and aggregate behavior from computational intelligence/soft computing and data mining insights, could perhaps be used to control the dynamics observed in the electrical signals, based on an external heuristic from a simulation environment of a robot where it moves around based on neural activity, and the objective function is to not go into walls.

The Master's work would be to lay and develop existing foundations for a closed-loop system for control of the simulated robot where MEA is part of the loop. Existing work has been done on programming the feedback system in Scala and C#.

The work requires contributions from neurobiology, cybernetics and computational science (MVA, statistics).

Practical tasks:

1. Modularize (and refactor) the distributed data processing framework (Scala, Python, C#, etc.) for easier use.
2. Add a real-time efficient implementation of the Short-Time Fourier Transform on data streams from MEA. Other possible implementations: CWT, HHT .
3. Multivariate analysis (MVA) on STFT output to reduce dimensionality.
4. Cooperate with PhD candidates at Sandvig's lab in order to make more complex stimulation and perhaps other environmental changes to the cultures. All this according to an objective function from the simulation environment.
5. Speeding up the MEA learning process using computational intelligence with further analysing sensorimotor rhythms for robot action. For example, associative memory abilities of artificial neural networks (ANNs) could help in this task on MVA preprocessed data.

6. Different approaches of MVA, including feature selection if needed, will be tested in order to further data interpretation and understanding.

1.2 Status report

Necessary biological background theory is explained in Chapter 2 the Appendix Chapter 8.3. Below is a summary of the current state of the proposed tasks (April 30th 2018)

1. By the distributed data processing framework, the author refers primarily to the robot simulator **SHODAN** (YouTube example, Github page, access required) (Aaser, 2017), and back-end digital signal processing software MEAME2 (link 1, link 2), all developed by PhD candidate Peter Aaser at Department of Computer Science, NTNU. There exist other software developed in the Cyborg project as well, for example remap which is a server written in the Golang language that converts raw electrode data from the MEA server (MEAME2, link 1 mentioned previously) into spike buckets. Being one of the co-supervisors, Peter spent significant time in helping the author learn the functional programming language Scala, the programming language used in SHODAN, for the real-time preprocessing. The real-time preprocessing, which will be explained in Chapter 4, was later implemented using a Scala library for streaming neuronal data over internet, Functional Streams 2 (fs2), in order to make it easy to implement it in the simulator. Before that, the author made substantial effort on incorporating the preprocessing in Apache Spark, which is a distributed computation framework commonly used on data science tasks (Zaharia et al., 2010, 2012). However, since the distributed nature of Spark Streaming (dstream) made it very difficult to maintain a global order (of stream processing), correct aggregation of neural timing information (the preprocessing, motivation explained in Chapter 2.6) was difficult hence leading to using fs2 that maintains a global order (on the cost of being less parallel). The author did not take part in improving the existing code in SHODAN, but made additional code for real-time preprocessing for MVA in the same language (Scala), explained in below point 2.

2. This was achieved, and Scala code and examples are presented on this Github page as real-time preprocessing. The method is described in Chapter 4. A comparable offline preprocessing method was also developed with Python and the audio editing program Audacity, and is described in Chapter 3. Python code with example can be found on this Github page. Continuous Wavelet Transform (CWT) and Hilbert-Huang Transform (HHT) are alternative methods that can construct spectrogram-like outputs that would be comparable to the spectrogram from STFT. They were not used in this thesis. CWT and HHT, especially HHT, are more computationally intensive to implement and the FFT algorhtm was preferred because of its efficient implementation. Empirical Mode Decomposition (EMD), a procedure part of HHT, was previously performed on a subset of the same data, and the a link to the TTK7 report is available in the references (Hovden and Gulbrandsen, 2017a)

3. Principal Component Analysis (PCA), the most common multivariate analysis (MVA) method was applied on the preprocessed data from point 2. The method reduces the dimensionality of a data set into latent variables called Principal Components (PCs). Multiple informative plots and metrics can reveal sample-wise and variable-vise important features in the data, and they are explained alongside with the method in Chapter 5.1 and 5.4. Projections of test samples were performed on some PCA models. Partial least squares projections to latent

structures discriminant analysis (PLS-DA) was then performed on selected experiments based on the PCA analysis. The multivariate software The Unscrambler®(CAMO, 2017b) and the distributed computation framework Apache Spark (Scala) were both used for PCA based on Singular Value Decomposition (SVD) (explained in Chapter 5.1), while The Unscrambler®and MATLAB were used for PLS-DA (using the NIPALS algorithm (Chapter 5.2)) and variable influence on projection (VIP) calculations respectively (point 6.). The PCA with Scala was mainly used in a procedure for determining electrode noise thresholds for the real-time STFT, explained in Chapter 4. Code for the procedure for extracting noise tresholds can be found on this Github page

4. In order to make progress in understanding the (electrical) neural dynamics from new recordings, extensive and well structured experiments with various types of (electrical) stimulation would be necessary. The author suggested that the Sandvig laboratory started experimenting with different types of stimuli to record more interesting data very early in the semester (19th January 2018). An excerpt from a mail with the stimulation suggestions are shown for completeness:

   ”..Practically, I would like some new recordings on MEA2 or a similar (adult?) culture containing various stimulations on one electrode, repetitively over a total duration of ex. 30 seconds (or longer). Stimulation protocols could be

   - Bi-phasic stimulation with linearly increasing rate, with ex. 15 repetitions.
   - Bi-phasic stimulation with linearly increasing amplitude, with ex. 15 repetitions.
   - Bi-phasic stimulation with a rate modulated according to a sine wave.
   - Bi-phasic stimulation with an amplitude modulated according to a sine wave.

   The thought is to

   (a) Find stationary properties correlating to stimulation protocol.

   (b) See if stimulation protocol changes stationary properties by comparing responses from start/early repetition with end/late repetition in the protocol.

   If some properties in the MEA activity is found to strongly correlate to a specific protocol, then more complex control of the MEA could probably achieved by stimulation more than one electrode.”

   Nevertheless, the suggestions were never fulfilled. However, SHODAN would likely be able to perform such complex stimulation experiments in the future. It must be noted that stimulation could also be regarded as other types of markers that can affect the growth and measured activity of a culture. For example, various types of proteins and other chemical compounds that lead to known chemical effects.

5. This task touches upon the Master’s thesis of PhD candidate Martinius Knudsen (Knudsen, 2016), where various approaches for training a biological neural network were discussed. One of them was using a spiking neural network (SNN) simulator, such as the The Brian spiking neural network simulator, to model the observed activity from a microelectrode Array (MEA) containing actual biological neurons. The work was central in laying the foundations for the Cyborg project. The Master’s thesis has a focus how the outcome of

MVA can help in this progress. MVA results could perhaps give directions when formulating mathematical models of specific MEAs and training black boxes such as artificial neural networks (ANNs) to model specific MEAs. This work do not introduce any computational models except for the linear PCA and PLS-DA.

6. Lastly, PLS-DA models were made on sets of two experiments proven interesting by PCA. The PLS-DA models were made in The Unscrambler®, as previously described i point 3. The goal of PLS-DA models was to analyze the main difference between two experiments. A method for calculating influence of individual variables (Galindo-Prieto et al., 2014) adjusted to PLS-DA case was performed as well (Chapter 6). The VIP calculations were done in MATLAB, and then visualized in The Unscrambler®.

1.3 Long-term goals and associated works

As previously stated in the project report, the main goal of the data analysis in the NTNU Cyborg project is to better understand signaling, dynamics, change and learning in biological neural networks (Hovden, 2017). Making a biological neural culture learn to drive a robot with the objective of not running into walls is a concrete approach into the landscape in the search for this knowledge (Aaser et al., 2017). The research are relevant in many sciences, such as cognitive sciences, and neuroscience such as translative and regenerative neuroscience. The dopaminergic neuonal cells analyzed were part of reserach on Parkinson's disease at St. Olavs, NTNU (Valderhaug, V. D., Ramstad, O. H et. al.). In the context of associated research projects of Cyborg coordinator Gunnar Tufte, biological neural networks are just one type of medium that that can be used for evolutionary computing. The NASCENCE project investigates how nanoscale material systems in microelectrode arrays (MEAs) can be used for computing tasks using artificial evolution (Broersma, 2018). This can involve regarding MEAs as black-box computing devices with no strict architectural requirements opposed to the traditional von Neumann architecture in modern CPUs. Reservoir computing (involving artificial recurrent neural networks, see (Burkow, 2016)) is one approach that has been investigated in the Cyborg project (Aaser et al., 2017). Related is the random boolean network (Gershenson, 2004) as a generalization of cellular automata (Langton, 1990). A recent interview on forskning.no can be found here (Norwegian). Another related project is NTNU Morphogenetic Engineering.

Making appliances with machine and living nerve tissue, such as controlling a robot with a biological brain, introduces interesting and challenging research (take a look at existing colorbluePhD, Master and specialization projects, publications and its Experts in Teamwork (EiT) page). The Cyborg project has research goals in both philosophical, biomedical and technological sciences. The author considers this work as part of the NTNU Cyborg project. Some external work that are related to this thesis are the project work (specialization project) of Helge-Andre Langåker (Langåker, 2017) (analysis of neural data) and upcoming Master's thesis and PhD work of Peter Aaser (Aaser, 2018) (computer-neuro systems).

Chapter 2

Background

This chapter starts with a biological background on the type of of neuronal culture analyzed as well as an overview of recording methodology at the lab (Chapter 2.1). It then continues with a review on the common methods of recording extracellular activity (Chapter 2.2), a short review of biological functions of dopamine in the brain in animals and humans (Chapter 2.2.1), a section about the classic EEG brain frequency bands (Chapter 2.3) as well as a small review of neural codes and control signals (Chapter 2.4). Then earlier results from analysis on the same microelectrode-array (MEA) culture are presented (Chapter 2.5). These results motivates for a new preprocessing approach (Chapter 2.6). Lastly, an overview of the final data selection is given (Chapter 2.7).

2.1 Neuronal cell differentiation at Sandvig lab

2.1.1 Briefly how the culture was made

The in-vitro midbrain DAergic neuronal culture was made following a procedure of neuronal cell differentiation from human induced pluripotent stem cells (hiPSCs) at Axel and Ioanna Sandvigs' laboratory of Translational and Regenerative Neuroscience at the Department of Neuromedicine and Movement Science (INB), Faculty of Medicine and Health Sciences, NTNU. The protocol for differentiating iPSCs into dopaminergic neurons was an adaption from several other protocols (Kirkeby et al., 2013, 2017; Doi et al., 2014).

The seeding, lab recordings and maintaining of the culture was conducted by PhD students Vibeke Devold Valderhaug, Ola Huse Ramstad and Rosanne van de Wijdeven. The seeding was a 15 day long specific protocol of adding various neuronal property enhancing ingredients. From day 16 and on, it has received a weekly feeding with neuronal maintenance media.

The resulting population was a mixed cell culture containing a specific type of dissociated human mDA neurons (naturally existing in substantia nigra pars compacta (SNc, A9 group)), which are known to have a selective vulnerability in Parkinson's disease (PD) (Arenas et al., 2015; Barker et al., 2015). To study their electrophysiological properties, the cells were maintained on a MEA2100 MultiChannel Systems MEA (MultiChannelSystems, 2017a) as seen in Figure 2.1c. When the activity was not recorded, the MEA was placed in an incubator with appropriate living conditions.

The neuronal culture was seeded on 11. of October 2016. Two experiments relating to Parkinson's research was done November 11th and 16th 2016, where DA was added, but they were not analyzed in this work. It was nicknamed "Dopey" by the PhD students from its characteristics and long-living property, and was registered as MEA 2.

2.1.2 MCS Experimeter/Analyzer to record/review experiments

Raw MEA recordings were obtained using MultiChannel Experimeter (MultiChannelSystems, 2017d). The raw data for selected experiment #2 is shown offline in MultiChannel Analyzer (MultiChannelSystems, 2017b) in Figure 2.2

Each experiment (a single recording of the culture on the MEA mounted on the MEA2100 device connected to a computer running MCS software, Figures 2.1c and 2.1d) typically resulted in 7-10 minute long discrete voltage recordings of 60 electrodes in the piko voltage (pV) range. The sampling rate was 10000 Hz. The recordings have unknown Signal-to-Noise Ratio (SNR). Additionally, the SNR seems to vary from experiment to experiment.

2.1.3 Stimulation in MCS Experimeter

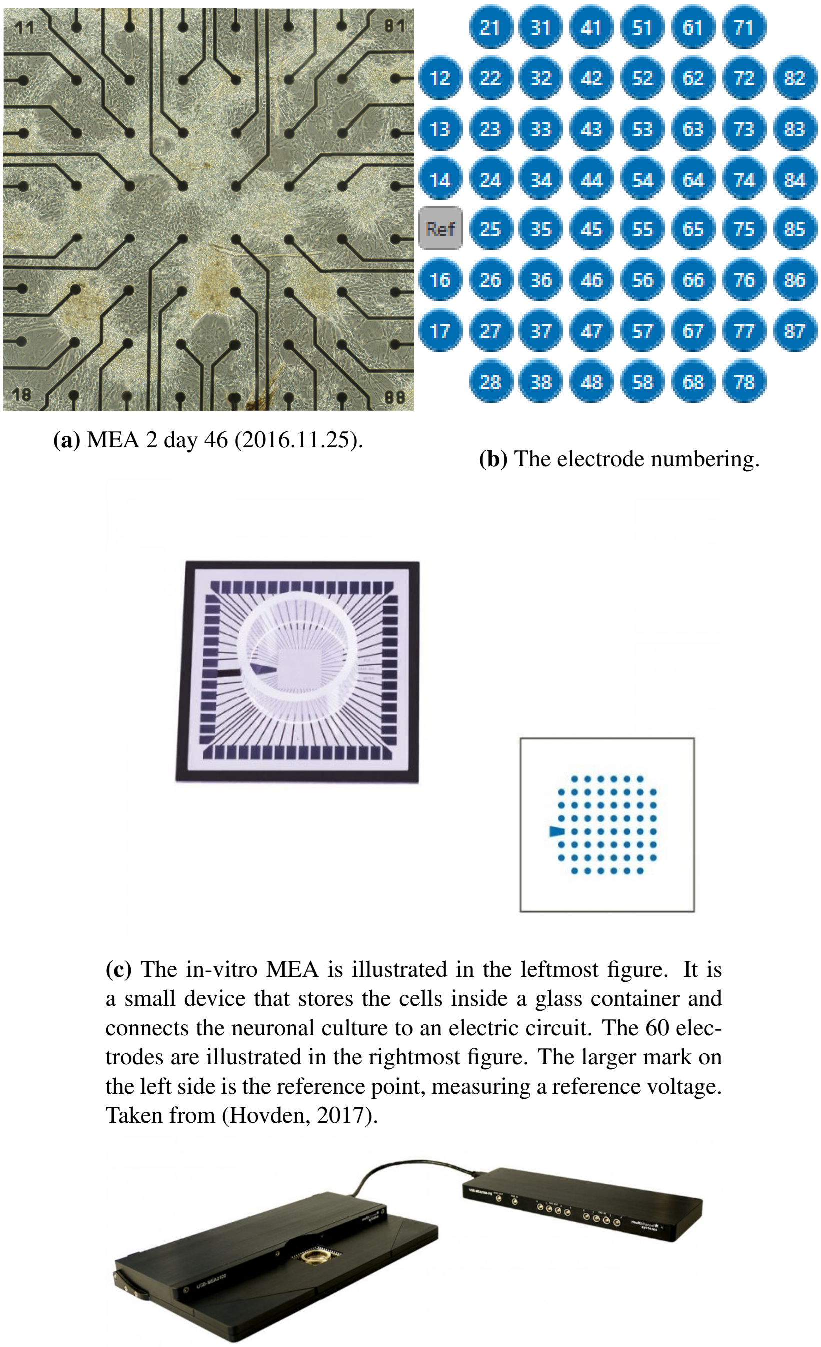
Many of the experiments received stimulation on electrode 74, however the data is of poor quality in this aspect. It was needed to manually look on the raw data in order to verify that stimulation was applied. The stumulation was an electrical bi-phasic pulse of ±500 µV, each phase lasting 100 µs, applied on electrode 74 every 30 s of reading. It is assumed that stimulation was applied in order to motivate neuronal growth / development.

2.2 Overview of recording extracellular activity

The consensus regarding signals and recording techniques today when looking at extracellular brain activity is summarized in Figure 2.3. In short, the recording approaches are Electroencephalography (EEG), Magnetoencephalography (MEG), Electrocorticography (ECoG) and various kinds of Intracortical Neuron Recordings (Nicolas-Alonso and Gomez-Gil, 2012). Although MEG is an intracellular recording technique, it is mentioned because its signals are comparable to EEG signals (Waldert et al., 2009).

For intracortical recordings such as MEAs placed onto the brain after surgery and variants (in-vivo, Viventi et al. (2012) (ECoG), Maynard et al. (1997)), Local Field Potentials (LFPs) are extracted by low-pass filtering ($f < 300$ Hz) of the neuron activity recorded from electrodes. Single-unit activity (SUA) signals are obtained by high-pass filtering ($f > 300$ Hz) signals of a single neuron, then spike detection and spike sorting. Multi-unit avtivities (MUAs) are obtained in the same way, but the signals may come from multiple neurons (Nicolas-Alonso and Gomez-Gil, 2012; Waldert et al., 2009; Im and Seo, 2016).

Extracellular Action Potentials (EAPs) are analog spike signals roughly in the frequency range $f \in [300, 3000]$ Hz caused by firing actions of single neurons (Obien et al., 2015; Pine, 1980). LFPs and EAPs are analog signals while SUA and MUA measure the spiking activity of single



(a) MEA 2 day 46 (2016.11.25).

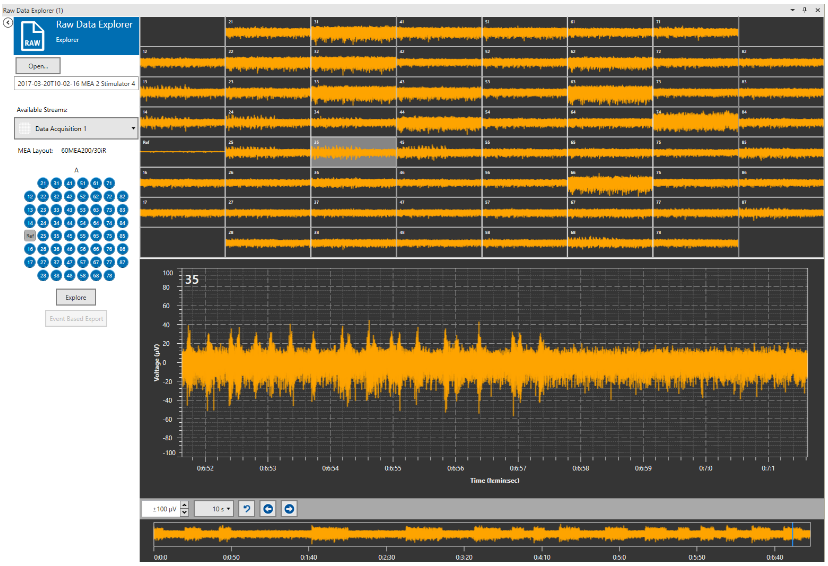
	21	31	41	51	61	71	
12	22	32	42	52	62	72	82
13	23	33	43	53	63	73	83
14	24	34	44	54	64	74	84
Ref	25	35	45	55	65	75	85
16	26	36	46	56	66	76	86
17	27	37	47	57	67	77	87
	28	38	48	58	68	78	

(b) The electrode numbering.

(c) The in-vitro MEA is illustrated in the leftmost figure. It is a small device that stores the cells inside a glass container and connects the neuronal culture to an electric circuit. The 60 electrodes are illustrated in the rightmost figure. The larger mark on the left side is the reference point, measuring a reference voltage. Taken from (Hovden, 2017).

(d) The MEA2100 recording and stimulation system (MultiChannelSystems, 2017a).

Figure 2.1: MEA2100 system.



Raw Data Explorer (1)

2017-03-20T10-02-16 MEA 2 Stimulator 4

Available Streams: Data Acquisition 1

MEA Layout: 60MEA200/30iR

Figure 2.2: Raw recording of #2 with electrode 35 marked. 35 shows some interesting bursting behavior. Stimulation electrode 74 shows high amplitudes (the small window to the upper right with largest amplitudes, not marked for display in the larger lower window). 10 second windows of the end of #2 from MCS Analyzer (MultiChannelSystems, 2017b). Taken from (Hovden, 2017).

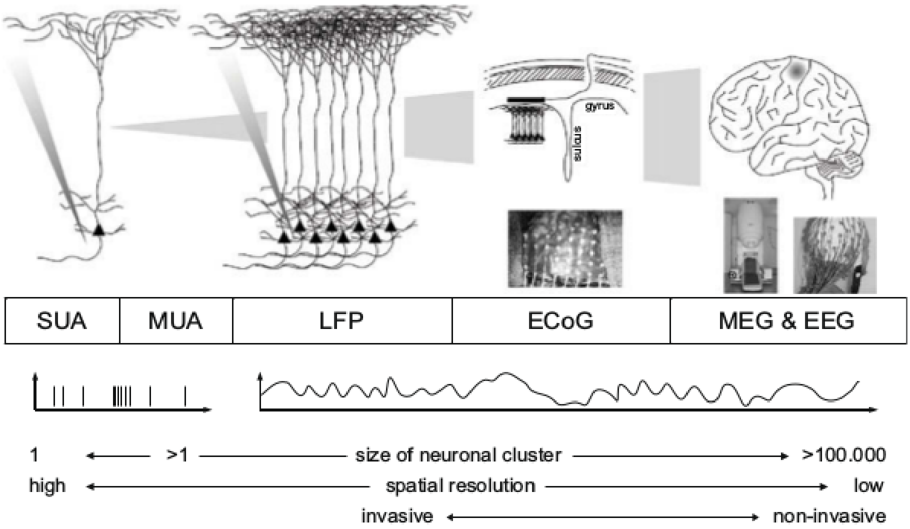


Figure 2.3: Various recording techniques and signals to measure brain activity. Taken from (Waldert et al., 2009). An in-vitro MEA such as MEA2100 used in this report, measures electrical currents on its electrodes. MEA measurements are a combination of LFPs and analog forms of SUAs and MUAs (or EAPs) (Obien et al., 2015; Waldert et al., 2009).

neurons and can be reduced to discrete events in time (Nicolas-Alonso and Gomez-Gil, 2012). (Obien et al., 2015) shows by experiments a correlation of intracellular action potentials (IAPs) and EAPs. Assuming that IAPs and SUAs are closely related, EAPs could be seen as an analog version of SUAs.

(Obien et al., 2015) proposes that data measured from in-vivo and in-vitro MEAs are similar. For this reason, it is assumed that in-vivo brain signaling such as LFPs can also be measured from in-vitro MEAs containing comparable neuronal cells. MEA recordings is therefore considered as electrophysiological extracellular recordings of LFPs, EAPs and summations of EAPs (Obien et al., 2015; Waldert et al., 2009). In this thesis, relatively quick rise and fall in amplitude (in the millisecond range) in the recordings is simply denoted as an action potential (AP). It could be caused by either EAPs or quick oscillations in LFP.

Measuring EAPs from in-vitro MEAs is well described in (Pine, 1980). In (Gold et al., 2006), an attempt to make a mathematical model of EAPs in order to draw another connection between intra- and extracellular recordings is made. (Bai and Wise, 2001) discusses a technique for recording SUAs using both in-vitro and in-vivo active MEAs.

EEG is the most common extracellular recording technique and consists of placing electrodes onto the scull (Buzsáki et al., 2012), hence non-invasive. EEG and MEG signals are said to be smoothed versions of the local field potential (LFP) (Buzsáki et al., 2012; Waldert et al., 2009).

EEG, MEG and ECoG measured signals are simply called EEG, MEG and ECoG signals, while Intracortical Neuron Recorded signals are Local Field Potentials (LFP), Multi-Unit activity (MUA) or Single-Unit activity (SUA) (Im and Seo, 2016; Nicolas-Alonso and Gomez-Gil, 2012). Opposed to the other signals mentioned, MEG reflects intracellular currents flowing through dendrites which produce magnetic fields measurable outside the head (Waldert et al., 2009).

2.2.1 Functions of dopaminergic neurons in animals and humans, and Parkinson's disease (PD)

In both animals and humans, the primary functions of dopamine (DA) neurons is in general related to motivation, and regulate approach behavior. The basic functionality is activation and inactivation of DA neurons, thereby regulating mood by inducing reward and aversion, respectively (Ikemoto et al., 2015). In the mammalian brain, activation of DA neurons such as the ones originating from the A9 group analysed in this report, leads to release of 3-hydroxytyramine (named dopamine), a metabolite of the amino acid tyrosine, into either of the four major dopaminergic (DAergic) pathways known in the mammalian brain (Beaulieu and Gainetdinov, 2011). Release of DA from midbrain DA neurons influences voluntary movement, cognition, and motivational state (Deignan et al., 2012). There is a diversity in types of DA neurons in humans and animals, and some subtypes even develop postnataly.

It is well known that lack of DA in the brain in humans is related to PD (Bernheimer et al., 1973). Other diseases related to inappropriate DA levels are Schizophrenia, Alzheimer's and Huntington's diseases (Seeman et al., 1987). Levels of DA have shown to have a role in both locomotor behavior and learning in humans (Beninger, 1983). Perhaps, in the simplistic biological view of the author, dopamine and dopaminergic neurons could be seen as the major functional parts in the biological neural system equivalent of reinforcement learning (see Kaelbling et al. (1996)). (Holroyd and Coles, 2002) describes such a biological equivalent. Recent research has made it possible to generate human pluripotent stem cell (hPSC)-derived DA neurons, capable of inducing behavioral recovery in animal models of PD. This could help in development of cell replacement therapy as a future treatment in PD (La Manno et al., 2016). (Jo et al., 2016) even developed a method to differentiate hPSCs into a large multicellular organoid-like structure that contains distinct layers of neuronal cells expressing characteristic markers of human midbrain. They detected electrically active and functionally mature mDA neurons as well as DA production.

DAergic signaling in has been related to various kinds of cognitive tasks in the mammalian brain, which makes analysis of DA neurons interesting in the Cyborg perspective. Some examples are:

- (Barter et al., 2015) reports from in-vivo experiments on movement of rats identification of DA SNc neurons representing vector components of velocity or acceleration in the directions up, down, left and right, hence showing DA neurons belonging to a velocity control circuit. They hypothesized that DA signaling implements gain adjustment for adaptive transition control (Barter et al., 2015). This is discussed in result section in the project report (Hovden, 2017).

- (Howe and Dombeck, 2016) reports a new optical method of in-vivo recording rapid signaling in distinct dopaminergic axons during locomotion and reward on mice, suggesting that DA neuromodulation can differentially impact motor control and reward learning with sub-second precision, and that the results indicate that both precise signal timing and neuronal subtype are important parameters to consider in the treatment of DA-related disorders.

- (Jay et al., 2015) conducted an in-vivo study of endogenous activity using paired supraspinal DA neuron and motoneuron recordings on awake, paralyzed zebrafish, reporting that show that supraspinal DAergic neurons generate two forms of output: tonic spiking and phasic bursting. Tonic spiking being related to inactivity, bursting correlating to locomotor output.

- (Gadagkar et al., 2015) even reports on the task of DA neurons in the basal ganglia in the zebra finch (a bird) on giving auditory feedback on the magnitude of distortion in its bird song, as well as relating to movement.

- (Wagenaar et al., 2005) investigates stimulation of in-vitro rat DAergic neuronal MEA cultures in a closed-loop feedback system in order to control spontaneous bursting behavior. The work is discussed in result section in the project report (Hovden, 2017).

2.3 Brain frequency bands

Traditionally, the frequencies in conventional EEG brain bands have been systematized into bands. A table presenting the EEG bands (MultiChannelSystems, 2017b) as well as an unspecified band is shown in Table 2.1.

(Nácher et al., 2013) makes it clear that coherent oscillations on theta, alpha, beta and gamma frequency range have been proposed as mechanisms that coordinate neural activity in large-scale cortical networks in sensory, motor, and cognitive tasks. They propose in this work that coherent delta-band oscillations between cortical areas correlate with decision making in monkeys. All in all, oscillations in any of the EEG bands have been found to relate to different tasks in both humans and animals.

Hz	Band
$[0.5, 4\rangle$	delta
$[4, 8\rangle$	theta
$[8, 13\rangle$	alpha
$[13, 30\rangle$	beta
$[30, 120\rangle$	gamma
$[0, 0.5\rangle$ and $[120, \infty\rangle$	unspecified

Table 2.1: The typical EEG frequency bands with an additional ”unspecified” band. Taken from (Hovden and Gulbrandsen, 2017a).

The preprocessing in this thesis analyses mainly APs visible in the form of envelope frequencies in the delta and theta band. These occur in different frequency bins of the sliding time-frequency window (Figure 2.19). In traditional EEG signals, large delta and theta oscillations are related to neurological disease in adults. Beta oscillations have been found to be attenuated or de-synchronized during real movement of motor imagery in humans (Nicolas-Alonso and Gomez-Gil, 2012), and have bin clearly observed in patients with PD progression (Hammond et al., 2007). The latter is further discussed in accordance to earlier analysis results in Chapter 2.5.1.

Alpha and gamma oscillations have previously been characterized as feedback and feedforward processing, respectively, in the monkey visual cortex, making a connection between oscillations in these bands regarding visual processing (van Kerkoerle et al., 2014). Besides being related to visual processing, alpha waves may also be related to memory functions and mental effort (Nicolas-Alonso and Gomez-Gil, 2012).

For further review on associated functions of EEG-band oscillations, see (Nicolas-Alonso and Gomez-Gil (2012), p. 1216 in section 2.1. ”Electroencephalography (EEG)”).

2.4 Neural codes and control signals

There exist a lot of research that try to decipher detected APs into useful information. They consist of different approaches of revealing neural codes, that might be observable through properties of the recorded APs from populations of or individual neurons.

Research has also gone further in interpreting how specific properties relate to specific physical tasks, for example by introducing the terms event-related desyncrhonization (ERD) and event-related syncrhonization (ERS) from observations of specific types of amplitude modulations in mu/alpha and beta rhythms (sensorimotor rhythms) in accordance with planning and physical movement of a finger in humans (Nicolas-Alonso and Gomez-Gil, 2012). In the cited work, basically, ERD (in alpha-band) appears before movement and reaches maximum during movement, while ERS (beta-band) reaches maximum after movement execution (Figure 2.4).

The most common neural codes are rate coding, temporal coding, population coding and dense coding (Knudsen, 2016). These are only the most basic widely accepted codes, and more sophisticated codes have also been reported in special brain and frequency regions. For example, (Lisman and Buzsáki, 2008) claims that theta and gamma oscillations work together to form a

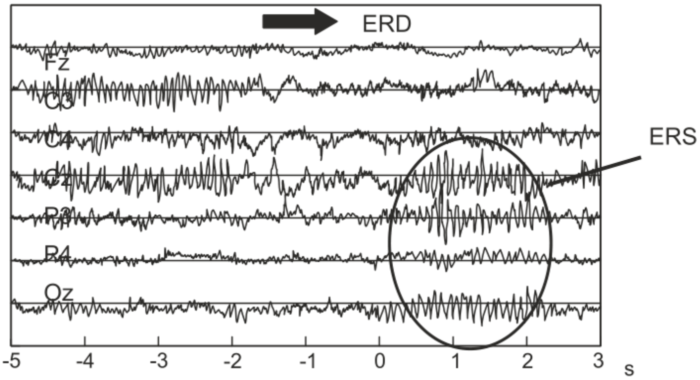


Figure 2.4: Possible control signals from EEG. Event-related desynchronization and event-related synchronization in ongoing EEG recorded during right finger movement. Taken from (Nicolas-Alonso and Gomez-Gil, 2012).

neural code in the rat hippocampus that is utilized during recall of sequences of items from long-term memory. Below are a brief overview and description of the basic neural codes and their connection to the analysis in the thesis.

2.4.1 Rate coding

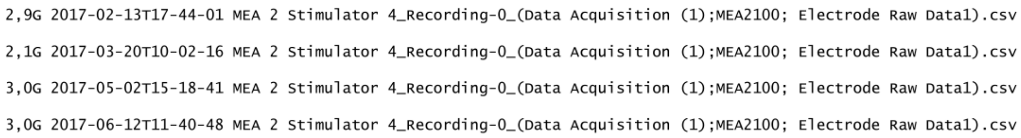
Information is encoded into the firing rate of the neuron. An analysis technique for rate codes is detecting spikes from APs and analyzing the inter-spike-interval (ISI). This was done in the project work.

2.4.2 Temporal coding

Information is encoded into timing of spikes in firing rate fluctuations. A temporal aspect of rate coding. Perhaps the preprocessing presented can capture some of these codes.

2.4.3 Population coding

Taking into account that APs are are typically summations of firing activity from a population of neurons, and each neuron is affected by firing of other neurons. Information is embedded into joint activities of a number of neurons. Correlation coding, a subcategory of population coding, suggests that additional information can be embedded into the correlations of multiple APs. This sounds good for MVA such as PCA, since it works on covariance measures (correlation is covariance divided by two standard deviations, from the two signals investigated).



```
2,9G 2017-02-13T17-44-01 MEA 2 Stimulator 4_Recording-0_(Data Acquisition (1);MEA2100; Electrode Raw Data1).csv
2,1G 2017-03-20T10-02-16 MEA 2 Stimulator 4_Recording-0_(Data Acquisition (1);MEA2100; Electrode Raw Data1).csv
3,0G 2017-05-02T15-18-41 MEA 2 Stimulator 4_Recording-0_(Data Acquisition (1);MEA2100; Electrode Raw Data1).csv
3,0G 2017-06-12T11-40-48 MEA 2 Stimulator 4_Recording-0_(Data Acquisition (1);MEA2100; Electrode Raw Data1).csv
```

Figure 2.5: The four selected experiments in earlier analysis. The figure displays .csv files containing full length raw recordings of MEA experiment #1, #2, #3 and #4 in the given order. They are files resulting from ASCII file export in MultiChannel DataManager (MultiChannelSystems, 2017c). The project work analyzed the entire $7 - 10$ minute length raw data (Hovden, 2017), while the TTK7 work analyzed only 2 second length sample segments of the data (Hovden and Gulbrandsen, 2017a).

2.4.4 Sparse coding

Takes into account that the APs come from a population of neurons (like population coding), but with the assumption that only a small percentage of the neurons carry information. In contrast, population coding is also called dense coding. Describing the data in a MVA model by only using a fixed number of components can be regarded as a sparse representation of the data.

2.5 Summary of results from earlier analysis

From a data set of 86 recorded MEA experiments in the time range 17. of October 2016 to 16. of June 2017, four experiments named #1, #2, #3 and #4 respectively, were previously selected for analysis (Figure 2.5). The experiments were recordings of the dopaminergic neuronal culture called MEA2, nicknamed ”Dopey”. See Chapter 2.1 for description of the culture, cell differentiation and recording procedures.

All experiments except #4 received external voltage stimulation described in the Chapter 2.1.3. The four experiments were converted from their original raw files to .csv format using Multi Channel DataManager, a tool from MultiChannel Systems (MultiChannelSystems, 2017c). Figures 2.7, 2.8, 2.9, 2.10 and 2.12 sum up the most important results from earlier analysis on MEA2 (Hovden, 2017; Hovden and Gulbrandsen, 2017b,a).

2.5.1 Empirical Mode Decomposition (EMD), Power Spectral Densities (PSDs) and energy calculations reveal power changes across experiments over time

The first analysis performed on the data were Electroencephalography (EEG) band (Chapter 2.3) signal reconstructions and energy calculations based on Empirical Mode Decomposition (EMD) and Power Spectral Densities (PSDs) in the course ”TTK7 Adaptive Data Analysis: Theory and Applications” (Hovden and Gulbrandsen, 2017a). EMD is a data driven method for decomposing a signal into Intrinsic Mode Functions (IMFs). Opposed to traditional Fourier analysis, these IMFs might contain nonlinear and/or non-stationary properties of the signal (Huang et al., 1998). PSD is a scaled version of the Fast Fourier Transform (FFT, Appendix Chapter 8.4.11) in order so that energy is represented independent of the sampling rate, hence gives an estimate of signal

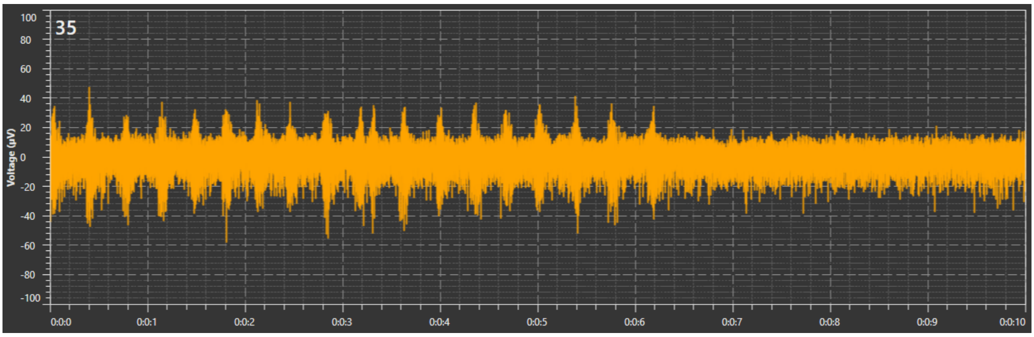


Figure 2.6: The first 10 seconds of raw data of electrode 35 from experiment 2017-03-20T10-02-16 (#2 from project report) as seen in MultiChannel Analyzer (MultiChannelSystems, 2017b). A 6 second long burst can be seen at the start of the recording.

power (see Appendix equation 8.15). The FFT is an efficient algorithm for computing the Discrete Fourier Transform (DFT) on the computer for Fourier analysis.

Figures 2.7 and 2.8 show reconstructed EEG band signals (EEG brain bands described in Chapter 2.3) by summation of IMFs after ensemble EMD (EEMD) on data from electrode 47 for the four experiments, as well as their PSDs. EEMD is a version of EMD using white noise to gain better decomposition (Wu and Huang, 2009).

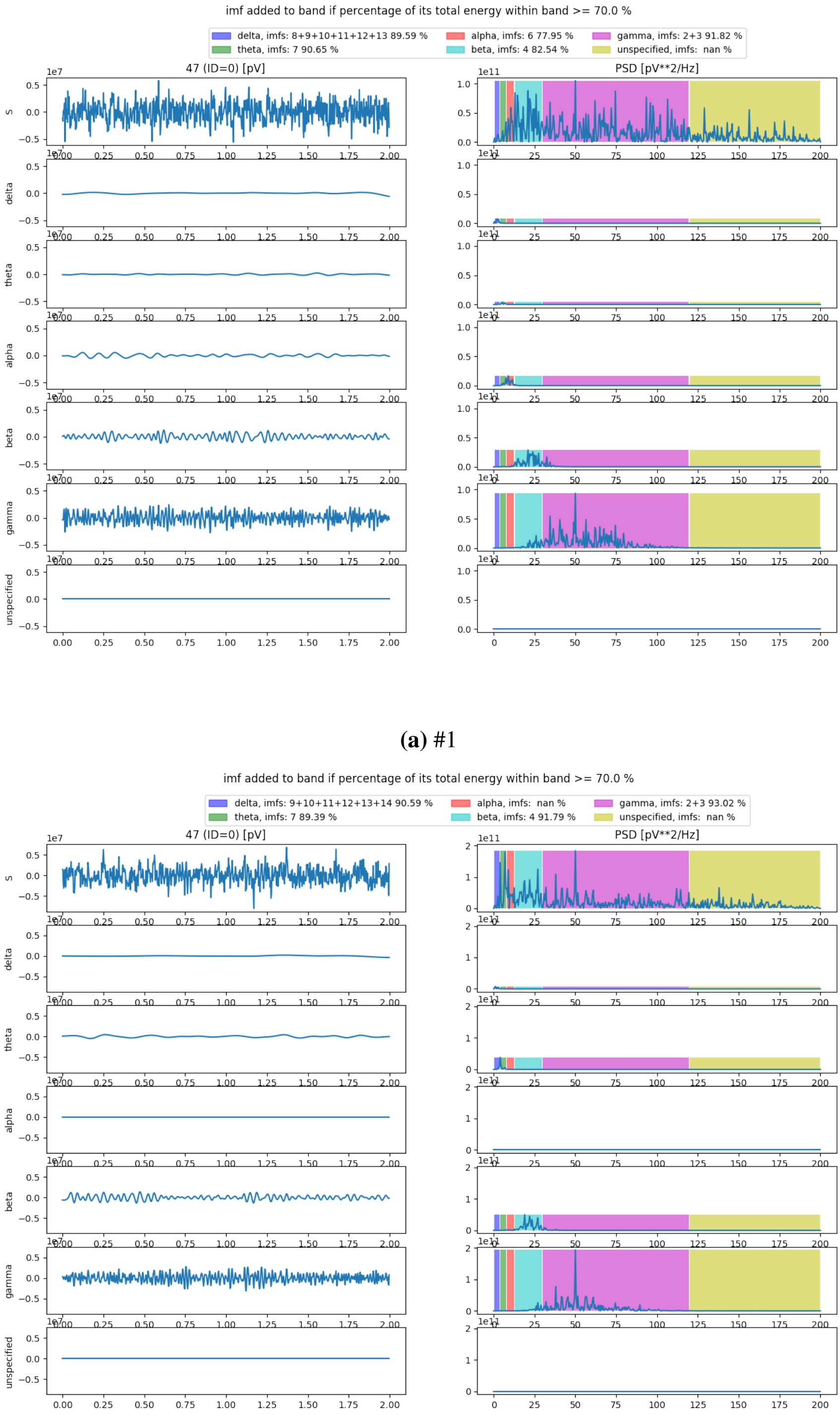
The reconstructed EEG band signals (Figures 2.7 and 2.8) were based on the raw signal from the last 2 seconds of electrode 47. The height of the marked colors in PSDs are determined from the max peak in the PSD. The gamma signal has the highest oscillations in #3. Decreasing beta oscillations across the experiments might bee seen. The trend in decreasing power in the beta signal is further verified in Figures 2.9 (b) and 2.10.

The selection of an appropriate number of IMFs to add together in order to reconstruct an EEG band signal was based on a criteria that minimum 70 % of the IMFs' energy were within the band in the PSD. Which IMFs that were made to reconstruct the band signals are shown in the legend (Figures 2.7 and 2.8). For each experiment, the amplitudes are scaled correctly according to each other for brain signals and PSDs. The reconstructed delta signal should be mostly noise because the original electrode signal was highpass filtered $f > 4$ Hz of same reason for removing $0 - 10$ Hz in PSDs for PCA in the project report (assumed low SNR) (Hovden, 2017).

Figures 2.9 and 2.10 plot the total power in the EEG bands for the four experiments using two different methods for calculating power (power from PSD EEG bands, power from reconstructed EEG band signals). The reconstructed EEG band signals were used to calculate the signal powers in Figure 2.10. The comparable signal powers in Figure 2.9 were calculated directly from the PSD of the raw 2 second signals.

Figure 2.9 (a) shows a significant power increase outside EEG bands (unspecified, $f \in [120, inf\rangle)$. Figures 2.9 (b) and 2.10 (a) show energy decrease in beta band (blue) and energy peak in gamma-band in #3, while Figure 2.10 (b) is a zoomed in version of 2.10 (a) to investigate the beta band power decrease.

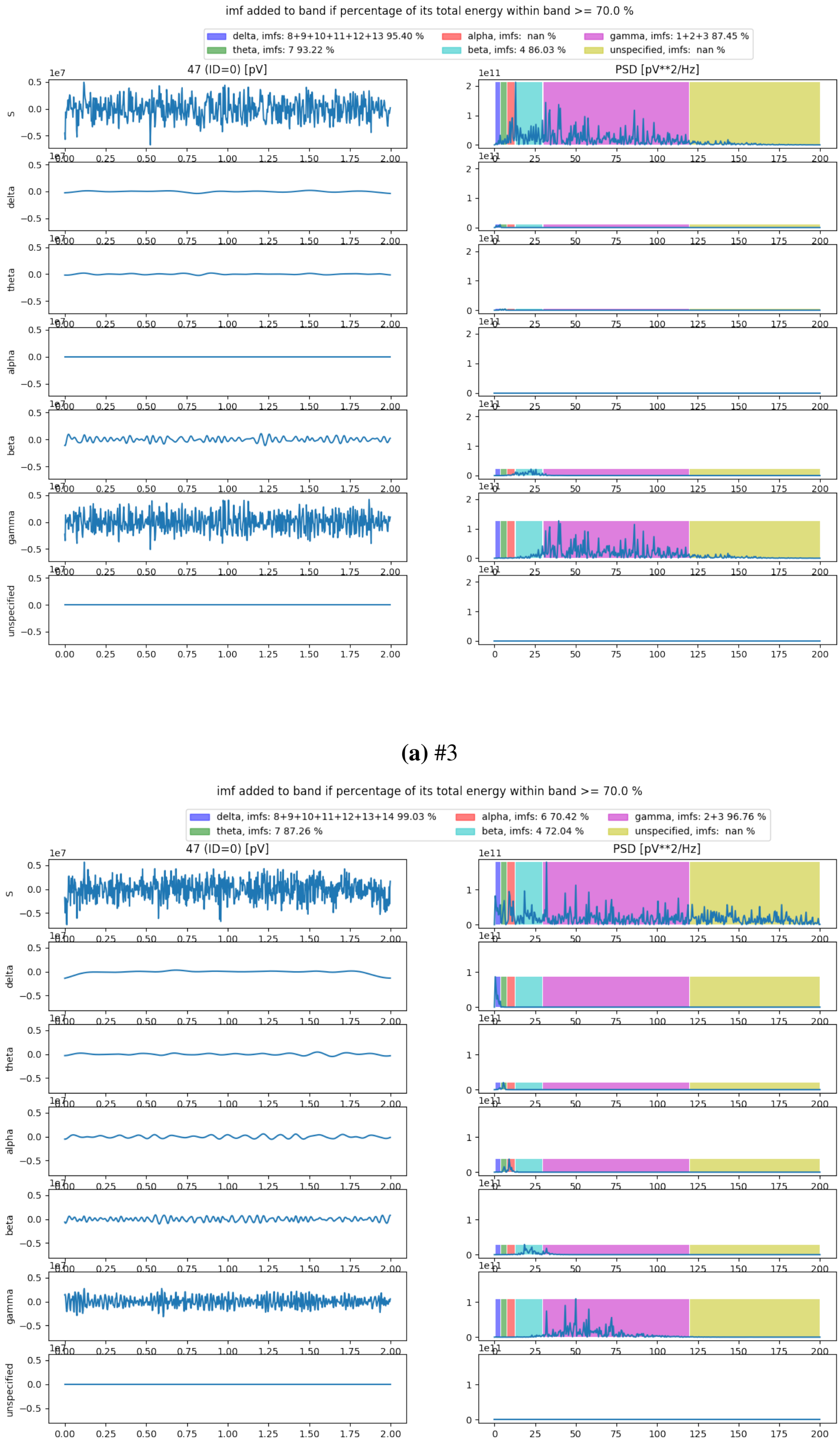
The significant result is that **beta band power decreases** throughout the age of the culture (#4 experiment of oldest culture), and that there is a **large power increase in frequencies above**



(a) #1

(b) #2

Figure 2.7: Earlier work: Attempt to reconstruct EEG-band signals from 2 seconds of MEA experiment #1 and #2 (by summation of IMFs). From TTK7 report (Hovden and Gulbrandsen, 2017a).



(a) #3

(b) #4

Figure 2.8: Earlier work: Attempt to reconstruct EEG-band signals from 2 seconds of MEA experiment #3 and #4 (by summation of IMFs). From TTK7 report (Hovden and Gulbrandsen, 2017a).

any of the EEG bands. The observed decrease in power in beta band ($13 - 30$ Hz) (and thus de-syncrhonization) was proposed to relate to the fact that the concentration of dopaminergic neurons increased as the age of the culture progressed (**less Parkinson's-like culture with age progression?**).

It is not clear, but highly likely, that the the amount of dopaminergic neurons increased over time. It was reported from the Sandvig laboratory that in order to find out the concentration of these neurons, the culture would have been killed (Valderhaug, V. D.). Nevertheless, a lot of existing research on PD patients report on a connection between loss of dopaminergic neurons and increased beta power (Weinberger et al., 2006; Brown et al., 2001; Levy et al., 2002; Jenkinson and Brown, 2011). In addition to this work (Hovden and Gulbrandsen, 2017a), results in the project work also observed decreased beta band power (Hovden, 2017).

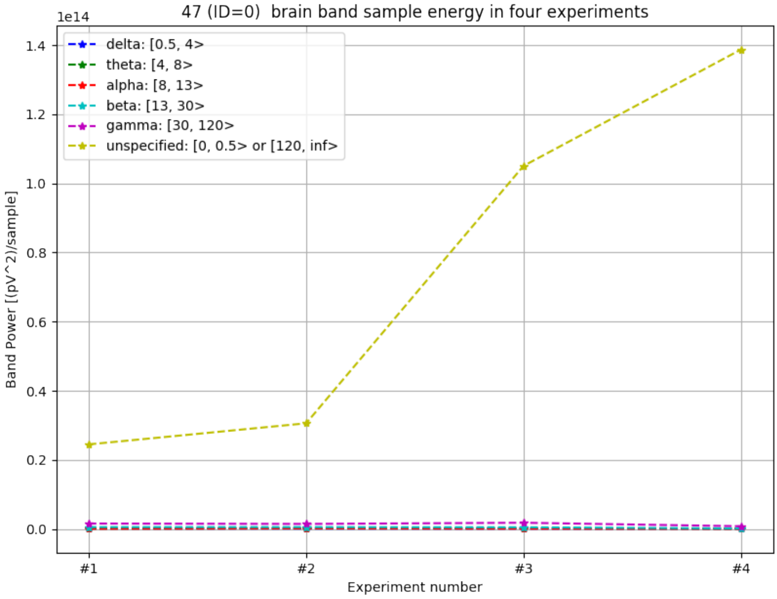
To sum up, the activity in the beta band was well investigated in the DAergic culture. However, the increase in power in higher frequencies outside the EEG bands was not. It was thus proposed that further analysis on the data should focus on frequencies higher than the EEG bands.

2.5.2 PCA on Power Spectral Densities and Interspike-Interval histograms to map energy variations in PSD and variations in spiking frequency to the electrode MEA layout

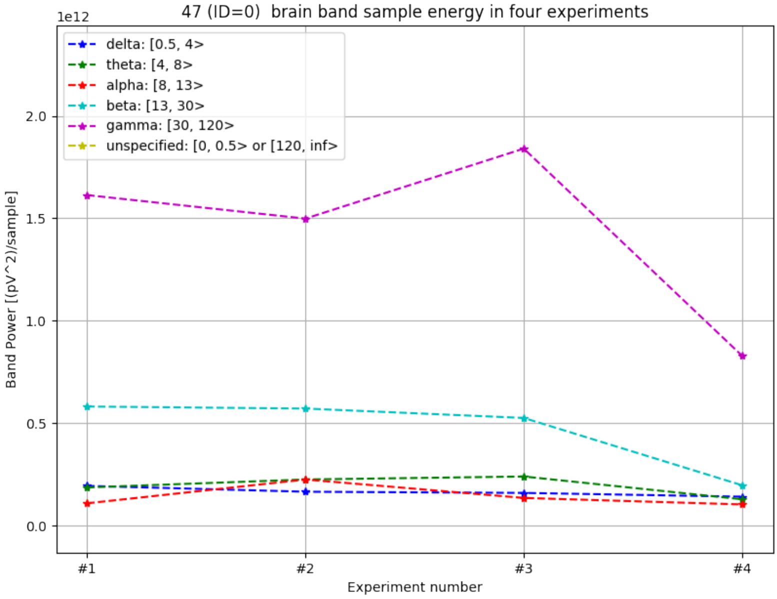
The discovery that most of the change in energy activity occurred in the larger ”unspecified” band lead to the motivation to perform PCA on PSDs. This was conducted in the project work (Hovden, 2017). For a selected experiment, for each approximately 10 minute long electrode recording, a single PSD was computed.

For a given experiment, the N PSDs were stacked column-wise in a $M \times N$ data matrix $\mathbf{X}$ where M is length of any of the PSDs and N number of electrodes. Each column thus contained the PSD for a given electrode. In order to get an overview of the general power trend across the experiments, an average PSD over all N electrodes for each of the four experiments was computed (Figure 2.11).

For each experiment, the preprocessed data in the form of PSDs in the PSD matrix $\mathbf{X}$ was then used to make a PCA model (Chapter 5.1). These models revealed patterns in energy activity among the electrodes. Additionally, Inter-Spike interval (ISI) histograms (Appendix Chapter 8.3.2) produced in MultiChannel Analyzer (MultiChannelSystems, 2017b) were used to make complementing PCA models that revealed patterns in spiking activity in the experiments. In this case, the data matrix was an $M \times N$ matrix $\mathbf{X}$ with M being estimated frequencies according to equation 8.2 and N the the number of electrodes. Insights on MEA activity from using both methods are summarized in Figure 2.12. These results tell that **some electrodes have more activity than others**, and that there is a **distinction between large variation of spiking activity and large variation of energy over frequencies**.

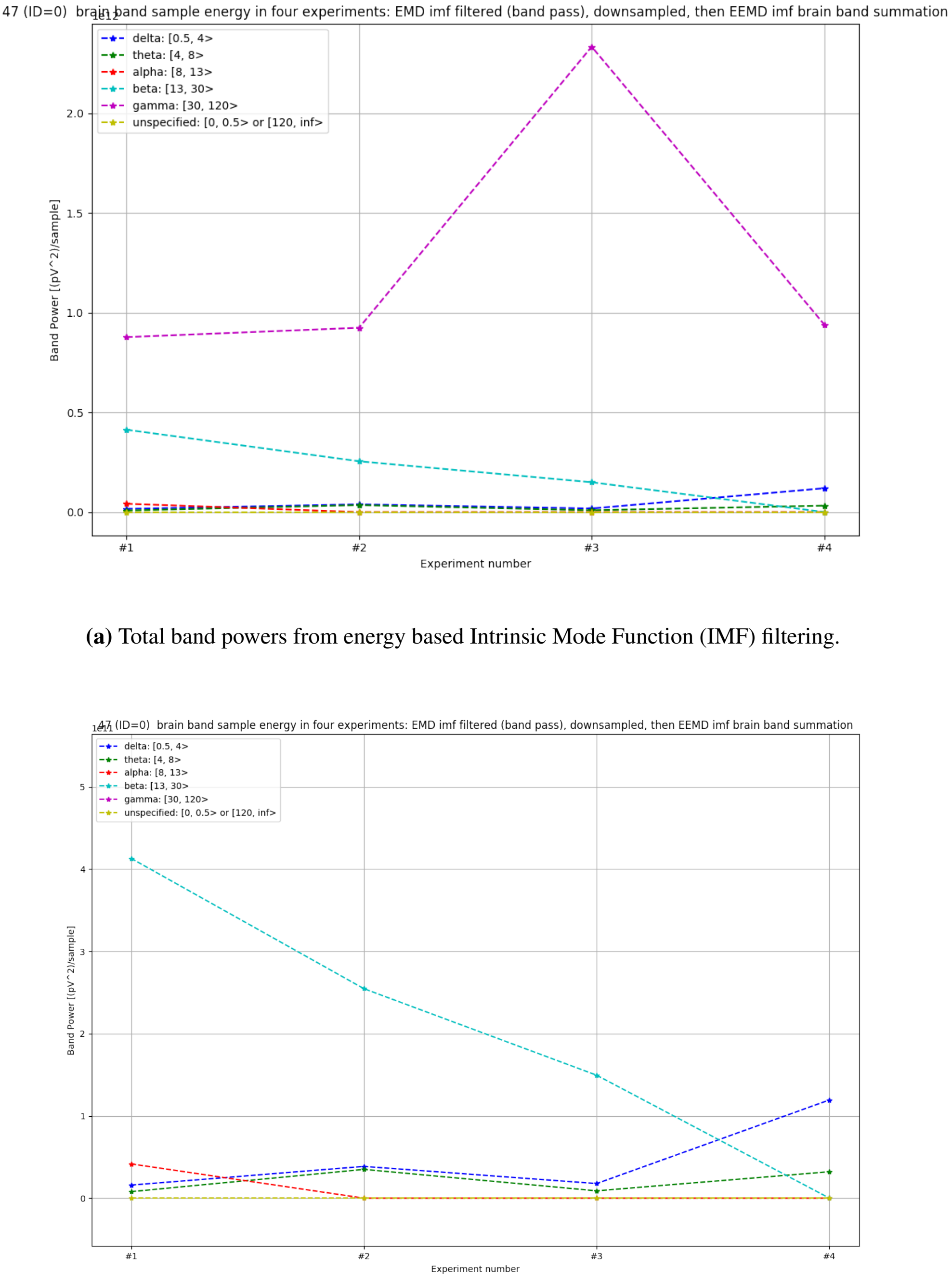


(a) Total band powers from PSD.



(b) (a) zoomed in on other bands

Figure 2.9: Power computations from PSD (frequency domain). From TTK7 report (Hovden and Gulbrandsen, 2017a).



(a) Total band powers from energy based Intrinsic Mode Function (IMF) filtering.

(b) (a) zoomed in on other bands.

Figure 2.10: Power computations from IMFs (time domain). From TTK7 report (Hovden and Gulbrandsen, 2017a).

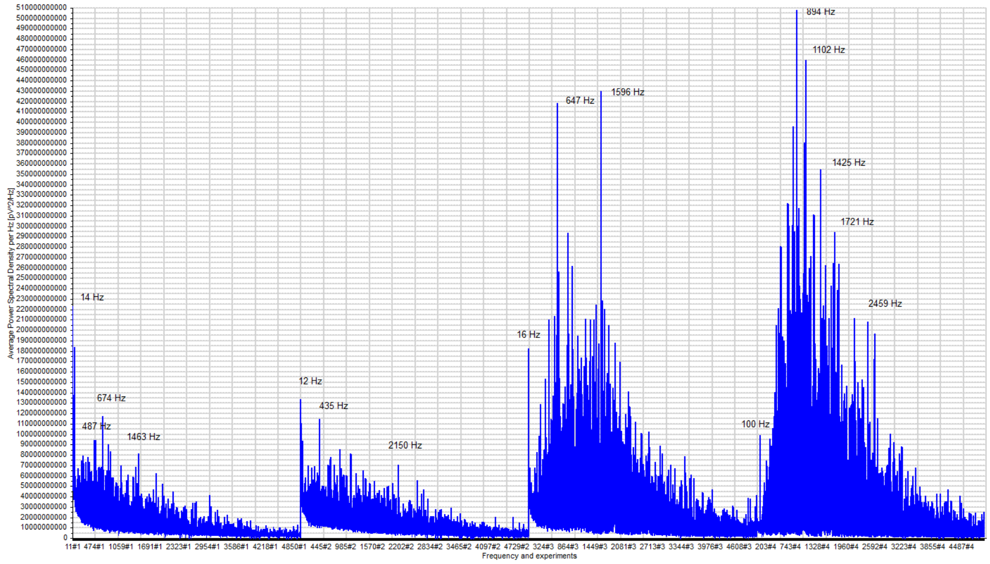


Figure 2.11: Average PSD for all 60 electrodes for the four experiments. The four sections in the x-axis are frequencies $11 - 5000$ Hz. The y-axis is pV^2/Hz. This shows clearly that there were significant energy increases in high frequencies. From project report (Hovden, 2017).

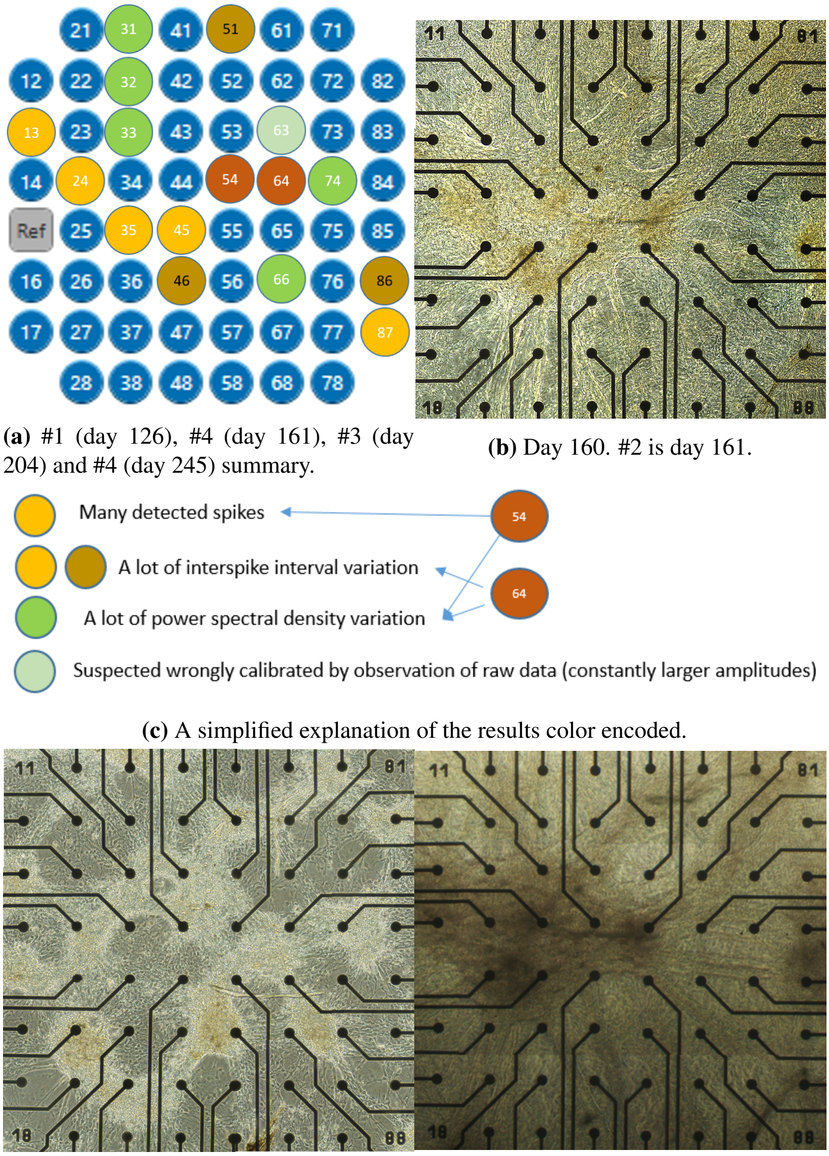
2.5.3 Spectrogram visualization of noise reduced signals shows that synchronicity in young culture develops to de-syncrhonized APs in frequency domain

Throughout the project work, it became clear from PCA models on PSDs that it might be interesting in future PCA models to more directly express variance in spiking activity instead of variance in power. As an extra effort into investigating spiking activity, selected electrodes from #2 and #4 were converted to sound, noise reduced in the audio editing program Audacity based on a visible noise segment (the same procedure later explained in Chapter 3.3) and visualized in the spectrogram (Appendix Chapter 8.4.11) visualizer Baudline as seen in Figure 2.13 (Hovden, 2017). Links to the sounds are in the description and a video of experiment #2 of the spectrogram on stereo sound with electrode 35 (left) and 87 (right) can be seen following this link. The conclusion is that **spectrograms might enable closer analysis of neural codes in the form of spiking activity in frequency components**.

2.6 Motivation for new preprocessing

From previous results, it seems that there is a lot more information in the activity of the neurons encoded in spike timing in frequency components, than there is into the (variations of) amplitudes in PSD over frequency in the culture.

This became apparent when performing PCA on Interspike-Interval (ISI, see Appendix 8.3.2)



(a) #1 (day 126), #4 (day 161), #3 (day 204) and #4 (day 245) summary.

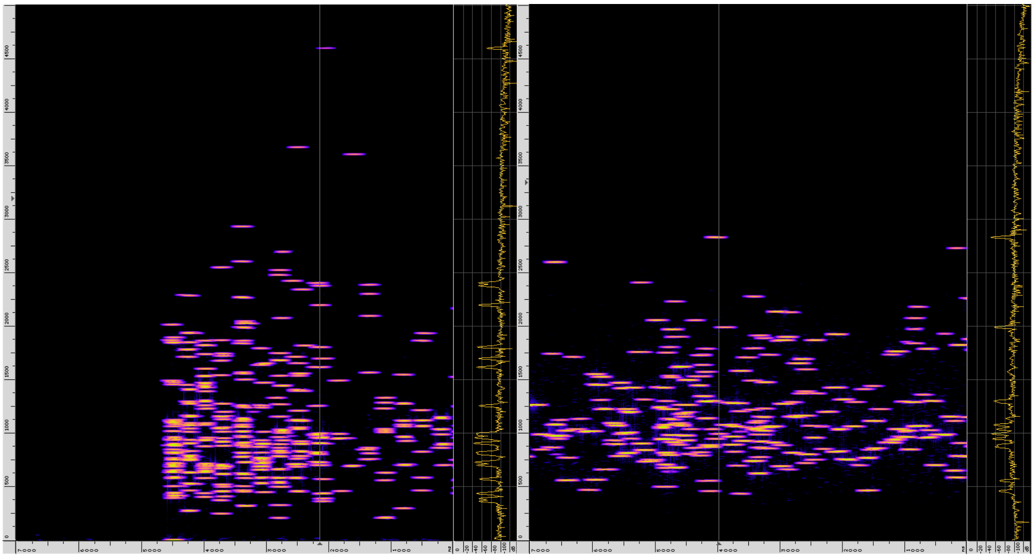
(b) Day 160. #2 is day 161.

(c) A simplified explanation of the results color encoded.

(d) Day 46.

(e) Day 403.

Figure 2.12: Summary of active electrodes in terms of variation in energy and number of detected spikes, as well as MEA pictures before, during and after selected experiments. Green electrodes had a lot of variation in their PDSs, which were well expressed by PCA on PSDs. Yellow and brown electrodes had a lot of spike activity, either in terms of detected spikes (yellow, 54) or variation of spike frequency of detected spikes (yellow and brown). 54 had a lot of variation in energy, as well as many detected spikes. 64 had a lot of variation in energy and in detected spikes. (a) and (b) might show that the dense brown areas (b) are connected to the spiking activity (yellow, brown and red), and/or that stimulation electrode 74 are connected to the varying energies (green, red). Work taken from project report (Hovden, 2017).



(a) #2 on 35 and 87. In-phase events. The sound can be heard here.

(b) #4 on 46 and 86. No phase-locked/synchronized events. The sound here.

Figure 2.13: Further investigation of spiking activity. Short-Time Fourier Transform (STFT) on a noise reduced selection of the most spiking-active electrodes from #2 and #4. The x-axis in (a) and (b) is a time window of 7 seconds, while the y-axis is frequency $f \in \langle 0, 5000]$ Hz. #2 shows the start of a 8 second long burst. In experiment #2, bursts seem to occur with an average length of approx. 10 s, in intervals of 10 s. In #4, comparable activity (bursts...) seem to happen almost continuously. In #4, most of the frequency components occur in a wide frequency range around 1000 Hz. Perhaps one can say that a synchronization of the frequency components are what makes the spikes in (a) and seen and detected in Figure 2.6. It is interesting that no continuously changing frequencies occur, and that the length of each individual frequency instance in (a) and (b) seem to be similar with this noise reduction. Also, most of each individual frequency instances happen at both electrodes (when not, it makes a stereo sound effect when the two electrodes are placed each on the left and right channel). However, the frequency of the instances happening simultaneously can be different. They can be apart from each other in frequency (height) in the figure. This also can be heard in stereo as described earlier. Links to the sounds are in (a) and (b). Work taken from project report (Hovden, 2017).

histogram data (on an $M \times N$ matrix $\mathbf{X}$ with M being estimated frequencies according to equation 8.2 and N the the number of electrodes) and comparing these results with PCA on PSDs (in an $M \times N$ matrix $\mathbf{X}$ with M being the length of the PSD and N the number of PSDs, each one correspinding to an electrode). At that stage, the PCA analysis on ISI data actually captured electrodes with large (variation of) spiking activity. (Hovden, 2017).

When spikes cannot be detected in traditional form in raw data by standard deviation based thresholds (Appendix Chapter 8.3.1) (and f. ex. no ISI histograms can be created), spectrograms on noise reduced data still show that spike-like activity with comparable high amplitudes (to the spikes in synchronized bursts) is happening in individual frequency components (Figure 2.13).

The general observed trend in the MEA is that power in high frequency (around 1000 Hz) frequency bands increase as the age of the culture increases (Figure 2.11). At the same time, the timing of spikes seem to go from synchronized activity to sporadic / advanced spike timing as seen in Figure 2.13. A comparison of electrode spectrograms in Figures 2.14, 2.15, 2.16 and 2.17 further indicate that the time-frequency behavior of electrodes get more advanced with the age of the culture, and that this happens on most electrodes. From this and earlier analysis, it is proposed that the colored MEA activity figure (Figure 2.12 (a)) appearing multiple times best describes the young dopaminergic MEA culture under periodic stimulation on electrode 74 (Chapter 2.1.3).

The PCA models in the project report did not have a time aspect. In any of the most basic neural codes explained in the Chapter 2.4, the time aspect is crucial in decoding neural information. It was thus proposed to let time pass along the rows (objects) in a preprocessed matrix $\mathbf{X}$ for PCA. Figure 2.18 illustrates how data from a single electrode is preprocessed using the new method, introducing multiple frequency components (frequency bins) for each electrode as new variables.

Additionally, the preprocessed data should be on a form that transforms as many of the neural codes (such as rate coding, temporal coding and population coding) into a form that multivariate methods (PCA and PLS in this case, Chapter 5.1 and 5.2) can analyze. For instance, significant variations in amplitudes are well described in variance and covariance measures, which are cornerstones in PCA and PLS. On the other hand, a variance measure (as in the Appendix Chapter 8.4.5) is the same for any ordering of the sequence of samples, so no temporal/timing aspects are captured. Covariance measures could perhaps capture some timing information in the neural codes, but not in the best way for PCA and PLS since the amplitude covariations resulting from timing information would be very small. But what if spike timing information is converted to amplitude information? The preprocessing methodology in this thesis proposes that individual spike counting in frequency bins in an overlapping sliding window can transform some neural codes into amplitude information in histograms (illustrated in Figure 2.19).

2.6.1 Neural codes in envelope frequency?

In figure 2.13 (a), amplitudes along many of the frequency components (frequency bins) oscillate similar to each other. Hence, coherence exists. The author denotes an oscillation in a frequency as envelope frequency. An envelope for a given frequency component of a signal (for example 1000 Hz) is regarded as the amplitude along that 1000 Hz bin in a spectrogram of the signal. In practical scenario, frequency resolution is not infinite (Appendix Chapter 8.4.10) and using Fourier analysis, a frequency component is approximated as the Fourier decomposed part of the

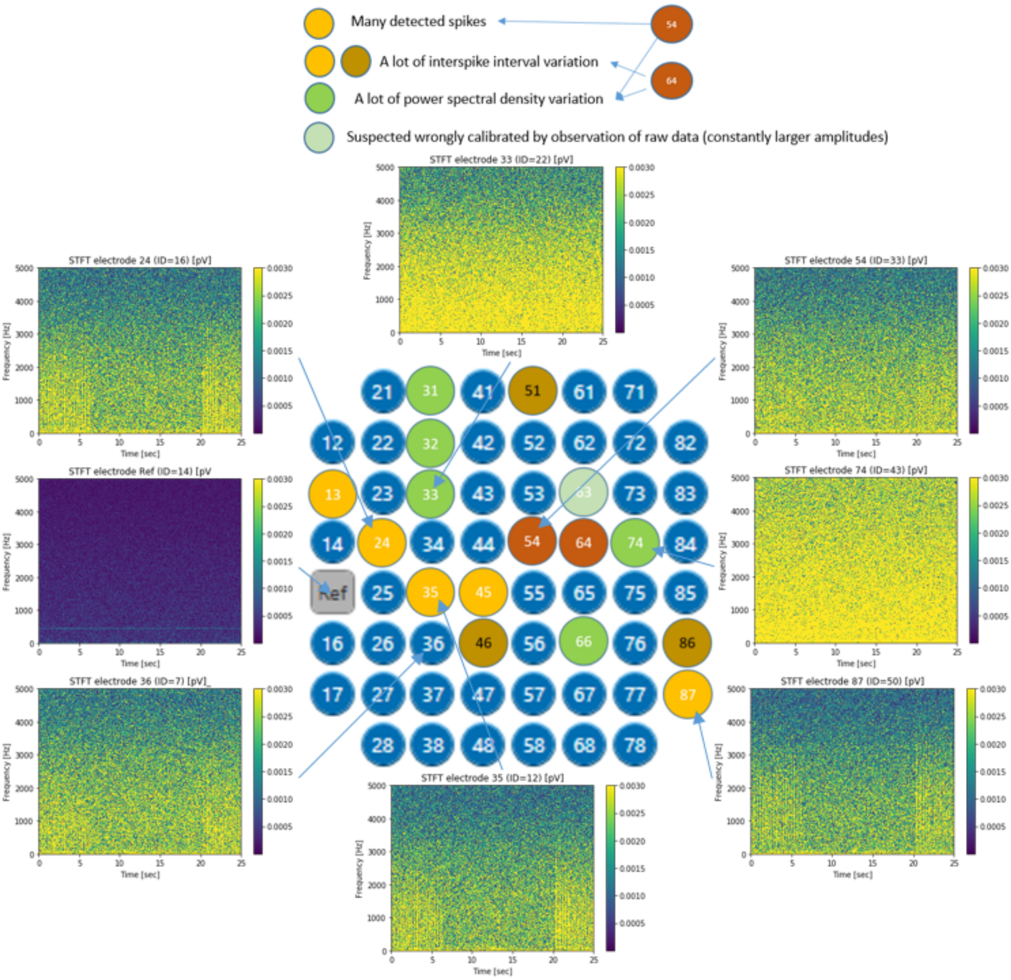


Figure 2.14: The first 25 seconds of raw data of some electrodes of #2 and their connection to the main results of the project work.

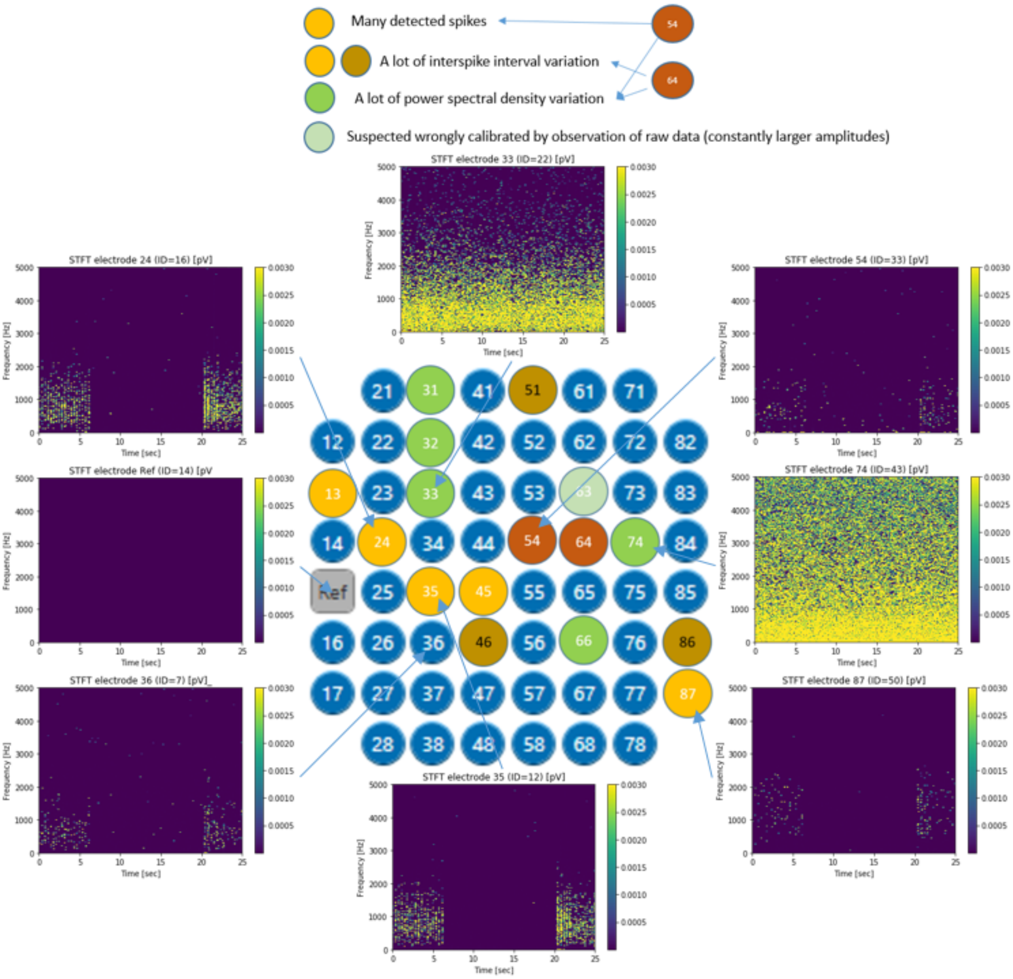


Figure 2.15: The first 25 seconds of noise reduced (new method, Chapter 3) data of some electrodes of #2 and their connection to the main results of the project work.

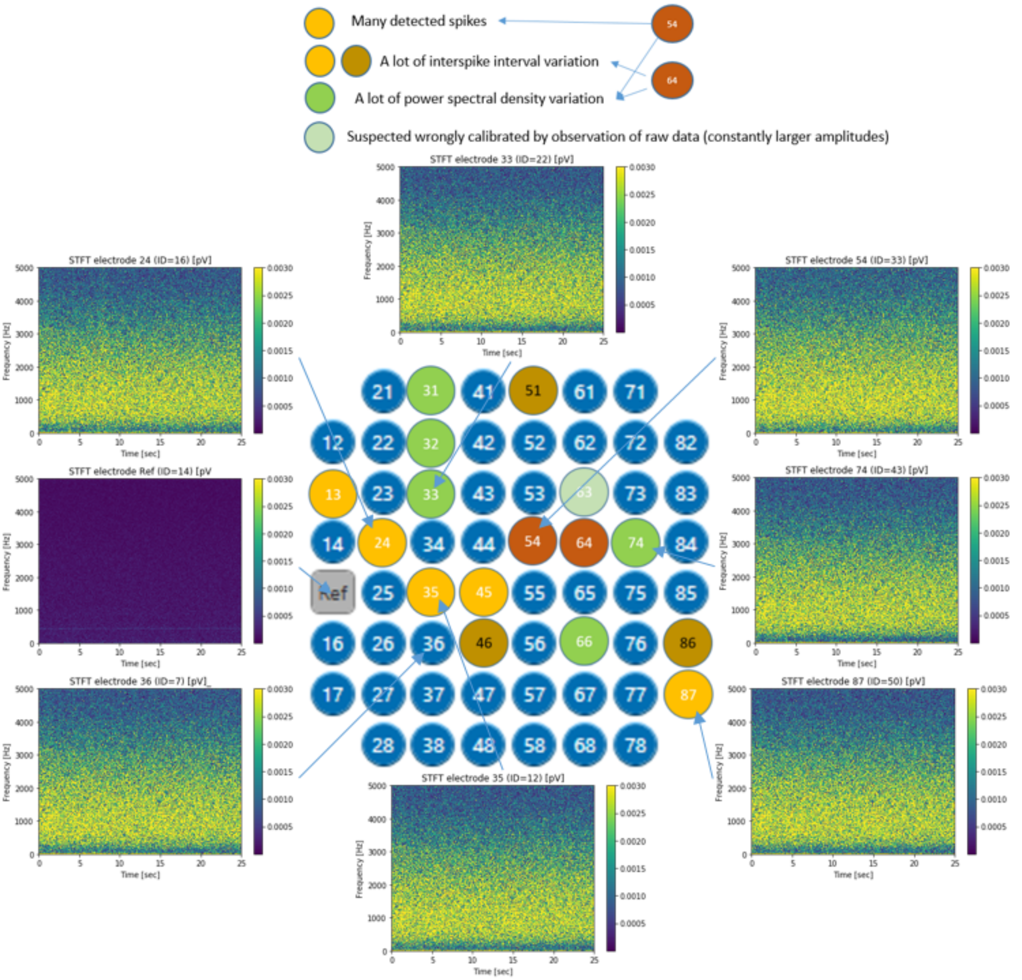


Figure 2.16: The first 25 seconds of raw data of some electrodes of #4 and their connection to the main results of the project work.

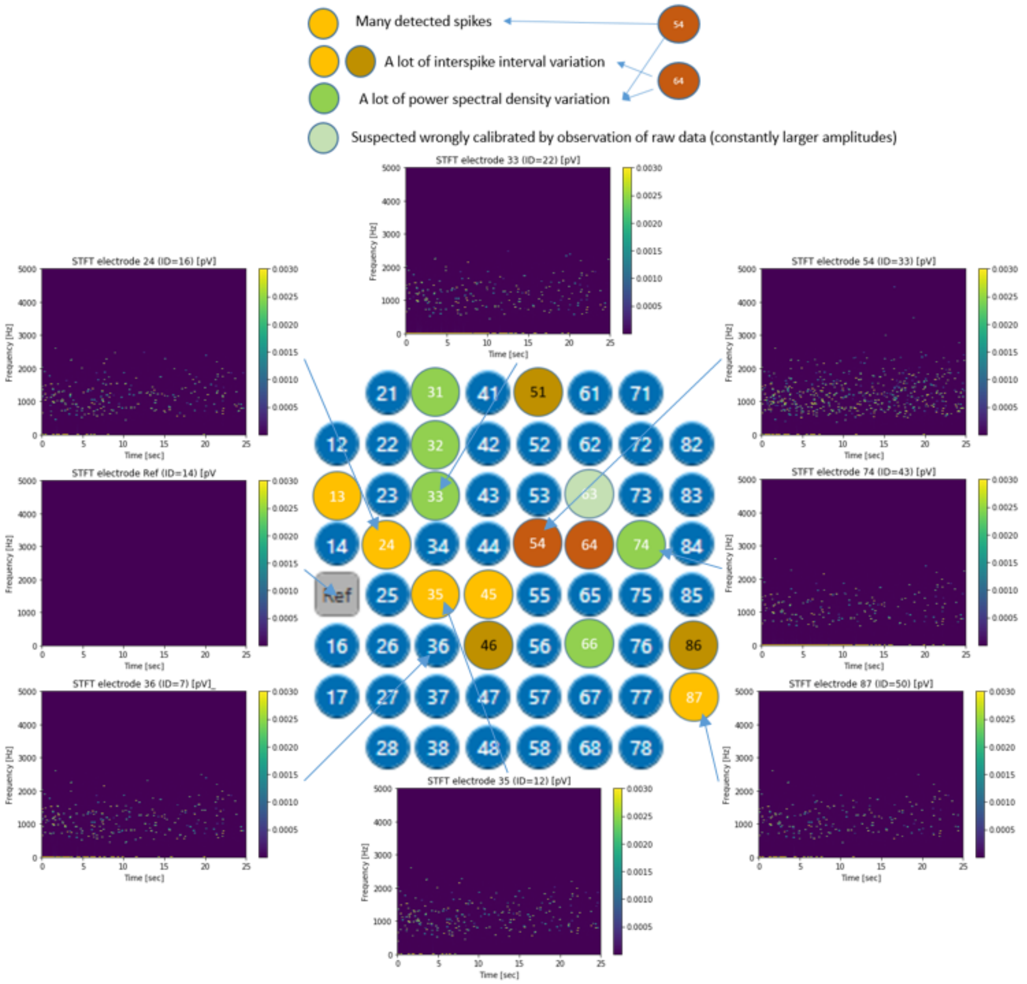


Figure 2.17: The first 25 seconds of noise reduced (new method, Chapter 3) data of some electrodes of #4 and their connection to the main results of the project work.

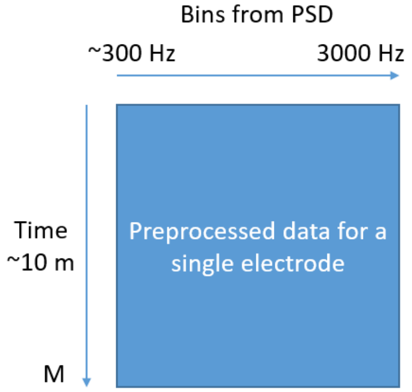


Figure 2.18: Overview of new preprocessing method for data of a single electrode. Frequency bins from PSD in the range $300 - 3000$ Hz as variables and time as objects is the first conceptual step in the preprocessing of a single electrode. The number of rows M in the final preprocessed matrix is a lot smaller than the original number of rows (samples) in the raw data. Because of slightly different parameters in the offline and online preprocessing explained in Chapter 3 and 4, the start frequency is 300 Hz and 296 Hz for the offline and online method respectively. The frequency range was selected based on previous analysis and external work (Langåker, 2017) and in accordance with the fact that SUAs and MUAs are typically detected from signals that are $f > 300$ Hz high pass filtered (Chapter 2.2).

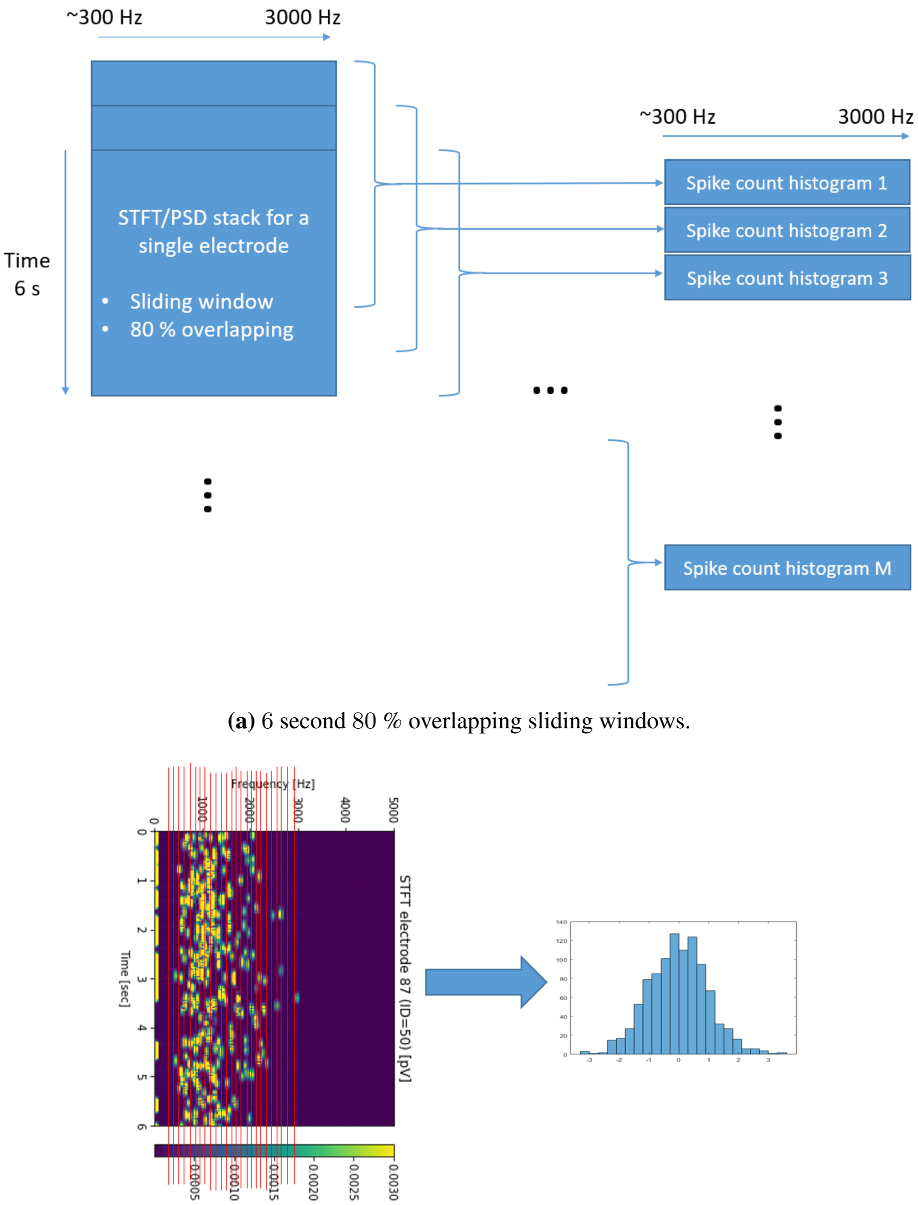
signal with sine waves that are within a small frequency range (for example $1000 - 1010$ Hz). This is why the frequency components in the spectrogram are called frequency bins in this work.

Still referring to Figure 2.13, the author regards each individual occurring mark in a frequency bin as a spike caused by an action potential (AP) in a frequency bin. It is clearly seen that for #2, these APs are synchronized across frequency bins, while in #4 they are more dispersed. It is important to note that it may not be correct to regard these marks as action potentials in this setting. Nevertheless, it is suggested that #4 encodes more information than #2, and thus spectrograms from Fourier analysis can be useful in the preprocessing.

In other words, the envelope frequencies in Figure 2.13 (a) (experiment #2) seem to be more similar to each other than in Figure 2.13 (b) (experiment #4). From raw data of #2, it is observed that most of these envelope frequencies are in fact the same frequency as the oscillation during a burst as in Figure 2.6. However, in experiments of older cultures such as #4, synchronized bursts can typically not be seen in raw data. More interestingly, envelope frequencies vary more. Perhaps information can be taken from the envelope frequencies as a type of neural code? This is the basis for the new preprocessing.

2.7 Data selection

The analysis continues on the same data as in earlier works, as well as on some additional new experiments. They are all MEA experiments on the same dopaminergic culture named MEA2 (Chapter 2.1.1). Table 2.2 and 2.3 give an overview of the subset of data used in the analysis. All



(a) 6 second 80 % overlapping sliding windows.

(b) A 6 second window on actual data (Experiment #4, electrode 87) and illustrated histogram.

Figure 2.19: Aggregation of binned spike counts in 80 % overlapping, 6 second windows for an electrode. A spike in a frequency bin in the Short-Time-Fourier Transform (STFT) of noise reduced data is counted by peak detection (offline method, Chapter 3), or from amplitude threshold exceeds in frequency bins in a vertical stack of Power Spectral Densities (PSDs) from unfiltered raw data (real-time method, Chapter 4). The real-time method uses a precomputed noise threshold per frequency bin, per electrode (Chapter 4.1).

experiments were selected by manual inspection of raw data in MCS Analyzer (MultiChannel-Systems, 2017b) (see Chapter 2.1.3).

A sign of a 30 second interval bi-phasic rectangular stimulation pulse on electrode 74 (Chapter 2.1.3) was manually observed on raw data of all experiments except 2017-06-12T11-40-48 (#4) in Table 2.2. The sign is a 5 millisecond pause of recording on stimulation electrode 74, which results from the fact that electrode 74 does not get recorded in the instant of stimulating. Raw data of experiment 2017-05-22T10-30-07, 2017-06-05T12-49-11 and 2017-06-05T12-55-34 (adult culture) show 30 second interval bi-phasic pulses on recordings of many electrodes. Perhaps in these experiments, the culture mimics (replicate) the external stimulation applied on electrode 74. All experiments in Table 2.3 did not receive any external voltage stimulation.

Manual inspection was needed since the quality on the data varies slightly over different experiments. It was necessary to select a subset of the data from which the results of the analysis could be comparable. For example, when comparing the PCA model of an experiment with the PCA model of another experiment, it is very important that the recording conditions of the two experiments were as similar as possible. Data on some electrodes acted as if they were miscalibrated or broken and had very large/small and unrealistic changes in amplitude.

The task of deciding if a constantly larger amplitude on an electrode opposed to the other electrodes in an experiments is a fact of actual distinguishing neural activity or just instrument calibration offset goes beyond the scope of the thesis. However, the problem was minimized by selecting experiments where amplitudes were visually as equal as possible on raw data. Lastly, a transient start of voltage onset occurred in some experiments. This also affects the preprocessing and preceding analysis.

Name [yyyy-mm-dd]T[hh-mm-ss]	Duration [mm:ss]	Description of raw data from MCS Analyzer (MultiChannelSystems, 2017b)
2017-03-20T10-02-16 (#2)	07:02	Experiment #2 from project report. Clearly visible bursts on many electrodes (ex. 35). Burst duration around 7 seconds. Clearly larger amplitudes on electrode 63, 66, 44, 33, 32, 31, 43 and 74 (stimulation electrode)). Has a noise profile that has been regarded as standard in this thesis.
2017-04-03T10-16-17	15:30	One clearly visible burst on many electrodes (ex. 35) at the very start of the recording. Some spike patterns can be seen across electrodes. Some very noisy electrodes (ex 74, 66).
2017-05-02T15-18-41 (#3)	10:01	Experiment #3 from project report. Some visible bursts on some electrodes (ex. 35). Small spike patterns can be seen across electrodes. Some electrodes with constantly larger amplitudes (ex 63, 66, 74).
2017-05-15T10-20-04	10:02	Different bursting on some electrodes (ex. 35). Burst duration seem to be around 1 second. Some spike patterns can be seen across electrodes. 63 (very), 66 (very), 74 distinguishable noisy electrodes. 63 and 66 have also a transient voltage onset (decrease) at the start.
2017-05-22T10-30-07	09:59	Not much visible bursting. Clearly visible 30 second positive spikes on almost all electrodes (from stimulation 2.1.3 ?) Small spike patterns can be seen across electrodes. 63, 66, 74 distinguishable noisy electrodes.
2017-06-05T12-49-11	05:03	Clearly visible 30 second positive spikes on almost all electrodes (from stimulation 2.1.3 ?) Small spike patterns can be seen across electrodes. 31 (very noisy, broken?) and 74 distinguishable electrodes.
2017-06-05T12-55-34	05:01	Clearly visible 30 second positive spikes on almost all electrodes (from stimulation 2.1.3 ?) Small spike patterns can be seen across electrodes. 74 distinguishable electrode.
2017-06-12T11-40-48 (#4)	10:01	Experiment #4 from project report. No signs of stimulation. Spike patterns can be seen across electrodes. Clear transient voltage onset (increase) on start of recording for 66, 63, 74

Table 2.2: The final selection of MEA experiments for preprocessing and analysis from old data set (from projec work) on MEA2 (20170110-20170612). Data on young cultures display more synchronized activity in the form of bursts than older cultures. This makes it easier to identify the noise in young cultures than in old ones. It is discussed in Chapter 7 whether visible bursts in raw data is a sign of young culture.

Name [yyyy-mm-dd]T[hh-mm-ss]	Duration [mm:ss]	Description of raw data from MCS Analyzer (MultiChannelSystems, 2017b)
2017-12-11T13-37-28	09:59	Verified no stimulation. No visible bursts (as in earlier recordings). Spike patterns can be seen across electrodes. 75 constantly larger amplitude.
2018-01-03T14-35-44	10:00	Verified no stimulation. No visible bursts (as in earlier recordings). Spike patterns can be seen across electrodes.
2018-01-08T13-40-39	10:00	Verified no stimulation. No visible bursts (as in earlier recordings). Spike patterns can be seen across electrodes.
2018-01-15T09-30-55	10:00	Verified no stimulation. No visible bursts (as in earlier recordings). Spike patterns can be seen across electrodes. 37 constantly larger amplitude.
2018-01-22T11-05-33	10:01	Verified no stimulation. No visible bursts (as in earlier recordings). Spike patterns can be seen across electrodes. 84 very varying amplitude.

Table 2.3: The final selection of MEA experiments for preprocessing and analysis from a new data set on MEA2 (20171206-20180122). There is an approximately five month gap of missing data of MEA2 between the old and new selected data.

Chapter 3

Offline preprocessing

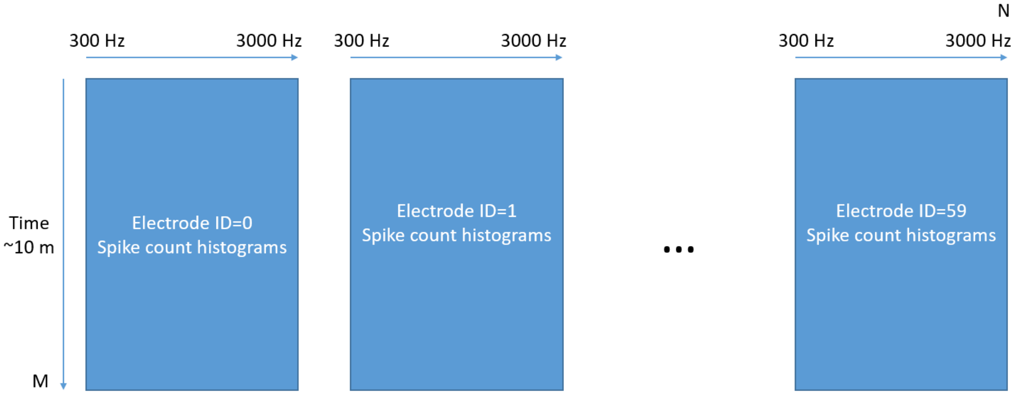


Figure 3.1: Illustration of the preprocessed matrix in the current offline preprocessing.

With data that is recorded with a sampling rate of 10000 Hz, the procedure is as follows

1. **Convert to ASCII**: Select an experiment to preprocess. Preferably by looking at its raw data in MCS Analyzer MultiChannelSystems (2017b). Then, convert the experiment to ASCII (.csv) format using MCS DataManager (MultiChannelSystems, 2017c).

2. **Convert to sound**: Convert the ASCII experiment to separate .wav files, one per electrode, with this Python script.

3. **Noise reduce**: Import all .wav files into the same Audacity project. Determine an electrode with clearly visible segments of noise to use for extracting noise profile with the Noise Reduction plugin. In the thesis, this was electrode 87 from experiment #2. Noise reduce using the plugin with a selected audio segment as a noise profile, then export the individual noise reduced electrode data to separate .wav files.

4. **Peak counting**: Preprocess the noise reduced electrodes to individual and combined .csv files with sliding window peak counts. This Python script takes in the noise reduced .wav files and computes the AP count histograms. For a selected experiment, the final preprocessed data contain one .csv file per electrode, as well as one .csv file that contains all the individual .csv files. The combined .csv file is illustrated in Figure 3.1.

This step-by-step guide is also available in the GitHub README for manualCyborgPreprocessing repository. The spectrograms in Figure 2.13, 2.14, 2.15, 2.16 and 2.17 are from manual noise reduced electrode data using Audacity (Appendix Chapter 8.5.5) in step 2 and 3. The following sections explain the details in each step.

3.1 1. Convert to ASCII

This step is well documented in (MultiChannelSystems, 2017c) and (MultiChannelSystems, 2017b) and requires no selection of parameters.

3.2 2. Convert to sound

The electrode data in the the ASCII file from step 1. is in the form of large positive or negative integers that is supposed to be pico voltage (pV, $10^{-12}V$). The first of two Python scripts does the following to the data of an experiment to convert electrode recordings to individual sound files:

```
$min\_list = empty$
$max\_list = empty$
**for** each electrode data, $d$ **do**
    Append $min(d)$ to $min\_list$
    Append $max(d)$ to $max\_list$
**end for**
$global\_min = median(min\_list)$
$global\_max = median(max\_list)$
$r = global\_max - global\_min$
**for** each electrode data, $d$ **do**
    $d\_preprocessed = (d - mean(d)) * (1/r)$
    Save $d\_preprocessed$ to .wav
**end for**
```

Algorithm 1: Pseudocode of the sound conversion algorithm

3.3 3. Noise reduce

This step uses the noise reduction plugin in Audacity. This link provides an explanation of how it works, while this link points to the latest source code. Roughly speaking, it uses the Fast Fourier Transform (FFT) of the signal to attenuate frequency components of the signal that fall below frequency thresholds determined from a noise segment, then it transforms the attenuated signal back to time domain with the inverse FFT. The real-time noise reduction in Chapter 4 is basically an attempt to reproduce the noise reduction in Audacity on streaming data (TCP or file streams).

In this step, all the electrode .wav files are imported in Audacity as separate tracks. The noise reduction is done as follows:

1. Mark a part of an electrode sound track that represent (look like) as much common noise in the electrodes as possible. This was selected to be a segment of electrode 87 from experiment #2 as shown marked in Figure 3.2. Important note: This same noise segment was used to noise reduce all the experiments described in Table 2.2. This is a clear weakness. It might be better to compose a noise signal from noise segments from different electrodes in order to get a better noise profile.

2. Having marked the noise segment, open ”Noise Reduction...” under the ”Effect” dialog. The little window in Figure 3.2 should appear. Click on ”Get Noise Profile” to let the plugin know what it should regard as noise.

3. Close the plugin by clicking on x in top left corner. Then mark all tracks (preferably using ctrl + a). Again, open the noise reduction plugin, then click ”OK” to perform the noise reduction. The parameters used in the noise reduction plugin in this work are specified in Table 3.1.

Setting	Value
Noise reduction (dB)	48
Sensitivity	4.00
Frequency smoothing (bands)	0

Table 3.1: The parameters used on the noise reduction plugin.

4. Export the noise reduced sounds to separate files in a separate folder using ”File” $->$ ”Export” $->$ ”Export Multiple...”.

3.4 4. Peak counting

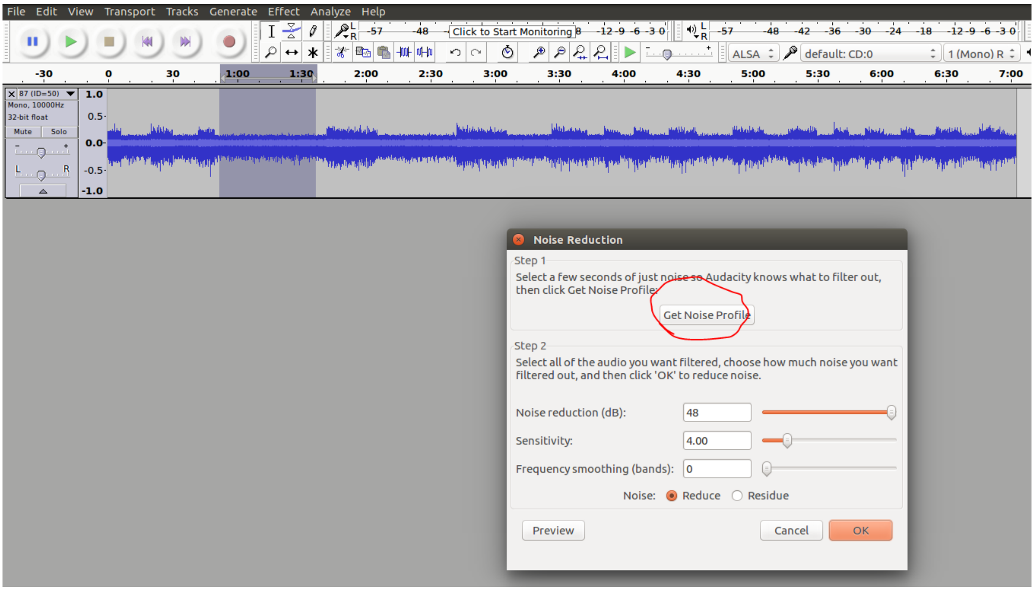
A second Python script counts the spikes in each frequency bin in a 6 second sliding, 80 % overlapping window as illustrated in Figure 2.19 a) . It does this counting sequentially iterating over each noise reduced electrode data. This involves computing Short-Time Fourier Transforms (STFTs) (See Appendix Chapter 8.4.11) of the 6 second sliding windows. For counting spikes, it uses a custom peak detection algorithm working similarly as the findpeaks implementation in MATLAB. The pseudocode of the algorithm, excluding the peak detection algorithm ($detect\_peaks$) displayed in Algorithm 2.

Figure 3.3, 3.4 and 3.5 visualize the peak counting Algorithm 2 for the three first 6 second time windows of electrode $d = 86$ of experiment #4. The storing of max amplitudes for (c) and (d) plots is not part of the algorithm, but included in the figures for interpretation of the data.

From figures 3.3, 3.4 and 3.5, a clearly visible property of electrode 86 on experiment #4 is that

Data set	$peak\_threshold$ value
Table 2.2	0.001
Table 2.3	0.002

Table 3.2: Values of peak_threshold used in Algorithm 2.



Noise Reduction

Step 1

Select a few seconds of just noise so Audacity knows what to filter out, then click Get Noise Profile:

Get Noise Profile

Step 2

Select all of the audio you want filtered, choose how much noise you want filtered out, and then click 'OK' to reduce noise.

Noise reduction (dB): 48

Sensitivity: 4.00

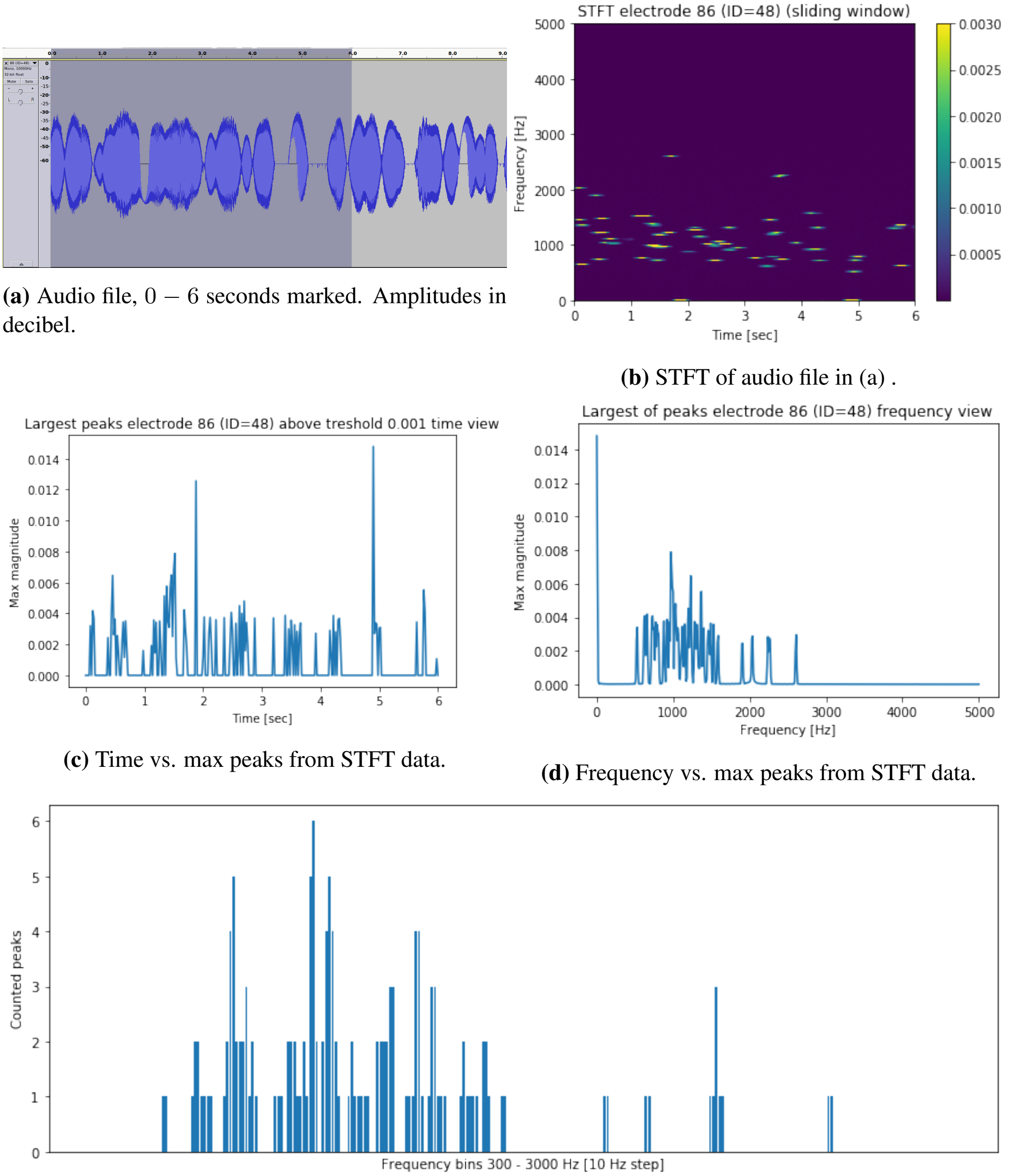
Frequency smoothing (bands): 0

Noise: ● Reduce ○ Residue

Preview Cancel OK

Figure 3.2: Manually selecting the noise segment to be the data between burst nr. 3 and burst nr. 4 of electrode 87 of experiment #2. NB!: Only data from one electrode is shown. In the noise reduction of an experiment, all 60 electrodes/tracks should be imported in the Audacity project. The little window is the noise reduction plugin. After marking a data segment to represent noise, click on ”Get Noise Profile” to let the plugin get the frequency spectrum of this data to represent noise in the noise reduction algorithm.

the very largest peaks seem to be caused by very low frequency components. This was discovered on many electrodes in earlier works as well, and gave the initial motivation to discard very low frequencies in the analysis. In this analysis, only peaks in the frequency range 300 – 3000 Hz of the STFT are counted.



(a) Audio file, 0 − 6 seconds marked. Amplitudes in decibel.

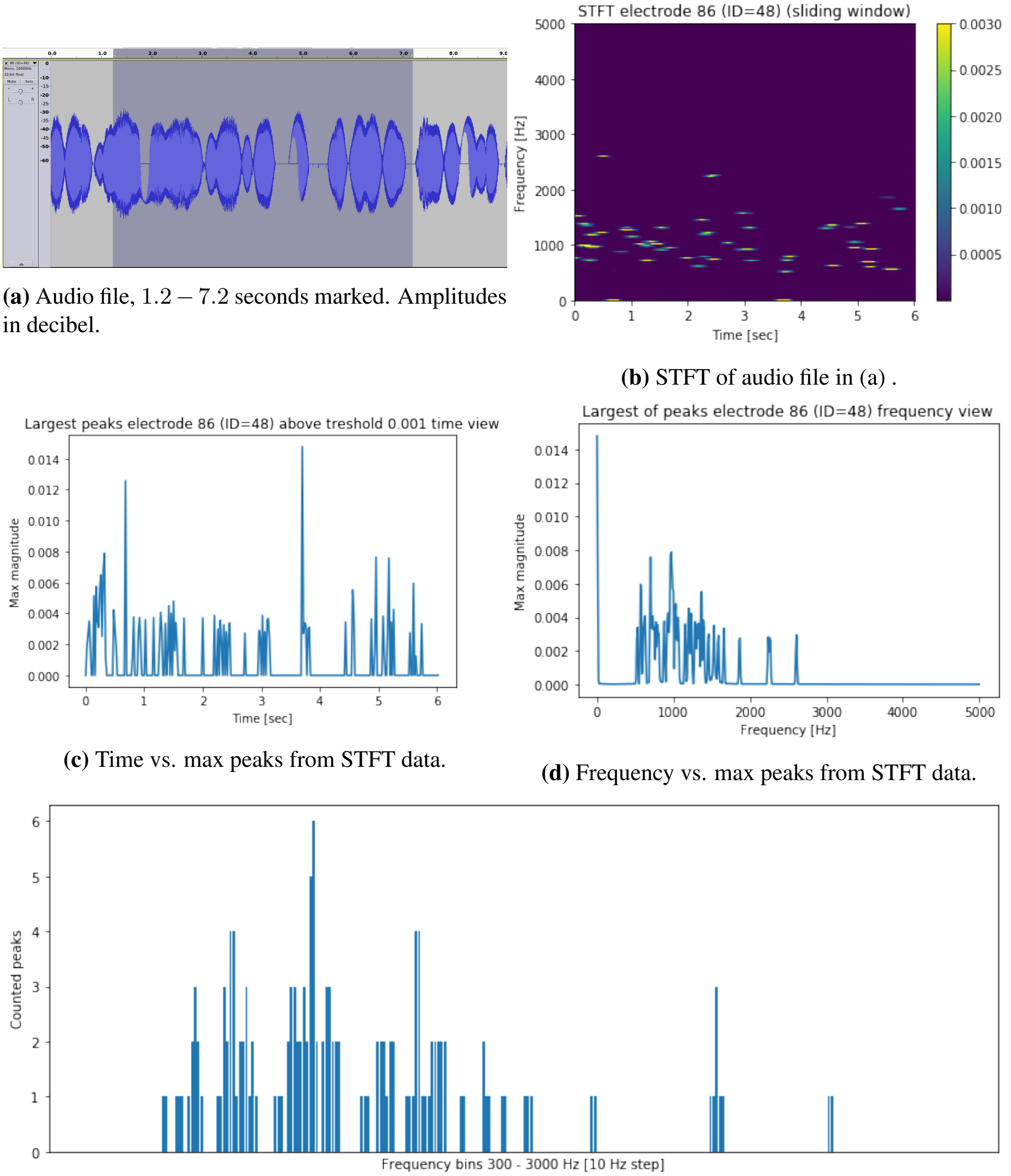
(b) STFT of audio file in (a) .

(c) Time vs. max peaks from STFT data.

(d) Frequency vs. max peaks from STFT data.

(e) The final spike count histogram for this window. The x-axis represent frequency bins with size 10 Hz in the range 300 − 3000 Hz, y-axis number of detected peaks for a frequency bin.

Figure 3.3: Visualization of preprocessing of the first (0 − 6 seconds) window of noise reduced data of electrode 86 from experiment #4.



(a) Audio file, 1.2 − 7.2 seconds marked. Amplitudes in decibel.

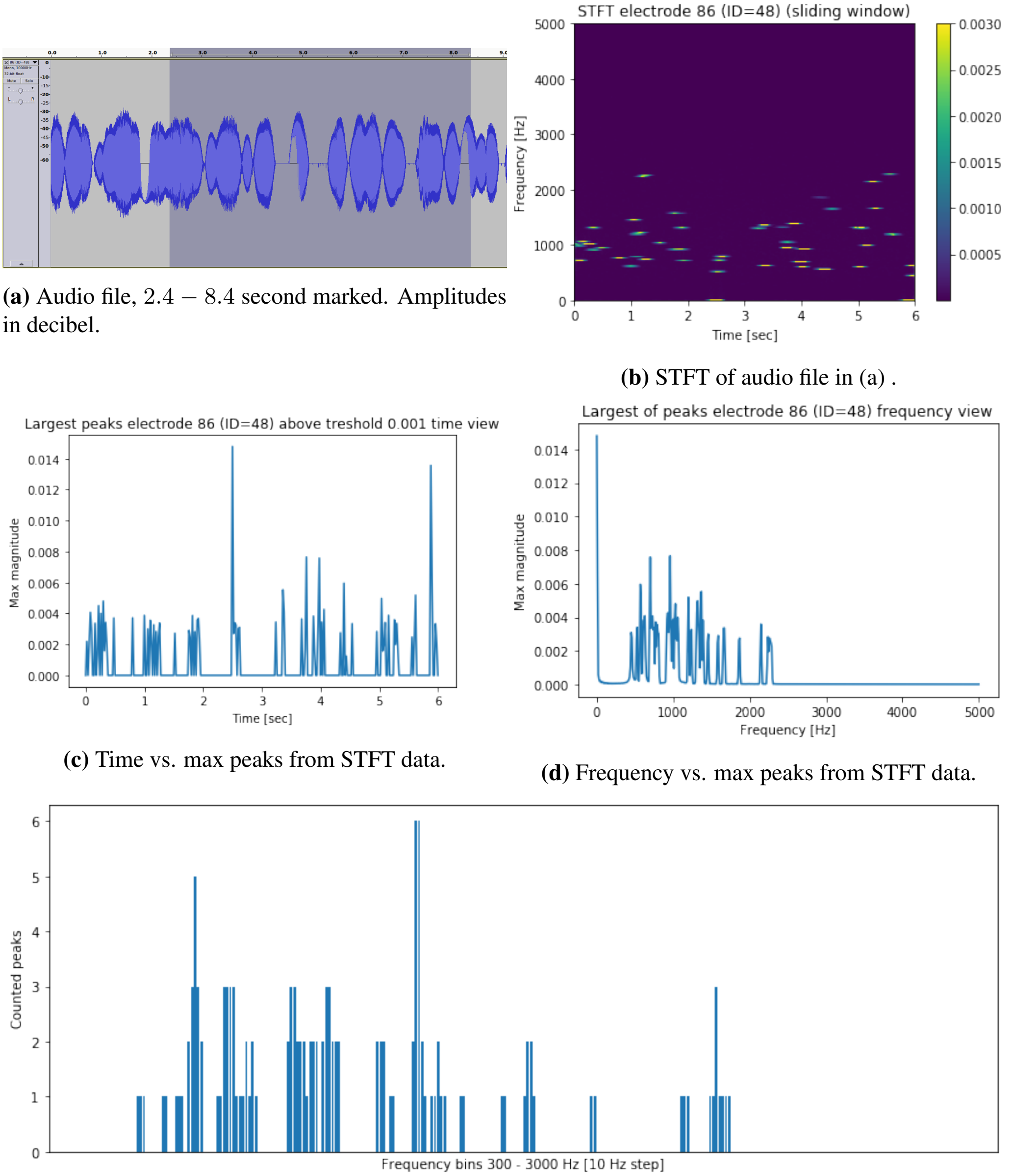
(b) STFT of audio file in (a) .

(c) Time vs. max peaks from STFT data.

(d) Frequency vs. max peaks from STFT data.

(e) The final spike count histogram for this window. The x-axis represent frequency bins with size 10 Hz in the range 300 − 3000 Hz, y-axis number of detected peaks for a frequency bin.

Figure 3.4: Visualization of preprocessing of the second (1.2 − 7.2 seconds) window of noise reduced data of electrode 86 from experiment #4.



(a) Audio file, 2.4 − 8.4 second marked. Amplitudes in decibel.

(b) STFT of audio file in (a) .

(c) Time vs. max peaks from STFT data.

(d) Frequency vs. max peaks from STFT data.

(e) The final spike count histogram for this window. The x-axis represent frequency bins with size 10 Hz in the range 300 − 3000 Hz, y-axis number of detected peaks for a frequency bin.

Figure 3.5: Visualization of preprocessing of the third (2.4 − 8.4 seconds) window of noise reduced data of electrode 86 from experiment #4.

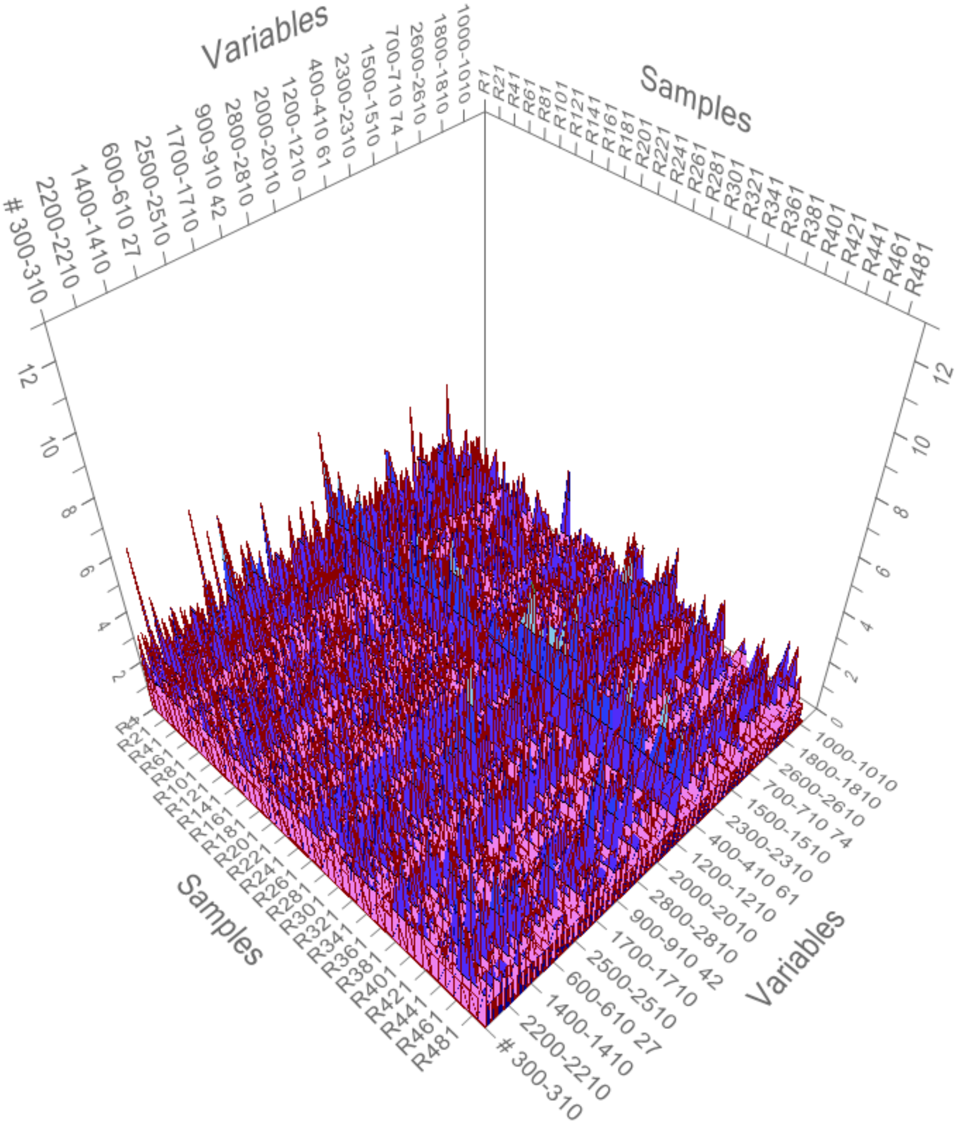


Figure 3.6: All electrodes in #4 preprocessed in the form of Figure 3.1 and visualized. Samples are over time and variables are 10 Hz frequency bins. The reason for 10 Hz frequency bins is because the window size used in STFT was 1000 and the sampling rate of raw data is 10000 Hz, hence by the golden equation (Appendix Chapter 8.4.10) frequency bin width (frequency resolution) $\Delta f = F_s/N = 10000/1000 = 10$ Hz. It is seen that the electrodes have some common aggregate peak behavior. This is later seen to form clusters in the loadings space of PCA models of such preprocessed data (Chapter 6). The clearly larger amplitudes throughout the samples are histograms of electrode 54, which has been previously identified to have constantly larger amplitude than the other electrodes in the raw data (Hovden, 2017) (here leading to constantly higher number of peaks detected, since most are probably larger than $peak\_threshold = 0.001$ from Table 3.2).

```
Allocate empty matrix to hold the spike count histograms for all electrodes, $MEA\_histograms$
for each electrode data, $d$ do
    Allocate empty matrix to hold the spike count histograms for $d$, $d\_histograms$
    for each sliding, 6 second, 80 $\%$ overlapping window of $d$, $w$ do
        $f, t, Zxx = stft(w)$
        Allocate empty vector to hold the spike count histogram for the window $w$, $w\_histogram$
        for each 6 second long frequency bin of $Zxx$, $b$ do
            $numDetectedPeaks = detect\_peaks(b)$ where amplitude of any detected peak $> peak\_threshold$
            add $numDetectedPeaks$ to the correct frequency bin of $w\_histogram$
        end for
        add the complete $w\_histogram$ to $d\_histograms$ (vertical stack)
    end for
    horizontally stack $d\_histograms$ in $MEA\_histograms$
    save $d\_histograms$ for $d$ to .csv file
end for
save $MEA\_histograms$ for all electrodes to .csv file
```

Algorithm 2: Pseudocode of the frequency binned peak counting algorithm. $detect\_peaks(b)$ is the peak detection algorithm and $peak\_threshold$ is to make sure that an amplitude of a detected peak from $peak\_threshold$ is sufficiently large (f. ex. not a peak in noise). Table 3.2 specifies the values used on the data.

Chapter 4

Real-time preprocessing

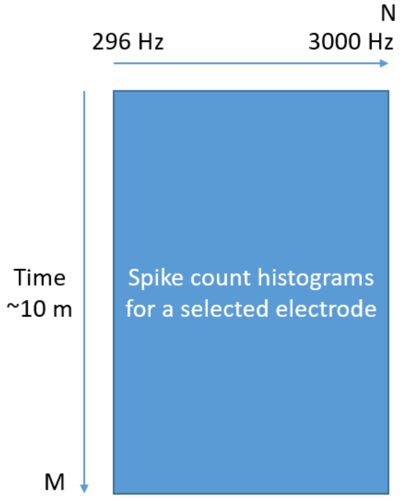


Figure 4.1: Illustration of the preprocessed matrix in the current real-time preprocessing (if the preprocessed PSD stream was saved to a matrix). It can be easily extended to concurrently preprocess all electrode streams, as in the offline case, by using multiple TCP ports (for example 60 for streaming the entire MEA) and running the program in parallel.

After observing interesting results from PCA models using the previously described preprocessing (f. example. Figure 6.9 (a)), the author wanted to obtain similar preprocessing in an automated fashion and thus get closer to make a preprocessing that can be used in a real-time scenario such as in the closed loop robot simulator SHODAN.

The methods presented include two steps, of which the first is only needed to be done one time:

1. **Compute noise thresholds**: Use a recording from a representative culture to determine good noise thresholds that will be used later in **Stream preprocessing**. A representative culture is preferably a culture with clear synchronized bursts in raw recordings on many electrodes, so that intermediate data between bursts can be easily identified as noise.

2. **Stream preprocessing**: Select one of two main types of preprocessing described below as Preprocessing 1 and Preprocessing 2 . Preprocessing 2 is mainly used in the analysis and outputs data as in the offline preprocessing and Figure 2.19.

Only stream preprocessing from a single electrode was implemented due to time constraints and software complexity. With inspiration from the Noise Reduction plugin of Audacity, the work ended up with two distinct preprocessing methods on streaming data of a given electrode:

1. From integer stream of pV to noise reduced PSD stream (**Preprocessing 1**).
2. From integer stream pV to frequency binned AP count stream as in Figure 2.19 (a) (**Preprocessing 2**).

The Scala code for both methods is available in this Github page. The README also contains two examples that sum up the two preprocessing methods, and they are included in Chapter 4.3 and Chapter 4.4. Preprocessing 2 is the default and will run without modifications as of this GitHub commit. It was written with the aim of making it easy to implement into the previously mentioned SHODAN robot simulator etc., which is written in Scala, and thus enable the possibility for using results of multivariate analysis in SHODAN. It must be noted that the program can be run standalone as well, preprocessing either from a file of TCP stream.

Both methods are essentially based on an implementation of streaming PSDs resulting in a live spectrogram. The difference between a spectrogram of PSDs and spectrogram of FFTs (STFT used in offline method) is only a scaling factor in the amplitudes (explained in the Appendix equation 8.15) .

From a 10000 Hz integer pV stream of an electrode:

- **Preprocessing 1** outputs a 40 Hz stream of PSDs that are noise reduced based on precomputed noise tresholds and an Infinite Impulse Response (IIR) low-pass filter. The preprocessed data is quite heavy in terms of storage space (approx. 100 MB for one 7 min recording of an electrode).
- **Preprocessing 2** counts treshold exceeds in the frequency bins based on the same precomputed tresholds, resulting in an 8 Hz stream of spike count histograms as in Figure 2.19 (a) . This results in lightweight data (approx. 500 KB for one 7 min recording of an electrode).

All the methods in this chapter are strongly inspired by how the Audacity Noise Reduction plugin works (info here, here and source code here). However, the noise reduction algorithm in Audacity can not be used on streaming / on-the-fly incoming data. The main point of the real-time preprocessing is that it can be used on streaming / on-the-fly incoming data.

4.1 1. Compute noise thresholds

For a given MEA culture and fixed recording preparations, this procedure is a one time step to extract a single PSD per electrode, each containing noise thresholds in each frequency bin in amplitudes. It is assumed that individual noise thresholds per electrode lead to better noise reduction (Preprocessing 1) and perhaps more accurate relative spike counts across electrodes (Preprocessing 2) when compared to the offline preprocessing:

Having an individual threshold PSD per electrode circumvents the problem of varying measurement noise over electrodes, assuming static measurement noise in a single electrode measurement. The offline preprocessing does not solve the problem of constantly larger measurement noise on some electrodes, which may result in constantly larger number of APs detected.

The procedure has one important assumption about the MEA experiment that is going to be used for extracting noise thresholds:

Clear synchronized bursts should be visible in raw recordings on many electrodes, so that intermediate data between bursts can be easily identified as noise.

The procedure for extracting noise thresholds for all electrodes in an experiment involves use of multiple Python and Scala scripts found on this GitHub site and is the following:

1. Select a MEA experiment where you think you can find good segments of noise, between clearly visible synchronized bursts (even in the raw timeseries data, at least on data from some of the electrodes). In the data analyzed experiment #2 was selected. Figure 4.2 shows visible bursts on some electrodes in experiment #2 in MCS Analyzer.

2. Make one large PCA model of the raw timeseries data. PC1 (or any other by inspection) score should clearly capture the bursts for the procedure to work. Figure 4.4 b) shows PC1 from PCA on #2, while Figure 4.3 (a) shows the singular values from the PCA. Save PC1 score to disk. Code in link: *"1. PCA with SVD on baseline experiment (#2)".ipynb* (Scala, Apache Spark) .

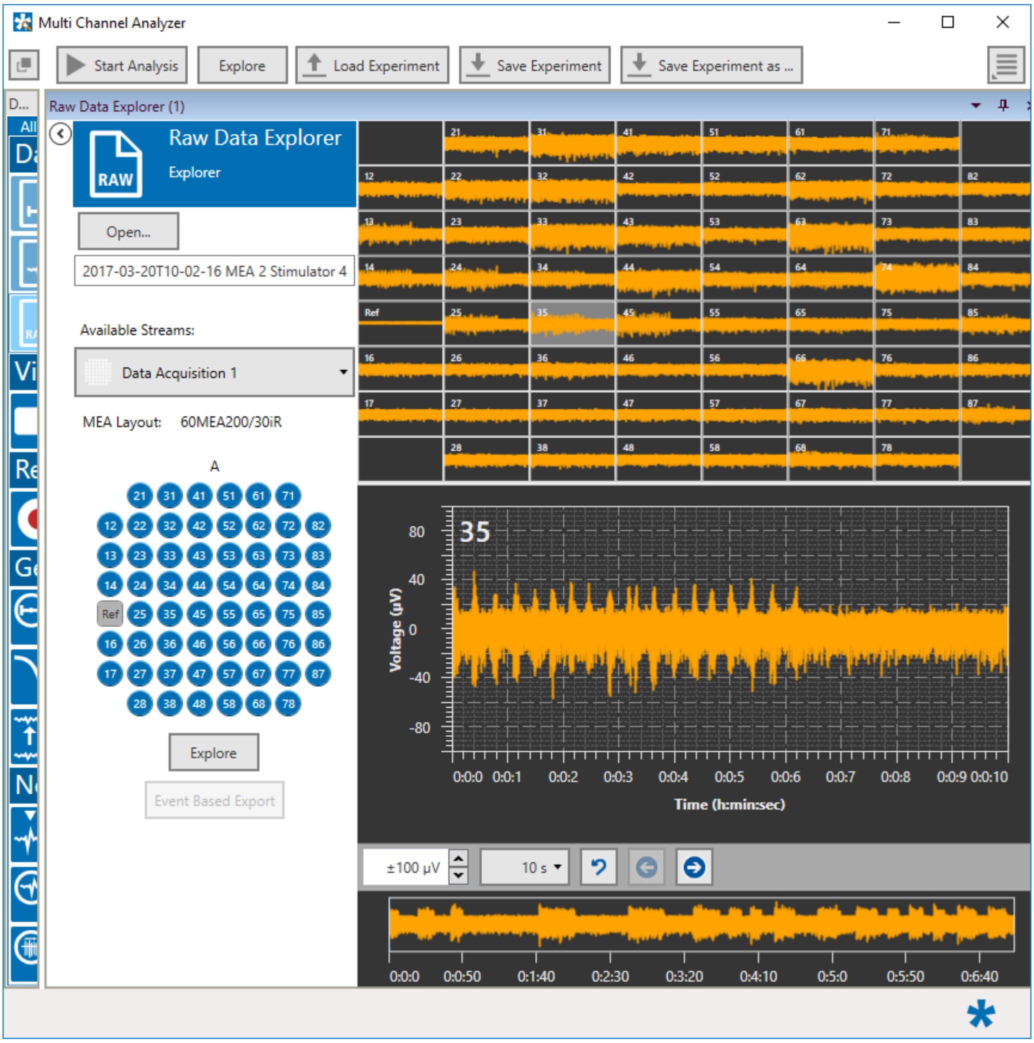
3. Another script extracts the noise segments from the PC1, score, along with an index where the noise sample was found in the PC1 score vector. The idea is to later use these indices to extract per-electrode noise segments from the earlier selected experiment (#2). The script uses a simple treshold ratio (27 % of max value used in the thesis) to detect where it is inside an outside of a burst, hence finding out where noise segments start and end between bursts. The extracted noise segments are then saved to file along with the index of where it found any noise sample in the original PC1 score. NB: It is important to note that the length of PC1 score is the same as the length of any raw electrode data in #2 . This is why the indices can be used to find exact noise profiles for any electrode based on raw time series data from selected experiment (#2 here). Save noise segments to file. Code in link: *"2. Extract noise segments and indices from #2 PC1 score" .ipynb* (Scala).

4. Another two scripts use the indices of noise segments to extract the noise segments for all electrodes (script 1) and create an average or max 1250 segment 80 % sliding window based PSD based on the noise segments for each electrode (script 2). Figure 4.4 c) displays the final selected noise segments of electrode 87, while Figure 4.5 displays noise tresholds of the same electrode computed from the data in Figure 4.4 c). Code in link: *"3. Extract noise segments for each electrode based on #2 PCA PC1 score".ipynb* (Scala) (script 1), *"4. Construct PSD tresholds from noise segments of each electrode (Python)".ipynb* (Python) (script 2). The last script (2) saves the final noise tresholds for all electrodes.

   Noise thresholds in the form of a PSD for a selected electrode can later be used in the stream preprocessing Scala program in the next section, by specifying the path to the noise tresholds .csv file in the variable *noiseTresholds87Path*.

4.2 2. Stream preprocessing

Now that the noise thresholds in the form of a PSD for each electrode is computed, the stream preprocessing will use these thresholds to either

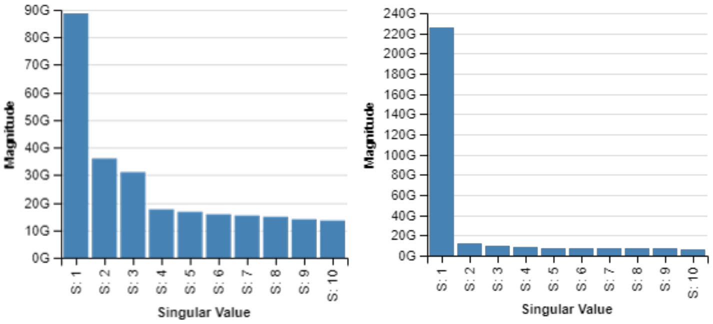


2017-03-20T10-02-16 MEA 2 Stimulator 4

Available Streams: Data Acquisition 1

MEA Layout: 60MEA200/30iR

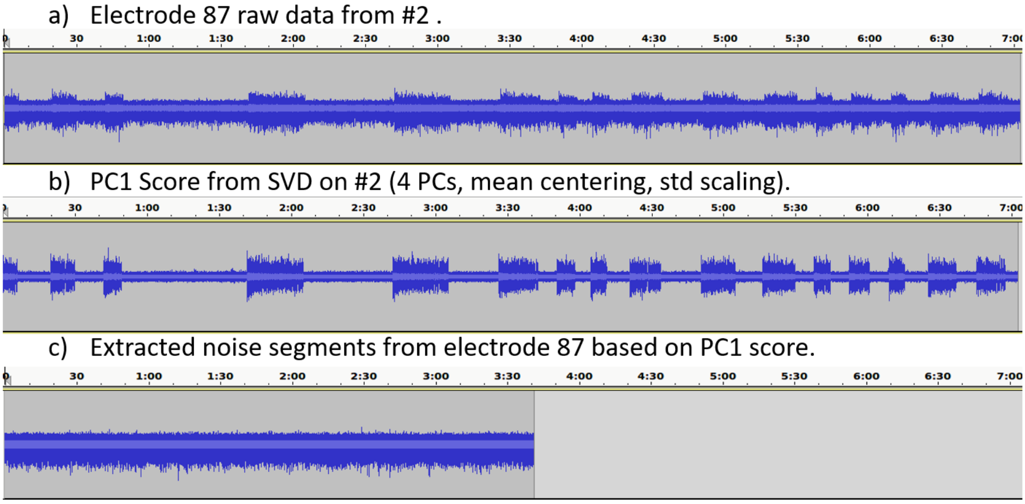
Figure 4.2: Experiment #2 raw data in MCS Analyzer. The lower window shows all 7 minute data for electrode 35, while the rest of the windows show the first 10 seconds.



(a) Experiment #2.

(b) Experiment #4.

Figure 4.3: Singular values from PCA using SVD on raw data of experiment #2 and #4 . Used to extract synchronized bursts. PC1 of experiment #2 was selected, singular values of SVD on #4 is shown for comparison (bad selection of experiment to extract synchronized bursts in a PC). G stands for Giga (10^9). The electrode voltage correlations in #2 are more simple (oscillations) than in #4. Opposed to #2, #4 did not have clear oscillations on raw data. PC1 of #2 clearly expresses bursts (Figure 4.4 b), while #4 do not (expresses 10 second transient voltage onset of electrodes at the start of the recordings).

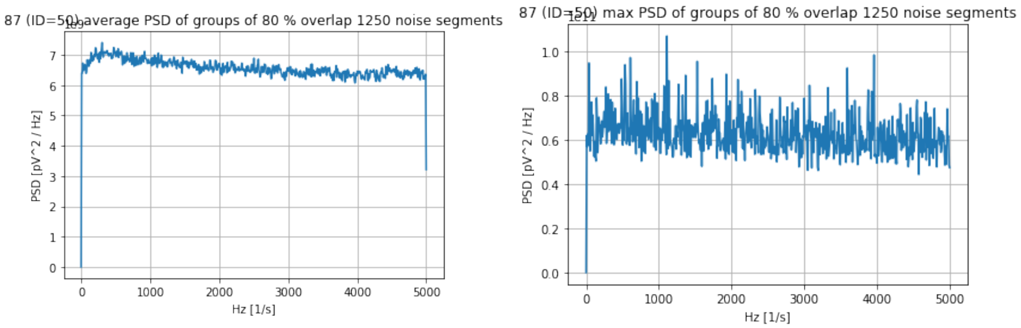


a) Electrode 87 raw data from #2 .

b) PC1 Score from SVD on #2 (4 PCs, mean centering, std scaling).

c) Extracted noise segments from electrode 87 based on PC1 score.

Figure 4.4: Extracting the noise segments of electrode 87 from experiment #2. Array indices of detected noise of b) are used to concatenate data on these indices for any electrode, here data of a), electrode 87. Data converted to sound for visualization in Audacity. Note that because of the sound conversion, the amplitudes of a), b) and c) in this figure are not comparable.



(a) Average PSD noise thresholds of electrode 87.

(b) Max PSD noise thresholds of electrode 87.

Figure 4.5: The final noise thresholds for electrode 87, being based on experiment #2. **(b)**, the version that summarizes the thresholds into the max amplitude bins from PSDs from 1250 length, 80 % overlapping sliding windows (on the extracted noise seen in Figure 4.4 c)), were found to perform best in the stream preprocessing section.

- smoothly attenuate amplitudes in any bin that fall below its threshold with the help of a low pass filter (Preprocessing 1), or
- count the number of times the amplitude in any bin exceeds its threshold in a 6 second, 80 % overlapping sliding window (Preprocessing 2).

Before any of the two tasks could be done, a streaming spectrogram based on PSDs needed to be implemented in Scala.

4.2.1 PSD stream as live spectrogram

The Scala implementation of streaming spectrogram uses the functional streaming library Functional Streams 2 (fs2), as well as an implementation of PSD from the signal processing library scalasignal (Appendix Chapter 8.5.5). A slight difference in scaling of amplitudes in this PSD implementation was corrected in the code to conform to the scaling in the PSD definition in Appendix equation 8.15 and in Scipy's periodogram function with *"density"* as the *scaling* parameter (PSD computation in Python). This was very important for correct computation of noise thresholds (since Python's periodogram was used to compute PSD thresholds that were to be compared with PSDs from the scalasignal library).

The program computes the PSD from each 80 % overlapping 1250 sample long segment in an integer stream implemented with fs2. An fs2 integer stream is regarded as a pipe. Computations on the pipe is done using recursive functions. PhD Peter Aaser consulted in the development of this code, and the instant Q&A forum fs2 on Gitter.im was used in some troubleshooting. Since the final preprocessing (Preprocessing 2) is comparable to the offline case as in Figure 2.19 (a) and doensn't introduce a new way of preprocessing, it is referenced the the source code for further implementation specific details.

- Raw data has a sample rate 10000 Hz . This means that the highest possible frequency in a neuronal voltage captured correctly is 5000 Hz (optimally Nyquist frequency of the recording, Appendix Chapter 8.16). If even faster oscillations occurred in the culture, they will be misinterpreted as other frequencies as an effect of aliasing. It is possible that the Nyquist frequency is higher than 5000 Hz in this data, and aliasing occurs. For future experiments it is confirmed that SHODAN can sample with a sampling rate of 20000 Hz, which makes it more likely to avoid aliasing issues.

- The window length for PSD was set to 1250 samples. (Note that in the offline preprocessing the window length in STFT was 1000). A window length of 1250 implies a frequency resolution in PSDs of $\Delta f = F_s/N = 10000/1250 = 8$ Hz (Appendix Chapter 8.4.10) (width of a frequency bin in PSD). The main reason to go from 1250 to 1000 window length was to have slightly better frequency resolution.
- The windows of 1250 samples overlaps with 80 %. This means that a new PSD is created for each 250 new samples of data. This results in a PSD stream of $10000/250 = 40$ Hz.

  Why overlapping windows? It is in order to sufficiently capture oscillations in the frequency bins (capture envelope frequency). For example, with 3 Hz oscillatory AP spikes in a frequency bin,

  - with non-overlapping windows, $100 * (1250/10000)/(1/3) = 37.5$ % of the AP is freezed“ inside a single PSD, while
  - with 80% overlapping windows, only $100*((1250*(1-0.8))/10000)/(1/3) = 7.5$ % of the AP is freezed“ inside a single PSD. Overlapping windows make shure that the change in amplitudes in the PSD happens according to the expected time-frequency activity in the culture.

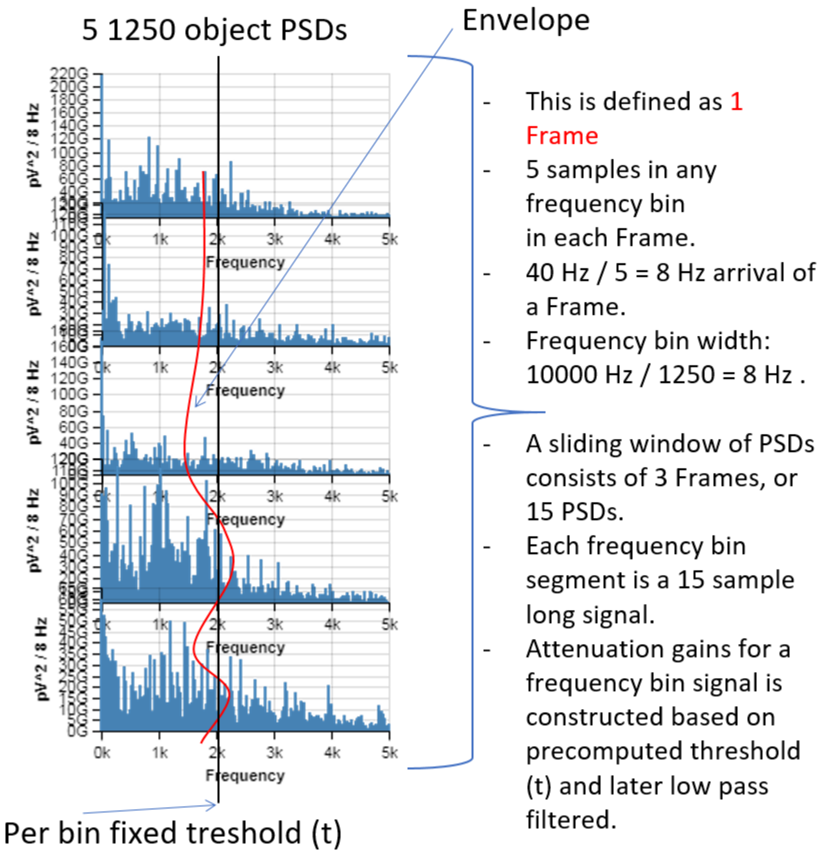
  It is assumed that spiking frequency from oscillatory APs in a time-voltage signal is comparable to envelope frequencies in a time-frequency representation of the voltage signal.

The overall goal of the real-time algorithm is to be able to preprocess an incoming TCP stream of electrode data as was done in the offline case. However, since a live TCP stream from an electrode of a living MEA was not available in the development of the algorithm, the live aspect was faked by reading data from a file instead. The fake part was streaming data for a selected electrode directly from the ASCII .csv file of an existing MEA experiment. In a simulated case, the stream can be throttled to live speed. It was no reason to throttle the stream when prepossessing data from file to file.

4.2.2 Preprocessing 1: Noise reduced PSD stream

This preprocessing noise reduces the 40 Hz PSD stream from Section 4.2.1 based on noise tresholds from Section 4.1. The algorithm uses an IIR low pass filter to smoothly attenuate frequency bin segments. An IIR filter was experimentally selected after comparison with different types of filters (FIR and IIR, see Appendix Chapter 8.4.14). Following is a step wise explanation of the algorithm with figures.

1. Buffer up 15 PSDs in a sliding window on the PSDs, overlapping with 10 PSDs. Figure 4.6 shows 5 PSDs of the stream.



- This is defined as 1 Frame
- 5 samples in any frequency bin in each Frame.
- 40 Hz / 5 = 8 Hz arrival of a Frame.
- Frequency bin width: 10000 Hz / 1250 = 8 Hz .

- A sliding window of PSDs consists of 3 Frames, or 15 PSDs.
- Each frequency bin segment is a 15 sample long signal.
- Attenuation gains for a frequency bin signal is constructed based on precomputed threshold (t) and later low pass filtered.

Figure 4.6: Illustration of 5 consecutive PSDs in the PSD. A sliding window of 15 PSDs, overlapping with 10 PSDs is the start of the noise reduction.

2. In each buffer of 15 PSDs, for each 15 sample long frequency bin segment, noise reduce the segment. Only output noise reduced data from the middle 5 PSDs. Because of a sliding window step of 5 PSDs, the result is a 40 Hz stream of noise reduced PSDs.

   In order to have smooth noise reduction in the stream, a low pass filter with cutoff frequency 6 Hz is used to construct values that are inversely proportional to the amount of attenuation that is going to be applied. With inspiration from the Audacity Noise reduction plugin, they are called ”attenuation gains”. The procedure for constructing attenuation gains from threshold exceeds is illustrated in Figure 4.7.

   The max attenuation in the program was set to $-48[db]$ as in the offline preprocessing. The following power ratio formula was used to attenuate an amplitude with power $P_{in}[\frac{pV^2}{Hz}]$ to the new power $P_{out}[\frac{pV^2}{Hz}]$ attenuated with a certain db (Kleinfeld, 2018):

$$\frac{P_{out}}{P_{in}}[db] = 10log_{10}(\frac{P_{out}}{P_{in}}) \tag{4.1}$$

   An attenuation gain of 1 is set to imply $0db$ attenuation, while a gain of 0 is set to the maximum $-48[db]$ attenuation. For each frame with calculated attenuation gains, the actual amplitude segment corresponding to the frame was attenuated using the following modification of formula 4.1

$$P_{out,i} = P_{in,i} * 10^{(1-aGain_i)*\frac{-48}{10}} \tag{4.2}$$

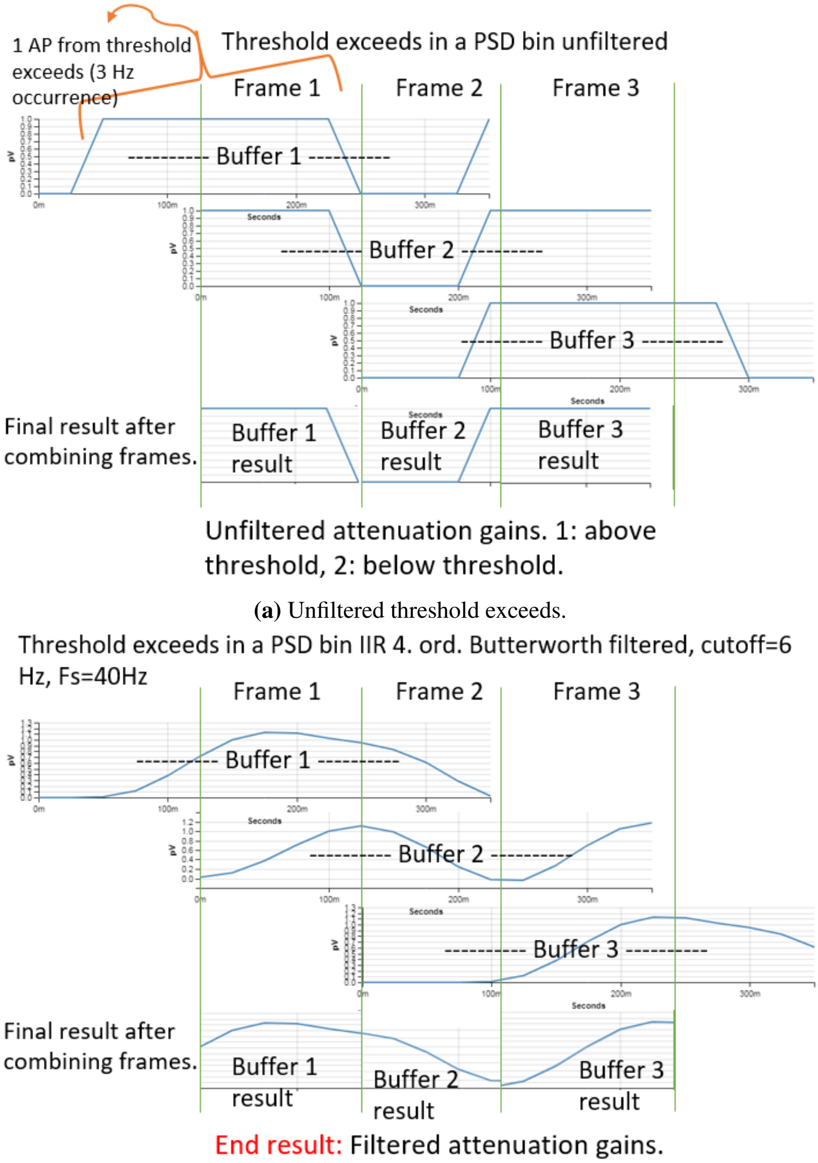
   where $P_{in,i}$ is the original amplitude at position i in the amplitude segment of a frequency bin, and $aGain_i$ the corresponding attenuation gain at position i of the attenuation gain segment. The final result is a stream of attenuated PSDs based on individual thresholds per frequency bin.

Best results using a 4th order IIR Butterworth filter - a small filter study

The selection of an appropriate low pass filter was not trivial. It needed to have sufficiently low phase delay, little overshoot and work on 15 sample long data. After comparing some FIR and IIR filters (see Appendix Chapter 8.4.14 for a quick overview of properties and differences) on synthetic burst data, the final selection was a 4. order IIR Butterworth filter.

MATLAB’s Filter Designer Tool (fdatool) was used to design and export filter coefficients. The different filters were then tested in Scala with the scalasignal library and plots plotted in Vegas plotting library (Appendix Chapter 8.5.5). Figure 4.8 show the magnitude response of the filters, while Figure 4.9 and Figure 4.10 and 4.10 show the result of using the different filters on two different test signals.

The first test signal was mainly to test out phase delay and overshooting, while the second signal mimics two synchronized bursts each with 3 Hz oscillations. As described in the Appendix Chapter 8.4.14, IIR filters require less computations (called ”taps”) to achieve a specific frequency response when compared to FIR filters. However, it introduces nonlinear phase delay characteristics. Despite this, it was decided that a 4th order IIR Butterworth filter would fit best for the task.



(a) Unfiltered threshold exceeds.

(b) Low pass filtered threshold exceeds.

Figure 4.7: Making smooth transition values to attenuate the amplitude data in a given frequency bin. For a 15 long segment of frequency bin data, 15 threshold exceeds (value 1 if above threshold, 0 else) are constructed. Only the middle 5 samples of the threshold exceeds are used further. For each Frame, the values to use to attenuate the actual data are the middle 5 threshold exceeds low pass filtered (”End result” in (b)). This seems to lead to smooth low pass filtering across Frames. Three consecutive Frames are shown (three 15 PSD long buffers, overlapping with 10 PSDs). In each Frame, complete noise reduction can be done, which will be explained shortly. For explanation (a) shows the data before low pass filtering.

Small phase delay on the test signals was considered more important than overshooting since 15 sample data is little space for large phase delay.

Overlapping windows of PSD to solve initial state problem or IIR filter

The library used for FIR and IIR filters (scalasignal, Appendix Chapter 8.5.5) did not have an option to change initial state in a filter. The initial state was always set to 0, so low pass filtering in a streaming case could not work with non-overlapping windows without introducing transients. For example, if Frame length were to be equal to buffer length (5 PSDs in this case), the filter would need time to go from the initial state in each output result of filtered buffer. This is illustrated in Figure 4.12.

A solution to the problem was to introduce an additional Frame before and after the output Frame. The additional Frame before the output Frame seems to take care of the transient from initial state of the filter, while the Frame after is a look-ahead Frame that might avoid discontinuities in some cases.

4.2.3 Preprocessing 2: AP count stream

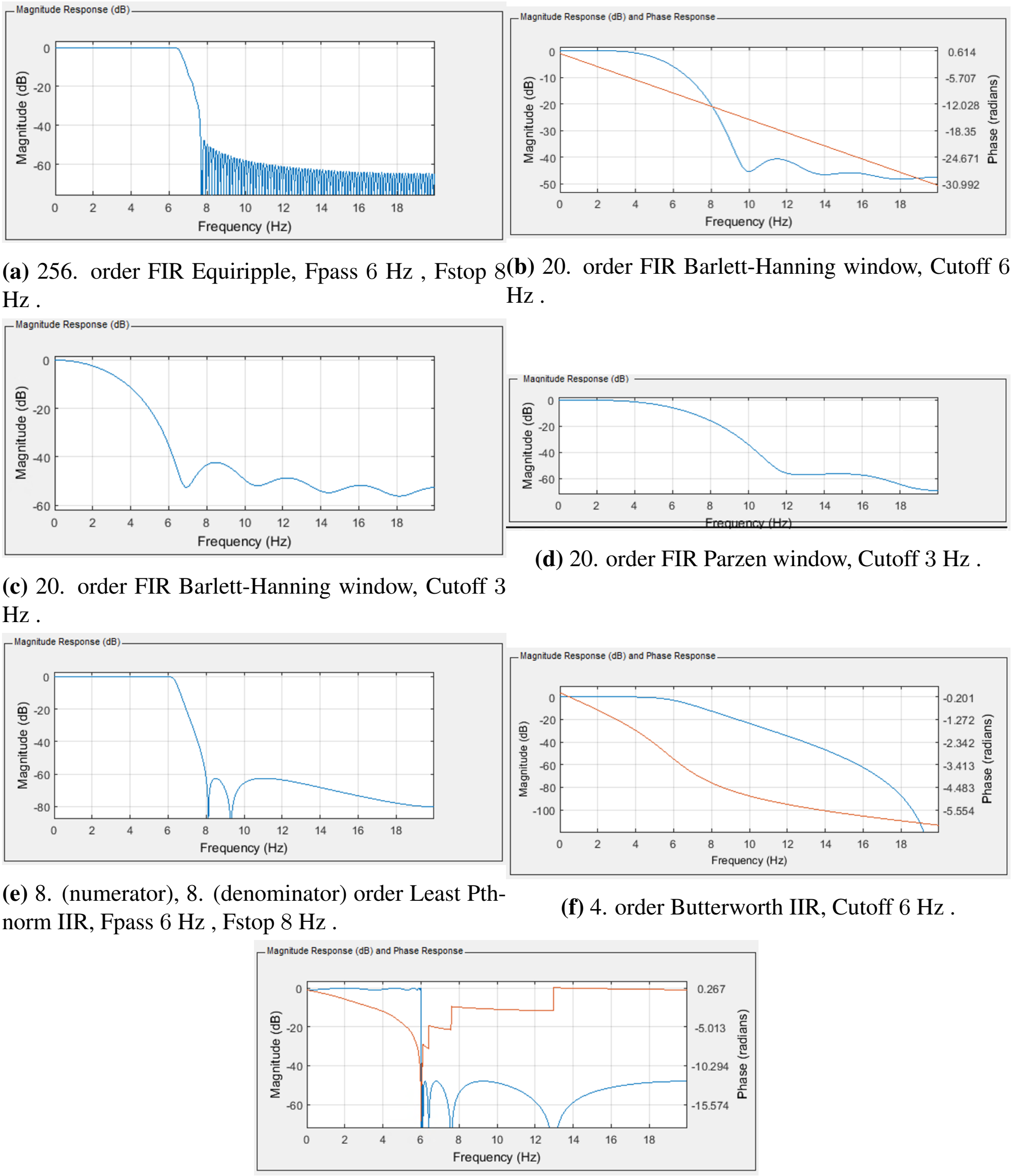
Although a lot of work was spent making Preprocessing 1, Preprocessing 2 is the method used to compare results with the offline preprocessing. The preprocessed data from Preprocessing 2. should be comparable to the preprocessed data of the offline preprocessing in the Chapter 3. The main difference is the following:

While the offline preprocessing uses peak detection to count APs in a frequency bin, the real-time Preprocessing 2. counts each time the amplitude in a given frequency bin exceeds a certain threshold.

The difference leads to more APs being detected in the real-time Preprocessing 2 (One AP as regarded as a mark in Figure 2.13 gets one detection in offline preprocessing, while many, approx. 12 detections in real-time analysis). However, it is postulated that the relative differences in aggregated amplitudes in the preprocessed data are similar in offline and real-time preprocessing.

4.3 Summary example Preprocessing 1: Noise reduced PSD stream of #2, electrode 87

1. Select electrode 87 from MEA2 Dopey experiment 2 (2017-03-20), preferably based on offline analysis (Chapter 3, 6). Raw data is converted to audio for visualization in Figure 4.13 (Chapter 3.2).
2. Extract noise segments, based on PC1 score from PCA of #2 raw data (Chapter 4.1) (Figure 4.14).
3. Construct noise thresholds for the PSD to be used in the real-time preprocessing (Chapter 4.1). Only the frequency range $296 - 3000$ Hz is used later in step 5 (Figure 4.15).



(a) 256. order FIR Equiripple, Fpass 6 Hz , Fstop 8 Hz .

(b) 20. order FIR Barlett-Hanning window, Cutoff 6 Hz .

(c) 20. order FIR Barlett-Hanning window, Cutoff 3 Hz .

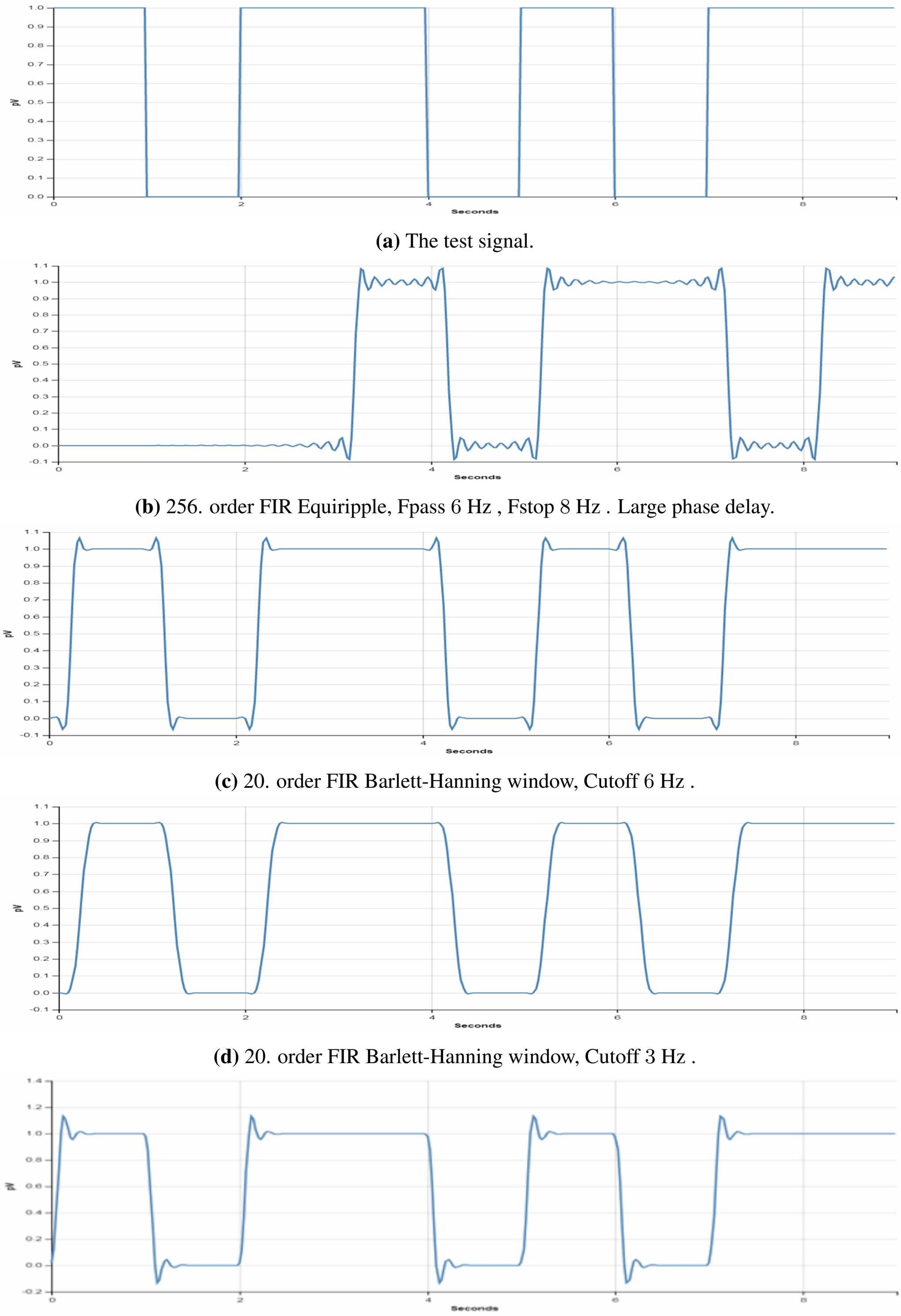
(d) 20. order FIR Parzen window, Cutoff 3 Hz .

(e) 8. (numerator), 8. (denominator) order Least Pth-norm IIR, Fpass 6 Hz , Fstop 8 Hz .

(f) 4. order Butterworth IIR, Cutoff 6 Hz .

(g) 20. order Elliptic IIR, Cutoff 6 Hz .

Figure 4.8: Frequency (magnitude) responses of the FIR and IIR filters tested. Phase response is shown for (b), (f) and (g) as well.



(a) The test signal.

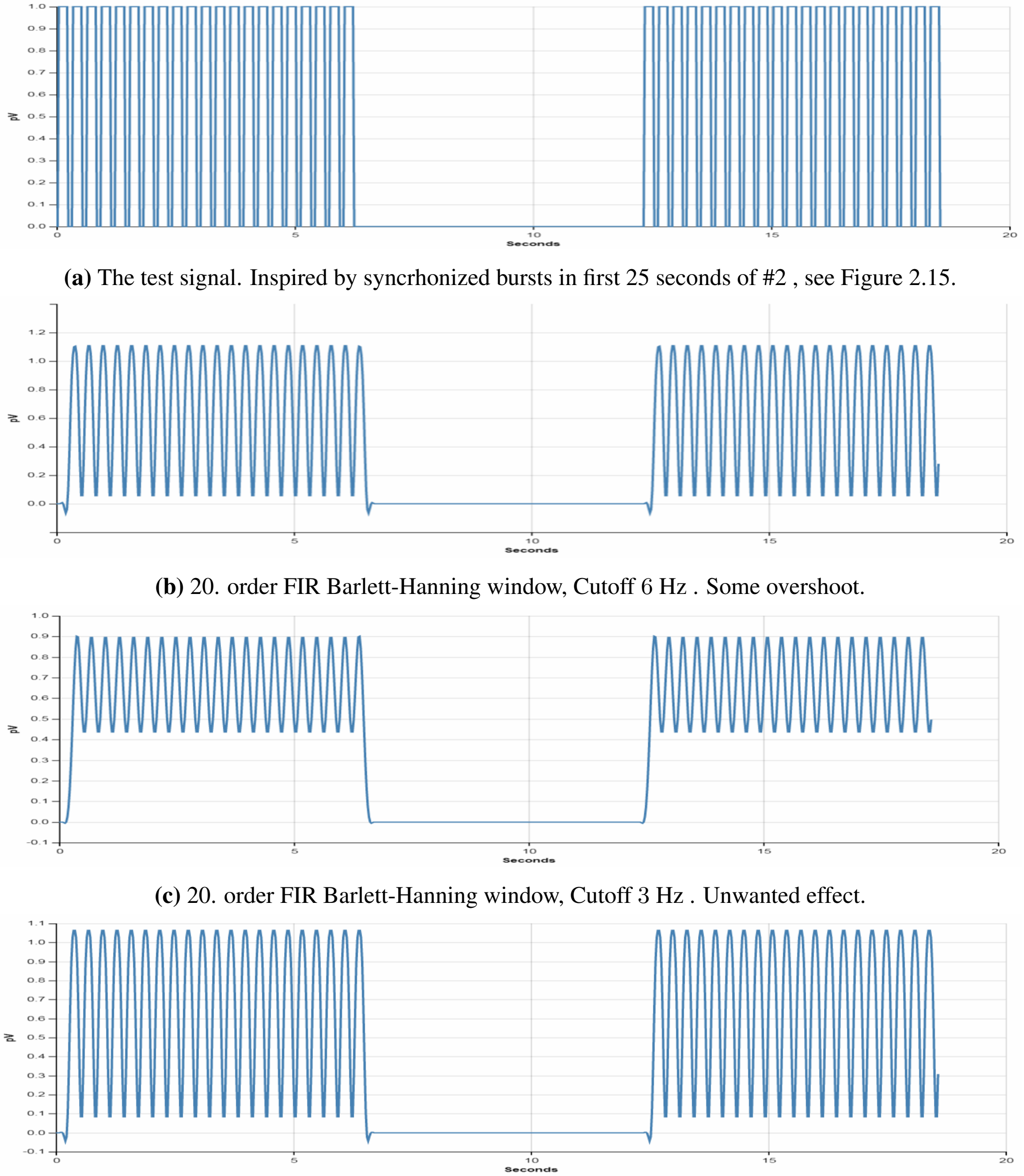
(b) 256. order FIR Equiripple, Fpass 6 Hz , Fstop 8 Hz . Large phase delay.

(c) 20. order FIR Barlett-Hanning window, Cutoff 6 Hz .

(d) 20. order FIR Barlett-Hanning window, Cutoff 3 Hz .

(e) 4. order Butterworth IIR, Cutoff 6 Hz .

Figure 4.9: Artificial test signal 1 (a) on the different filters to test overshoot and phase delay. Approx. 9 seconds long. FIR Parzen, Least Pth-norm IIR and Elliptic IIR were not tested on this signal.



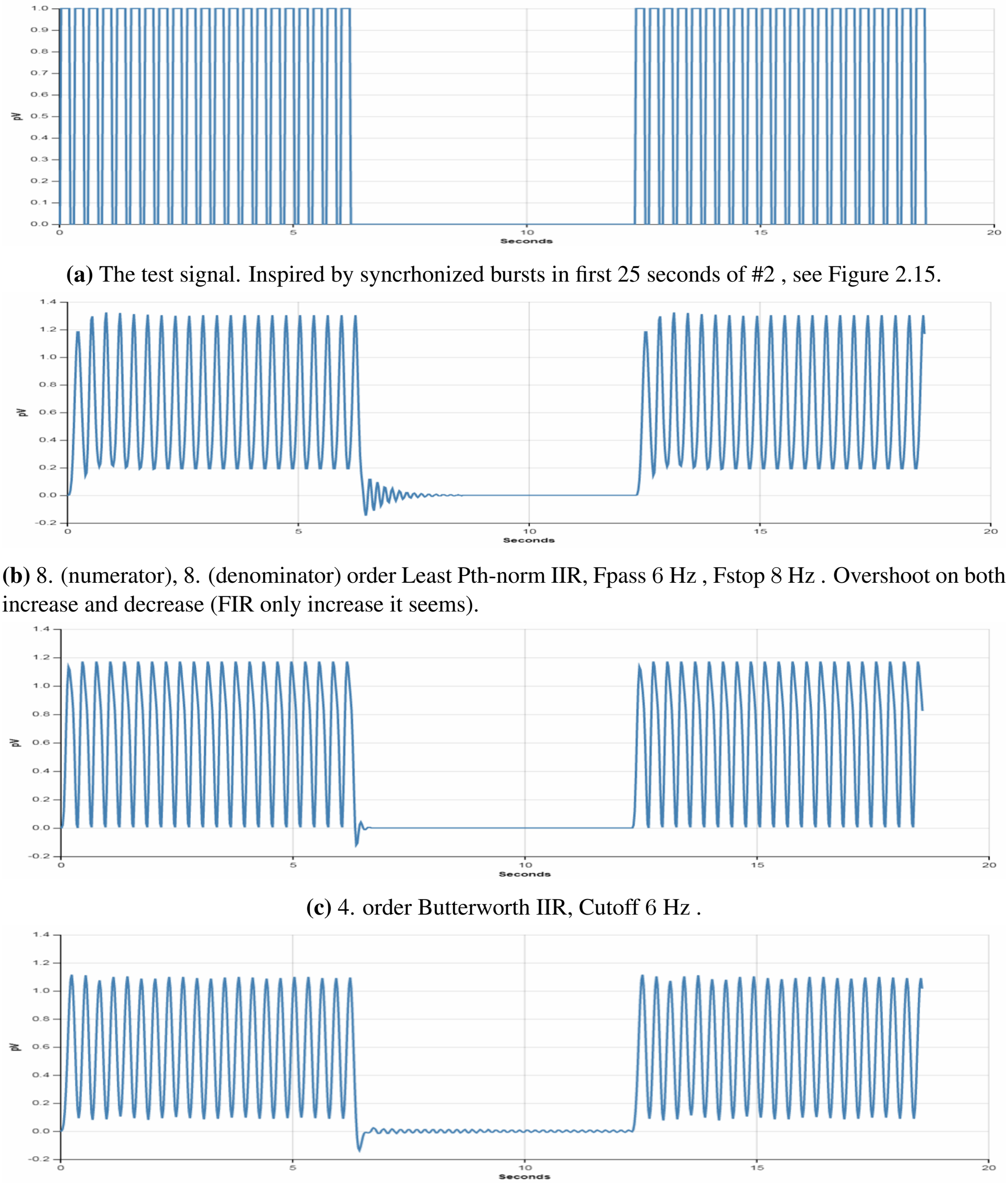
(a) The test signal. Inspired by syncrhonized bursts in first 25 seconds of #2 , see Figure 2.15.

(b) 20. order FIR Barlett-Hanning window, Cutoff 6 Hz . Some overshoot.

(c) 20. order FIR Barlett-Hanning window, Cutoff 3 Hz . Unwanted effect.

(d) 20. order FIR Parzen window, Cutoff 3 Hz . Some overshoot.

Figure 4.10: Artificial test signal 2 (a) on the different filters (**Part 1**) to test two artificial bursts (inspired from the STFT for first 25 seconds of #2 in Figure 2.15). 20 second long signal. The oscillation in a burst is set to 3 Hz .



(a) The test signal. Inspired by syncrhonized bursts in first 25 seconds of #2 , see Figure 2.15.

(b) 8. (numerator), 8. (denominator) order Least Pth-norm IIR, Fpass 6 Hz , Fstop 8 Hz . Overshoot on both increase and decrease (FIR only increase it seems).

(c) 4. order Butterworth IIR, Cutoff 6 Hz .

(d) 20. order Elliptic IIR, Cutoff 6 Hz . Nonlinear behavior, see Figure 4.8 (g).

Figure 4.11: Artificial test signal 2 (a) on the different filters (**Part 2**) to test two artificial bursts ((inspired from the STFT for first 25 seconds of #2 in Figure 2.15). 20 second long signal. The oscillation in a burst is set to 3 Hz .

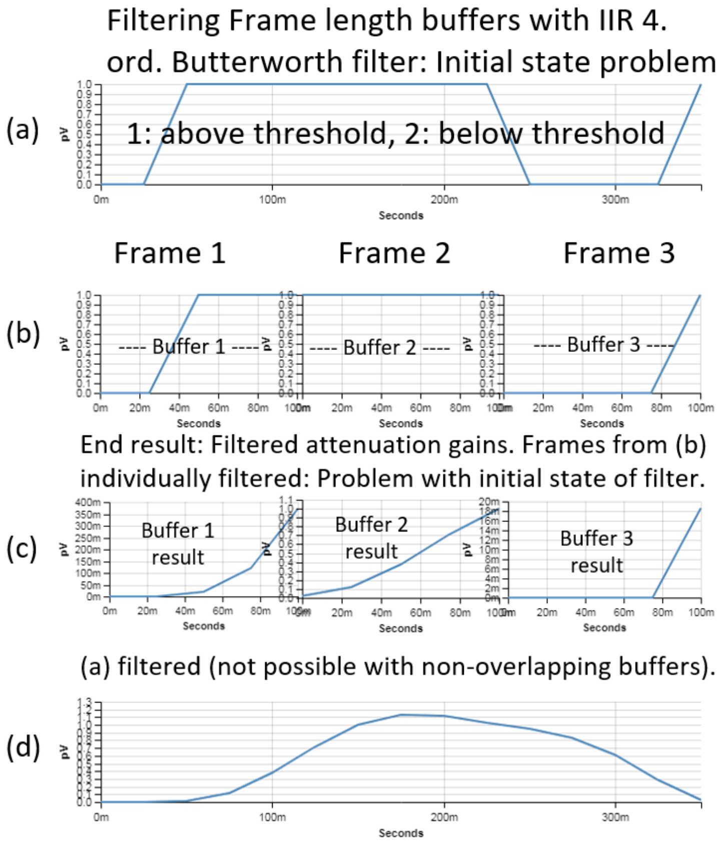


Figure 4.12: The problem with non-overlapping buffer when it is to be low pass filtered. The filter is stateless and always starts at 0. This leads to discontinuties in the end result (c). The solution with overlapping buffer seem to solve the problem on this data, see Figure 4.7 (b).

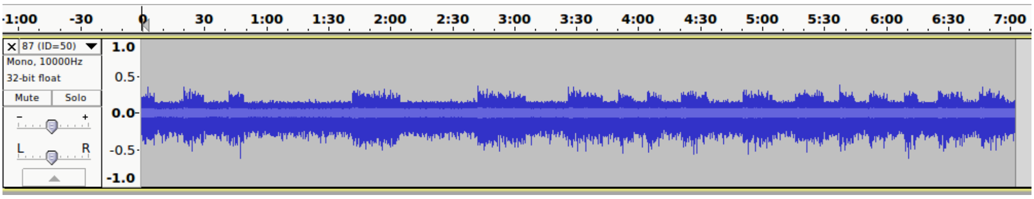


Figure 4.13: #2 electrode 87 raw data converted to sound for this figure.

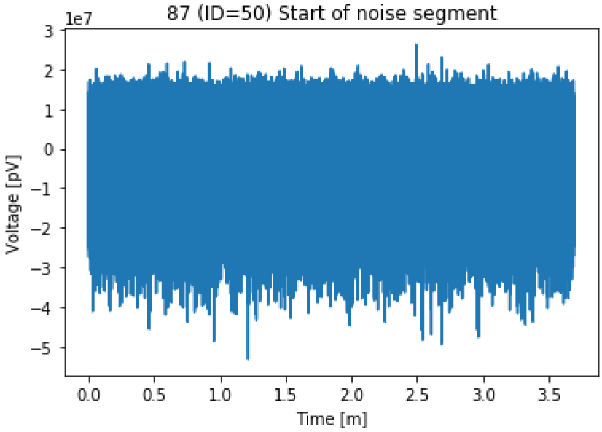


Figure 4.14: #2 electrode 87 combined raw noise segments.

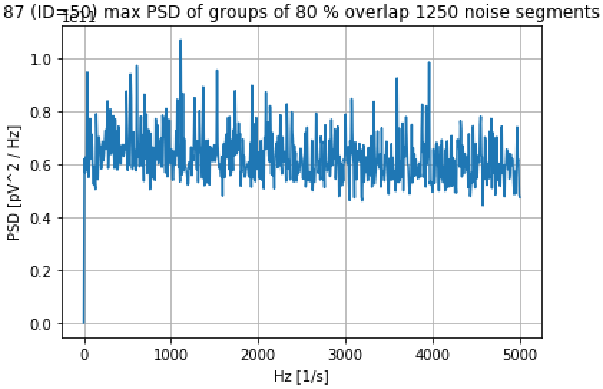


Figure 4.15: The constructed PSD noise thresholds for electrode 87, based on experiment #2, electrode 87 combined raw noise segments (Figure 4.14).

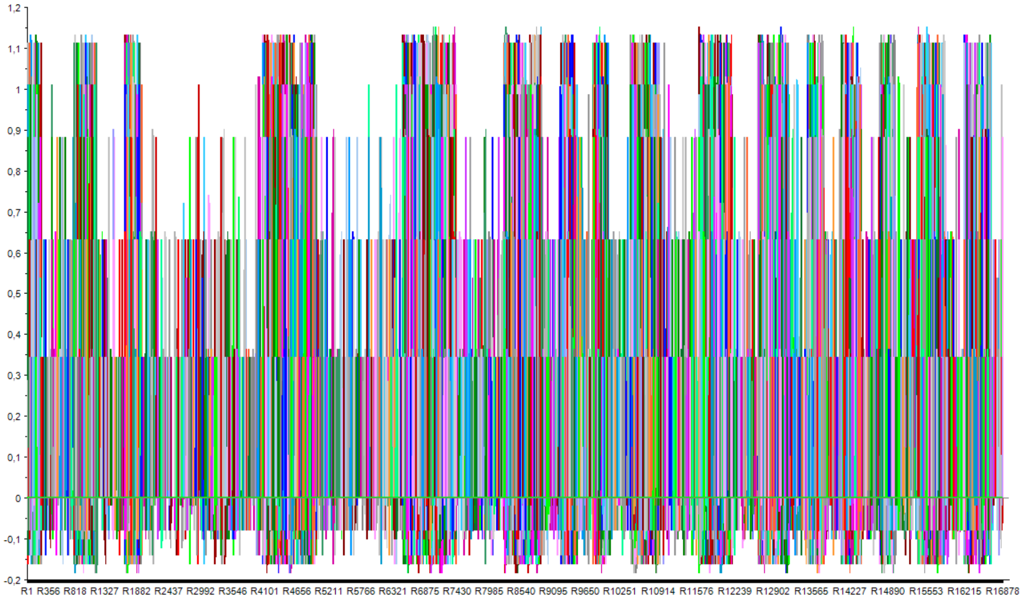


Figure 4.16: #2 electrode 87 all $(3000 - 296)/8 = 338$ frequency bin segments of low pass filtered attenuation gains. These gains will be used to attenuate the data in Figure 4.18 using equation 4.2.

4. Compute PSDs in intermediate steps (Chapter 4.2.1). Line plot of a matrix of computed PSDs along the time axis (approx. 7 minutes) is shown for completeness of the example in Figure 4.18.

5. Use noise thresholds to construct varying decibel attenuation gains over time. Max attenuation is fixed, and set to -48 db in this example. Smooth varying attenuation gains are ensured by using a 4. order IIR Butterworth filter with cutoff 6 Hz (Chapter 4.2.2). The attenuation gains for all bins is displayed in Figure 4.16.

6. Attenuate the raw PSDs (Figure 4.18) with the attenuation gains (Figure 4.16), resulting in a 40 Hz preprocessed PSD stream (Figure 4.17).

4.4 Summary example Preprocessing 2: Spike count histogram stream of #2, electrode 87

Remebering treshold exceeds for last 6 seconds, outputting in intervals of $6 * (1 - 0.8) = 1.2$ seconds (Figure 2.19 (a)). Builds varying frequency-to-above-treshold-counts shapes that is proposed to be good for PCA models. It is an attempt to convert sliding window history of action potentials (timing information) to amplitude information in aggregated shapes. It is proposed that multivariate analysis on this preprocessed data can capture timing patterns for the last 6 seconds in action potentials across different frequencies.

1. Select electrode 87 from MEA2 Dopey experiment 2 (2017-03-20), preferably based on

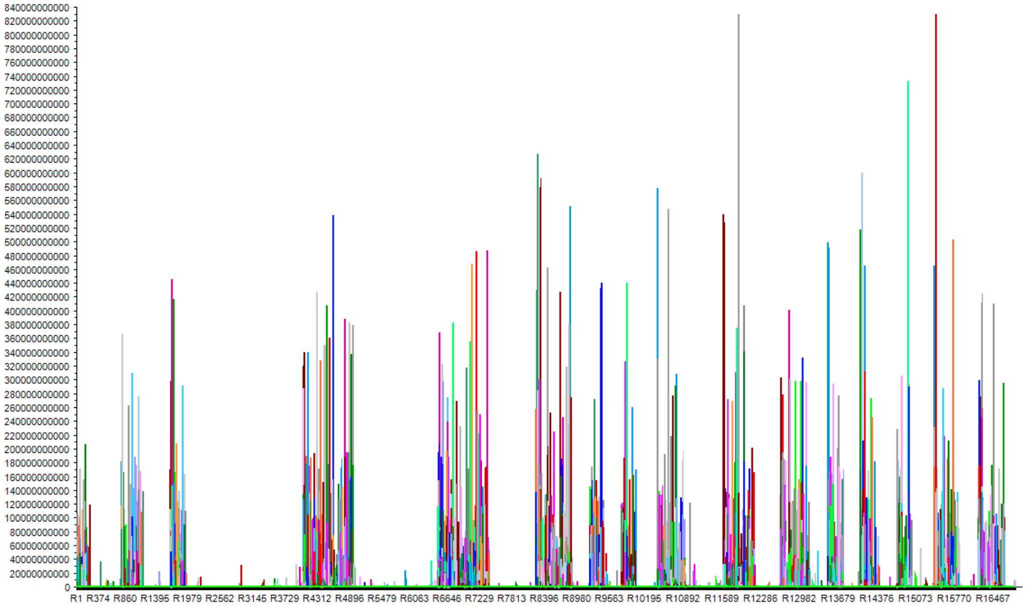


Figure 4.17: #2 electrode 87 final noise reduced PSD stream along time axis.

offline analysis (Chapter 3, 6). Raw data is converted to audio for visualization in Figure 4.13 (Chapter 3.2).

2. Extract noise segments, based on sequences of PC1 score from PCA of #2 raw data (Chapter 4.1) (Figure 4.14).

3. Construct noise thresholds for the PSD to be used in the real-time preprocessing (Chapter 4.1). Only the frequency range $296 - 3000$ Hz is used later in step 5 (Figure 4.15).

4. Compute PSDs in intermediate steps (Chapter 4.2.1). Line plot of a matrix of computed PSDs along the time axis (approx. 7 minutes) is shown for completeness of the example in Figure 4.18.

5. Use noise thresholds to count each time an amplitude in the raw PSD exceeds the threshold in a given bin. This is done in a sliding window of 6 seconds, with 80 % overlap (Figure 2.19). This results in a stream of aggregated counts each 1.2 seconds (0.83 Hz stream). From Figure 4.19, it is seen that burst nr. 4 builds up a larger amplitude because it durated a little bit longer than most of the other bursts. This is the essence in this preprocessing. It is proposed that these effects will be more evident in more adult cultures when comparing APs over different frequency bins.

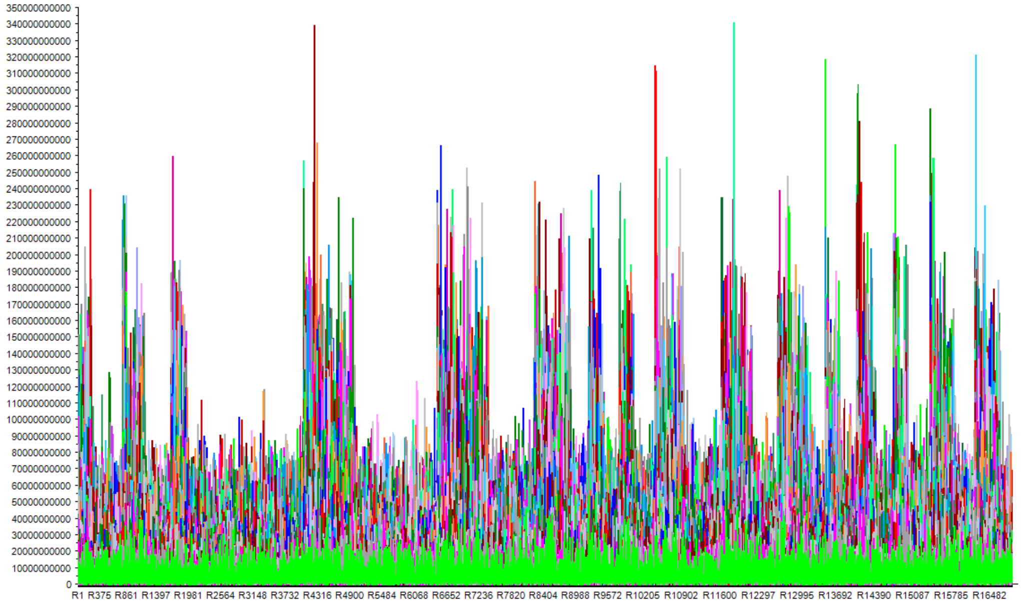


Figure 4.18: #2 electrode 87 PSDs on raw data from time axis. The different colors are the $(3000 - 296)/8 = 338$, 8 Hz frequency bins.

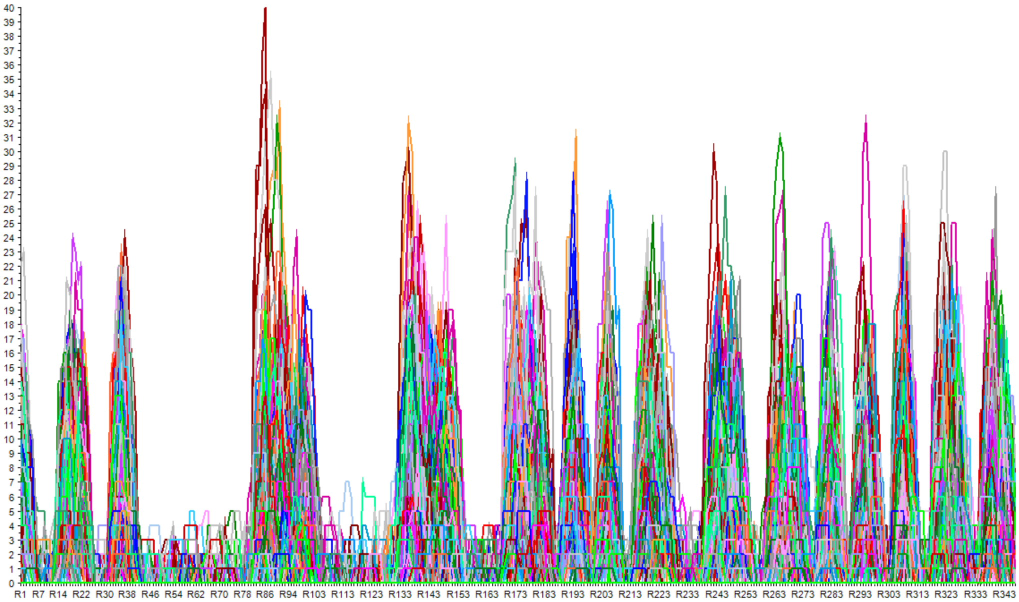


Figure 4.19: #2 electrode 87 counts of threshold exceeds plotted along time axis. A color is the data of threshold counts from 6 second, 80 % overlapping sliding windows for one of the $(3000 - 296)/8 = 338$ frequency bins.

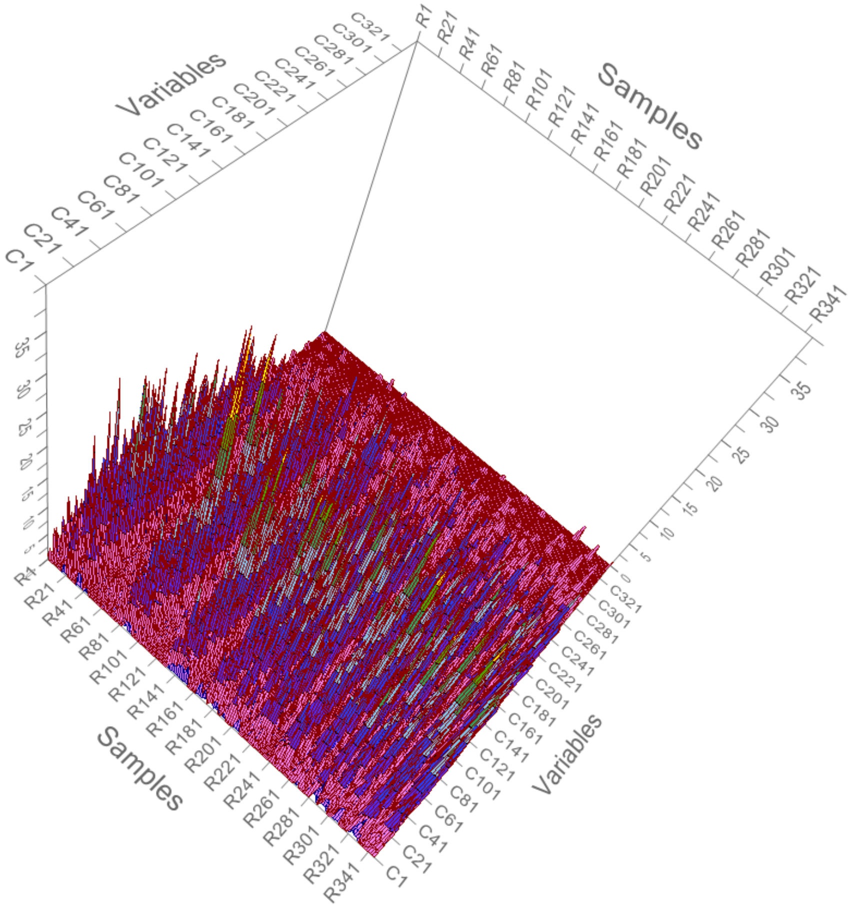


Figure 4.20: Figure 4.19 in 3D.

Chapter 5

Multivariate analysis

This chapter explains the two main MVA methods used, namely Principal Component Analysis (PCA, Chapter 5.1) and Partial Least Squares projections to latent structures (PLS, Chapter 5.2). Simple projection of new samples in score space are explained for the MVA models (Chapter 5.3). Lastly, various metrics and plots used to interpret the MVA models are described (Chapter 5.4 as well as using a complementing variable influence on projection (VIP) analysis for the PLS-DA models in Chapter 5.5).

5.1 Principal Component Analysis (PCA) Model

PCA is a linear dimensionality reduction method that expresses the most important variations in a collection of points in space $\mathbf{X}$ (Pearson, 1901; Hotelling, 1933). This involves spanning the data points in a latent variable space using principal components (PCs). The PCs are linear combinations of the original N variables and transforms each of the M samples/objects into PC space.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{bmatrix} \in \mathbb{R}^{M \times N} \tag{5.1}$$

The method expresses each object in an orthonormal basis, consisting of $Rank(\mathbf{X})$ linearly independent basis vectors which are the principal components (Jolliffe, 2002). The fact that they are linearly independent often means that each object is expressed in a less dimensional vector space. There are no co-variation over the PCs, so each PC captures distinct variational patterns in the data.

The linear combinations can be shown from Singular Value Decomposition (SVD) of $\mathbf{X}$ (Wold, S., Esbensen, K. and Geladi, P., 1987) or from Eigenvalue Decomposition of $\mathbf{X}\mathbf{X}^{\mathbf{T}}$ or $\mathbf{X}^{\mathbf{T}}\mathbf{X}$ (Jolliffe, 2002). PCA can be algorithmically solved in computer programs by direct SVD implementation (Golub and Reinsch, 1970) or other algorithms such as using nonlinear iterative partial least squares (NIPALS), which involves iteratively computing partial least squares estimations of slopes using vector-matrix multiplication (Wold, S., Esbensen, K. and Geladi, P., 1987). If maximum number of PCs are desired, an SVD implementation is preferable. Otherwise if only a the first PCs are necessary, NIPALS is more efficient. NIPALS can also handle missing data / missing observations on variables (Wold, S., Esbensen, K. and Geladi, P., 1987). However, SVD was used in the computation of the PCA models.

A mathematical connection between SVD end eigen decomposition for PCA was previously outlined in the project report (Hovden (2017), Appendix Chapter 6.1.2, page 51). In this thesis PCA is outlined using SVD. Let $\mathbf{X} \in \mathbb{R}^{M\times N}$ be a real matrix with $\mathbf{M} > \mathbf{N}$. Then $\mathbf{X}$ can be written on the form

$$\mathbf{X} = \mathbf{USV^T} \tag{5.2}$$

with a unitary matrix $\mathbf{U} \in \mathbb{R}^{M\times M}$ with columns consisting of left-singular vectors, a diagonal matrix $\mathbf{S} \in \mathbb{R}^{M\times N}$ with non-negative singular values, and another unitary matrix $\mathbf{V} \in \mathbb{R}^{N\times N}$ with right-singular vectors on the columns. The right-singular vectors are the eigenvectors if PCA was shown with eigen decomposition (eigenvectors of eigen decomposition of the coviariance matrix of $\mathbf{X}$, Appendix equation 8.4).

$\mathbf{U}$ and $\mathbf{V}$ have orthogonal columns so that

$$\mathbf{U^TU} = \mathbf{I} \tag{5.3}$$

and

$$\mathbf{V^TV} = \mathbf{I} \tag{5.4}$$

where $\mathbf{I}$ is the identity matrix. Thus, they are orthonormal, and form the orthonormal basis for PCA. The two identity matrices can have different dimensions. Note that in much of the data analyzed have an $\mathbf{X}$ with $M < N$. Since only the first components are used in the analysis, this does not have a negative impact on the SVD used in the PCA models in the analysis. The SVD implementation in The Unscrambler®is closed source, so the author can not exactly specify what the algorithm does in this case. Nevertheless, many software implementations let the $N - M$ last singular values be 0 in this case given that $\mathbf{X}$ has full Rank.

The singular values s_i on the diagonal of $\mathbf{S}$ are related to the eigenvalues λ_i from eigen decomposition of $\mathbf{XX^T}$ or $\mathbf{X^TX}$ according to equation 5.5

$$\sqrt{\lambda_i} = s_i \quad , i \in Rank(\mathbf{X}) \tag{5.5}$$

Additionally, the singular values are sorted in decreasing magnitude along the diagonal of $\mathbf{S}$. This means that the PCs are sorted in decreasing importance in describing the variation.

SVD of $\mathbf{X}$ introduces the concept of scores and loadings, which come directly from its decomposition according to equations 5.6 and 5.7

$$\mathbf{X} = \mathbf{T}_k\mathbf{P}_k^T + \mathbf{E} = Structure + Noise \tag{5.6}$$

where

$$\mathbf{T}_k = \mathbf{U}_k\mathbf{S}_k \quad , \quad \mathbf{P}_k^T = \mathbf{V}_k^T \quad and \quad k <= Rank(\mathbf{X}) \tag{5.7}$$

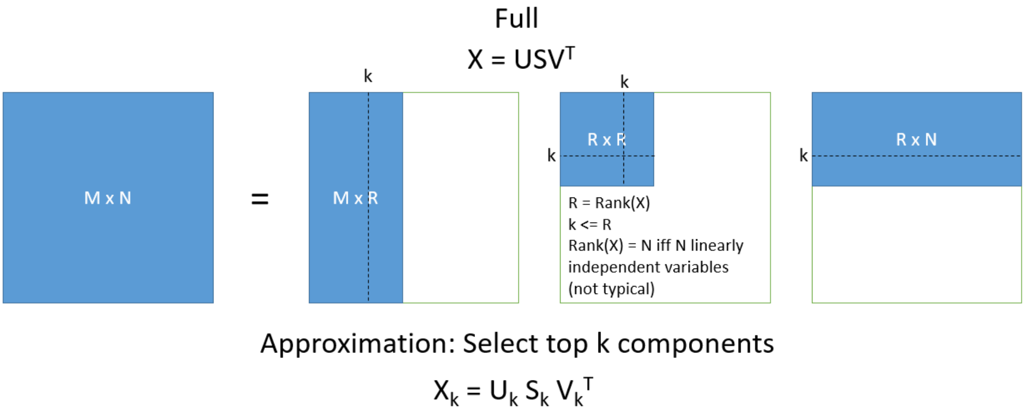

Full
X = USV^T
M x N
=
k
M x R
k
k
R x R
R = Rank(X)
k <= R
Rank(X) = N iff N linearly independent variables (not typical)
k
R x N
Approximation: Select top k components
X_k = U_k S_k V_k^T

Figure 5.1: An illustration of SVD and approximating $\mathbf{X}$ by selecting only first (top) k components/singular values. Typically, at least some of the variables in $\mathbf{X}$ are linearly dependent, hence $Rank(\mathbf{X}) = R <= N$. The white areas are 0 when illustrating SVD in this way.

denoted as scores $\mathbf{T}$, loadings $\mathbf{P}$ and number of selected principal components k (Wold, S., Esbensen, K. and Geladi, P., 1987; Martens, 2016; CAMO, 2017a). For example, by selecting the first $k = 2$ PCs, $\mathbf{T}_k\mathbf{P}_k^T$ is an approximation of $\mathbf{X}$ where only the two largest (most influential) variational patterns are used to reconstruct the matrix. The smaller variational patterns are discarded and modelled as $Noise$. $\mathbf{T}_k\mathbf{P}_k^T$ is the $Structure$ part, while $\mathbf{E}$ is the $Noise$ part. Naturally, the $Noise$ part includes less of the variance in $\mathbf{X}$ when k is increased, and $\mathbf{X}$ is better approximated until $k = Rank(\mathbf{X})$ and $\mathbf{X}_k = \mathbf{X}$ and $\mathbf{E} = 0$. This normally requires a large k, and is hence unpractical and not a satisfying dimensional reduction. Describing $\mathbf{X}$ in this $Noise + Structure$ form is what the author calls a ”PCA Model”.

Typically, only the first k principal components of the projection account for most of the variance in the objects. This is because singular values and -vectors of the SVD are Rank-ordered, meaning that less variance in $\mathbf{X}$, or less magnitude of singular values, is explained when adding a principal component to the model. For this reason, the other ”less important” principal components are discarded in further analysis. According to this procedure, the reconstruction of the original data matrix would not be lossless, but the significant activity in the data regarding variance is hopefully included along the existing principal components.

In a practical scenario, $\mathbf{X}$ is typically mean centered and scaled before conducting PCA, in order to get a centered PCA model and deal with different data sources, respectively.

Each i value in a column n of $\mathbf{X}$ is mean centered by a column-wise subtraction of the following mean in equation 5.8

$$x_{mean,n} = \frac{1}{M}\sum_{i=1}^{M} x_{i,n} \tag{5.8}$$

The mean in equation 5.8 for a given column n is subtracted from each i value $x_{i,n}$ in that column

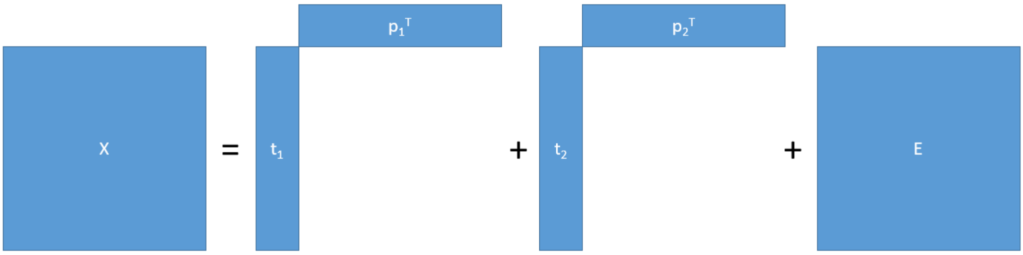


Figure 5.2: Illustration of an approximation of $\mathbf{X}$ by using only the $k = 2$ first PCs from a PCA model of the data contained in $\mathbf{X}$. The approximation of $\mathbf{X}$ is the linear independent variations in the data expressed in PC1 and PC2 and is a sum of matrix products of the scores and loadings from PC1 and PC2 respectively. This is the $Structure$ part of the figure and is the approximation of $\mathbf{X}$. $\mathbf{E}$ is the $Noise$ part and contains the rest of the variational information in $\mathbf{X}$, thus not expressed in the selected PCs. The rest of the principal components (from for example SVD) are discarded in the approximation when only a restricted set of components (here 2) are selected to approximate $\mathbf{X}$.

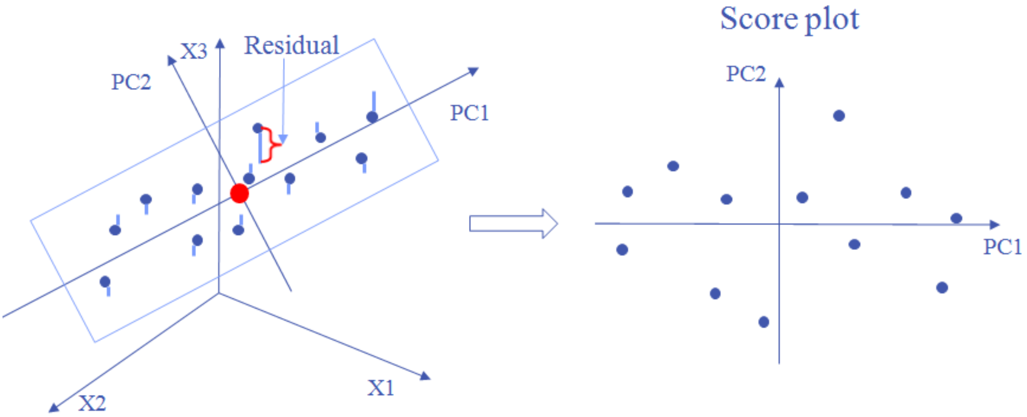


Figure 5.3: A geometric interpretation of PCA by looking at first two PCs. Samples are shown in both variable space (X1, X2, X3) and component (reduced dimensional) space (PC1, PC2). Residuals are displayed for each sample as well. PCs capture latent variables, and they express different variational patterns.

according to equation 5.9, resulting in mean centered values $\widetilde{x}_{i,n}$

$$\widetilde{x}_{i,n} = x_{i,n} - x_{mean,n} \tag{5.9}$$

A typical scaling is to divide each column n by its standard deviation. This is called scaling to unit variance. Unless explicitly stated, scaling is not done in the models in the thesis, since the variables are from the same source and thus comparable without scaling. Referring to (Galindo-Prieto, 2017) for scaling methods.

5.2 Partial Least Squares projections to latent structures (PLS)

Also called Partial Least Squares Regression, PLS introduces a new matrix $\mathbf{Y}$, which acts as a response block to the $\mathbf{X}$ block containing vector annotations/labels for each row sample in $\mathbf{X}$. PCA is often referred to as an exploratory method while PLS regression is focused on modelling relationships between two or more blocks of data (for example between $\mathbf{X}$ and $\mathbf{Y}$). The columns of $\mathbf{X}$ are regarded as independent variables, while columns of $\mathbf{Y}$ are dependent variables. The number of rows are the same in both matrices. A PLS model decomposes $\mathbf{X}$ and $\mathbf{Y}$ into equation 5.10 and 5.11 (outer relations) while using regression to maximize the covariance between $\mathbf{X}$ and $\mathbf{Y}$ (by introducing the inner relation 5.12) (Geladi and Kowalski, 1986; Wold et al., 1983; CAMO, 2017c). After this calibration step, new samples for $\mathbf{X}$ can be used to predict new samples for $\mathbf{Y}$ by using the inner relation 5.12 and loadings $\mathbf{Q}$ as in Section 5.3.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} = \sum t_h p_h^T + \mathbf{E} \tag{5.10}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F}^* = \sum u_h q_h^T + \mathbf{F}^* \tag{5.11}$$

$$\hat{u}_h = \frac{u_h^T t_h}{t_h^T t_h} t_h \tag{5.12}$$

In equation 5.10, 5.11 and 5.12, h is called a factor and is the analog to a PC in PCA. The equations are taken from (Geladi and Kowalski, 1986). The optimal decomposition is found using a modification of the Nonlinear iterative partial least squares (NIPALS) algorithm (Geladi and Kowalski, 1986). In the NIPALS PLS algorithm used, $\hat{u}_h = 1 * t_h$, hence (CAMO (2017c), Chapter 16.2.4, p. 680)

$$\mathbf{U} = \mathbf{T} \tag{5.13}$$

Other algorihms for PLS exist as well such as the SIMPLS algorighm used default in Matlab (de Jong, 1993) and kernel PLS algorithm (Lindgren et al., 1993). SIMPLS might be preferred over NIPALS (Aylin, 2009; de Jong, 1993), while kernel PLS is a non-iterative algorithm that is expected to be efficient on data with a large number of samples compared to number of variables (Lindgren et al., 1993; CAMO, 2017c). When the number of samples and variables are both large, NIPALS is preferable over kernel PLS (CAMO, 2017c). This is the case with most of the data analyzed.

The decompositions in equation 5.10 and 5.11 are still in the form of scores and loadings as in Figure 5.2. However, they are based on maximizing the covariance between the scores $\mathbf{T}$ and $\mathbf{U}$. An effect of this optimization is, for example, that if $\mathbf{X}$ has a latent structure that is important in predicting $\mathbf{Y}$, it will be weighted much in the PLS model, even if it would correspond to a small singular value from SVD on $\mathbf{X}$ alone (Wold et al., 1983).

5.2.1 Partial Least Squares Regression Discriminant Analysis (PLS-DA)

If the response block uses categorical variables, it is called a PLS-DA model (Bylesjö et al., 2007). The $\mathbf{Y}$-variables then represent the classes / categories and affect the calibration and prediction.

5.3 Projecting a new sample onto score space in a bilinear model

For a selected number of components k, consider the following bilinear model of $\mathbf{X}$, equation 5.14 (for example in PCA and PLS)

$$\mathbf{X} = \mathbf{T}_k\mathbf{P}_k^T + \mathbf{E} \tag{5.14}$$

where $\mathbf{X} \in \mathbb{R}^{M \times N}$, $\mathbf{T}_k \in \mathbb{R}^{M \times k}$, $\mathbf{P}_k^T \in \mathbb{R}^{k \times N}$ and $\mathbf{E} \in \mathbb{R}^{M \times N}$.

A new sample $x \in \mathbb{R}^{1 \times N}$ can be projected onto the score space by modification of equation 5.14, equation 5.15

$$x_T = x\mathbf{P}_k \tag{5.15}$$

where $x_T \in \mathbb{R}^{1 \times k}$ and $\mathbf{P}_k \in \mathbb{R}^{N \times k}$.

The new sample and its projection onto score space can be appended row-wise to $\mathbf{X}$ and $\mathbf{T}_k$ respectively and new residuals for the $M + 1$ samples follow equation 5.16

$$\mathbf{E}_{new} = \mathbf{X}_{new} - \mathbf{T}_{k,new}\mathbf{P}_k^T \tag{5.16}$$

5.4 Important metrics derived from a PCA and PLS models

When interpreting a MVA models from PCA and PLS, there are a few important graphs that can reveal important properties in the data. Most of them are discussed in the results section for the data analyzed.

5.4.1 Interpreting scores

The scores is the matrix product of the left-singular vectors and singular values from SVD (PCA case) or NIPALS (PLS case), and project each sample into an ortogonal basis. It is thus a map of objects.

Radically different samples will be projected differently, which can be displayed in various ways. Plotting a single score vector displays how each object relate to each other in terms of magnitude for this particular component. Plotting score vectors up to each other can reveal groups of similarly projected objects when expressed in terms of these components.

5.4.2 Interpreting loadings

The loadings are the right singular vectors from SVD or NIPALS, and contain a map of contributions from original variables to the model in terms of amplitude (loadings), as well as the correlation between the variable and component (correlation loadings). Line and multidimensional loading plots display which of the original variables contribute the most to patterns captured in the components (loadings).

For a given variable x and component y, the correlation loading is the cosine of the angle θ between the variable and component, which with mean centered variables, is the correlation between the variable and component (CAMO, 2017c)

$$cos(\theta) = \frac{x^T y}{||x||||y||} = \frac{\frac{1}{M-1} x^T y}{\frac{1}{M-1}\sqrt{x_1^2 + x_2^2 + .. + x_M^2}\sqrt{y_1^2 + y_2^2 + .. + y_M^2}} = \frac{Cov(x,y)}{s_x s_y} = Cor(x,y) \tag{5.17}$$

where s_x and s_y are the standard deviations for x and y respectively. Exact correlation yields a correlation loading of 1, while an exact opposite correlation yields a correlation loading of -1.

The scaling of loadings to correlation loadings is done according to the following formula (Martens and Martens, 2001; CAMO, 2017a)

$$\mathbf{P}_{corr,k}^T = \frac{\sqrt{\mathbf{T}_k^T \mathbf{T}_k}}{\sqrt{\mathbf{X}_k^T \mathbf{X}_k}} \mathbf{P}_k^T \tag{5.18}$$

With k selected PCs.

Loadings are very important in this work, as they show exactly which electrode or frequency component is the most responsible for a particular pattern. During analysis and interpretation of the various PCA models it became clear that the standard loading plot yielded the same information as the correlation loadings. Therefore, the tradition loading plots were included.

5.4.3 Residuals and explained variance

If the MVA model expresses 100% of the variance in $\mathbf{X}$, then $\mathbf{E}_k = 0$ and $k = Rank(\mathbf{X})$. This means 100% explained variance and 0% residual variance from the relation

$$ExplainedVariance + ResidualVariance = 100\% \tag{5.19}$$

The residual variance is calculated from residues in $\mathbf{E}_k$. A residue for a sample e_i is a distance between the sample in variable space and the representation of the sample in component space. It is important that variable segments in $\mathbf{X}$ are mean-centered. Then the residual variance is relative to the average sample of $\mathbf{X}$, and is calculated using the formula (Risvik, 2007)

$$ResidualVariance = e_{tot}^2 = \sum_{n=0}^{N-1} e_i^2 \tag{5.20}$$

5.19 and 5.20 is used to make a plot of explained variance, which can be used to select an appropriate number of components for the model.

5.4.4 Validating models - stability and optimal model dimensionality

Validation is very important in all scientific work. In relation to the methods described above the focus is on the numerical aspects. In this thesis, cross validation (Appendix Chapter 8.4.12) is applied in order to find the optimal dimensionality of a multivariate model i.e. to avoid either overfitting or underfitting and also provide correct interpretation. This is often mostly presented for regression models in terms of prediction or classification error but it is also important for exploratory analysis and methods such as PCA (Westad and Marini, 2015).

Cross validation is also applied to estimate the stability of the models, i.e. some samples are systematically held out and a model is calculated on the remaining samples until all samples have been held out once. This results in the validated explained variance and is to be compared with the calibration variance as more components are added to the model. If there is no clear structure in the data, the validation curve will reveal that one is starting to model noise, and thus the model is not stable. Validation can be applied at various levels, e.g. across time, experiments etc. Projecting a new set of data onto an existing model can also be seen as a way of validation in the sense if something has changed compared to an earlier state of a system. This will be exemplified in the Results section.

5.4.5 Influence plots

Various statistical metrics for the objects can be summed up in an influence plot, with the main aim of detecting outliers. In The Unscrambler software used, an Influence Plot shows the Q- or F-residuals vs Leverage or Hotelling's T-squared statistics (Hotelling, 1931, 1933) for any sample. They are typically used to detect outlier objects.

Generally speaking, the metrics on the y-axis (Q- or F-residuals) describe the sample distance to the model. Large value means that the sample is poorly described by the model. The x-axis (Leverage or Hotelling's T-squared) describes how well the sample is described by the model. Large value means that the sample is well described in the model.

Q- and F-residuals are just two different ways for testing a sample's distance to the model. The difference between Hotelling's T-squared and Leverage is only a scaling factor. Hence, choosing the correct Influence Plot should not be the most important matter.

Both axes in the influence plot introduces statistical limits. For Hotelling, the limit on the x-axis is based on a student-t distribution. For F-residuals, the limit on the y-axis is based on an f-test (CAMO, 2017c).

5.4.6 Ellipses in 2D scores and loading plots

Hotelling's T^2 ellipse in 2D scores

The same confidence limit originating from a student-t test in x-axis in the influence plots (Hotelling, 1931, 1933) can be used directly on 2D score plots, to give a view of influential samples in the model. According to this metric, samples outside of this ellipse have too large impact on the overall MVA model, or too high leverage, when compared to other samples and might lead to a worse MVA model with respect to the variational patterns of the other samples.

Correlation loadings in 2D loadings

In the 2D loadings plot, the importance of individual variables in the model might be more clearly visualized with correlation loadings only. Since correlation loadings range from -1 to 1, plotting an inner and outer ellipse can help in determining important variables. A variable spanned by only one component, and directly on the outer ellipse have an exact correlation with that component. In the Unscrambler software used, the inner ellipse limit means 50 % explained variance, which conforms to a correlation of the variable and component of 0.707.

5.5 Variable Influence on Projection (VIP) for Orthogonal PLS adapted to PLS-DA

There exist other methods to quantify the influence of individual (original) variables in PLS-DA models. The variable influence on projection (VIP) algorithm presented by (Galindo-Prieto et al., 2014) (OPLS-VIP) was adapted to PLS-DA case in this work (i.e., considering only the PLS predictive components, but using the OPLS-VIP advantages (Galindo-Prieto et al., 2014, 2015)).

A combination of normalized versions of the loadings ($\mathbf{p}$) and the sum of squares of the $\mathbf{X}$ matrix and the $\mathbf{Y}$ response (SSX and SSY) is used to obtain a VIP value for each variable. The pre-processed $\mathbf{X}$ and $\mathbf{Y}$ matrices used in the PLS-DA model were used as inputs for computing the VIP values in MATLAB, as well as the normalized loading matrices for $\mathbf{X}$ and $\mathbf{Y}$ ($\mathbf{P}$ and $\mathbf{Q}$), the loading weights matrix ($\mathbf{W}$), and the $\mathbf{X}$-score matrix ($\mathbf{T}$). VIP is thus a model based variable selection method. The VIP calculation can be summarized as shown in Equation 5.21.

$$VIP_{OPLS(pred)} = \sqrt{K_p \times \left(\frac{\left[\sum_{a_p=1}^{A_p} (\mathbf{p}_{p,a_p}^2 \times SSX_{comp,a_p}) \right]}{SSX_{cum}} + \frac{\left[\sum_{a_p=1}^{A_p} (\mathbf{p}_{p,a_p}^2 \times SSY_{comp,a_p}) \right]}{SSY_{cum}} \right)} \tag{5.21}$$

where

$$K_p = \frac{K}{\frac{SSX_{cum,a_p}}{SSX_{cum}} + \frac{SSY_{cum,a_p}}{SSY_{cum}}} \tag{5.22}$$

K is the total number of original variables in $\mathbf{X}$ and K_p is an expression of K for the predictive components $_p$. A_p is the total number of predictive components. $\mathbf{p}_p$ are normalized versions of loadings for the predictive components. Lastly, the SS values are sum of squares measures. SSX_{comp,a_p} is the SS of $\mathbf{X}$ explained by predictive component a_p, while SSX_{cum} is the total SS of $\mathbf{X}$ explained by all components in the model (Galindo-Prieto et al., 2015).

PLS-DA only computes predictive components. The resulting VIP vector presents importance or influence of the original variables in predicting $\mathbf{Y}$ in the PLS-DA model.

5.5.1 Sum of squares of X for a component in a bilinear model

When reconstructing $\mathbf{X} \in \mathbb{R}^{M \times N}$ by bilinear approximation using either all components in the MVA model ($\mathbf{X}_{cum}$) or a selected component ($\mathbf{X}_{comp}$), the sum of squares for the reconstructed $\mathbf{X}$ is defined in equation 5.23

$$SSX = \sum_{j=1}^{N} \sum_{i=1}^{M} (x_{i,j} - \bar{x})^2 \tag{5.23}$$

where $\bar{x}$ is a scalar. It is the sum of the squared differences of each observation from the overall mean, over all variables for the reconstructed matrix.

Chapter 6

Results

6.1 PCA

6.1.1 Models on offline preprocessed (OP) data show trend in features captured in increasingly more PCs as the age of the culture increases

All models were made in The Unscrambler with mean centering, no scaling to unit variance and with 20 segment random cross-validation. Data from reference electrode (Ref) was not included in any PCA model. For some experiments, certain electrodes with constantly larger amplitudes or transient behavior were kept out from the model.

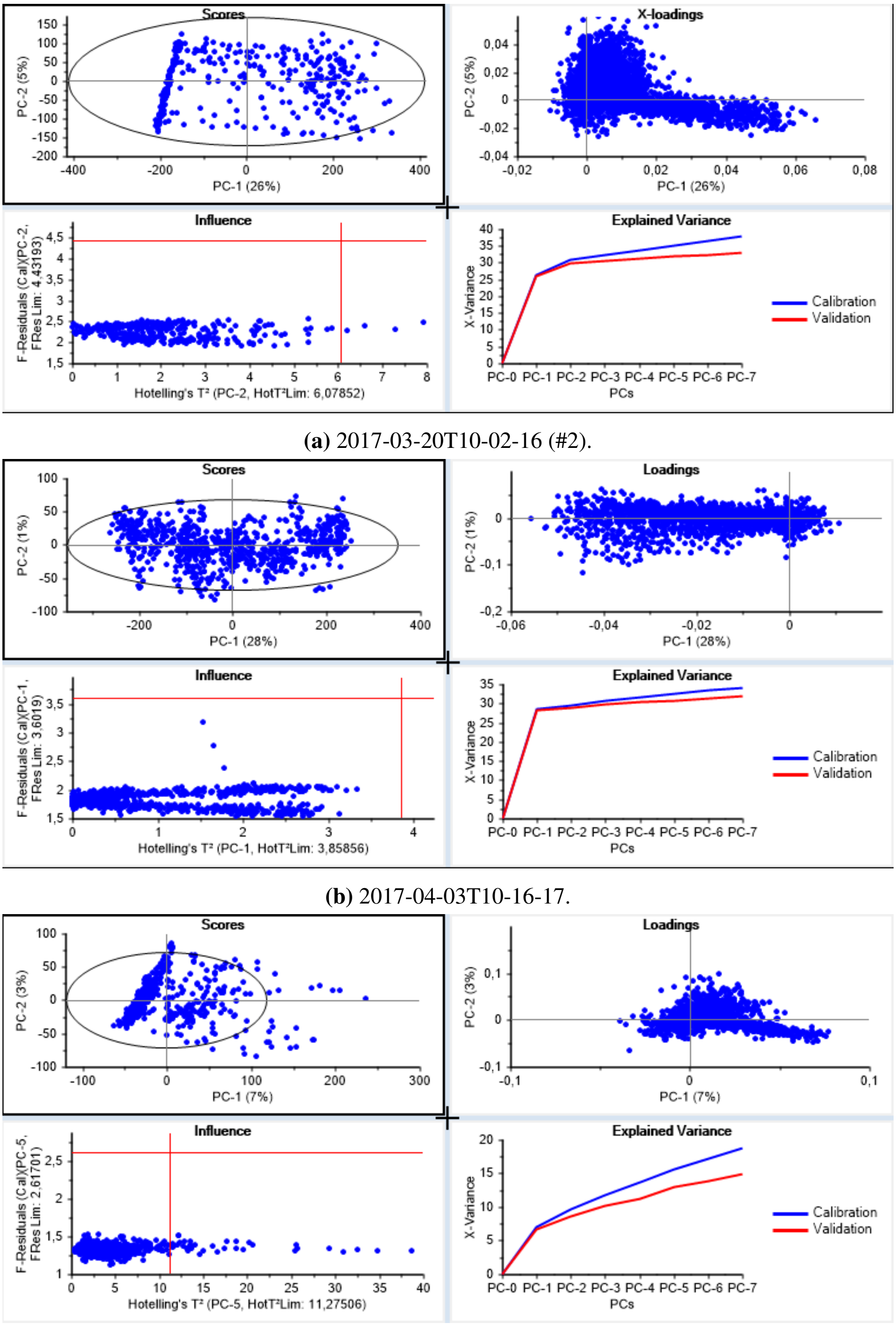
Overview of scores, loadings, influence plot and explained variance for PCA on each experiment can be seen in Figures 6.1, 6.2 and 6.3 for the offline preprocessing with noise reduction on old data in Table 2.2, and in Figures 6.4 and 6.5 for offline preprocessing on non-noise reduced data in Table 2.3.

From explained variance plots of models on the properly noise reduced data (Figures 6.1, 6.2 and 6.3) it is seen that the culture requires increasingly more principal components to explain the variance as the age of the culture increases. For the experiments on the even older culture (Table 2.3), the PCA models explain most of the variance with only one principal component. It is assumed that this must be because the experiments were not noise reduced by skipping the noise reduction step 3.3 (not that the activity has developed into less complex forms).

Why the new data set was not properly noise reduced

From looking at raw data it became clear that the noise profiles (Signal-to-Noise Ratio (SNR)) on the experiments on the oldest cultures in the new data set in Table 2.3 were very different to the data in the old data set in Table 2.2. This lead to the fact that the same noise segment (#2, electrode 87 data between third and four visible burst) used to noise reduce experiments in the Table 2.2 could not be used to noise reduce the new data set satisfyingly.

Additionally, since recordings of the adult culture in the new data set did not seem to show visible bursts and the SNR seemed to be higher (less noise) than in the old data set, it was not investigated further on how to noise reduce the new data set. A higher SNR could legitimate to not do noise reduction and still reveal information in the PCs. However, this seemed to not be the case in the PCA models on the not noise reduced data (Figures 6.4 and 6.5).

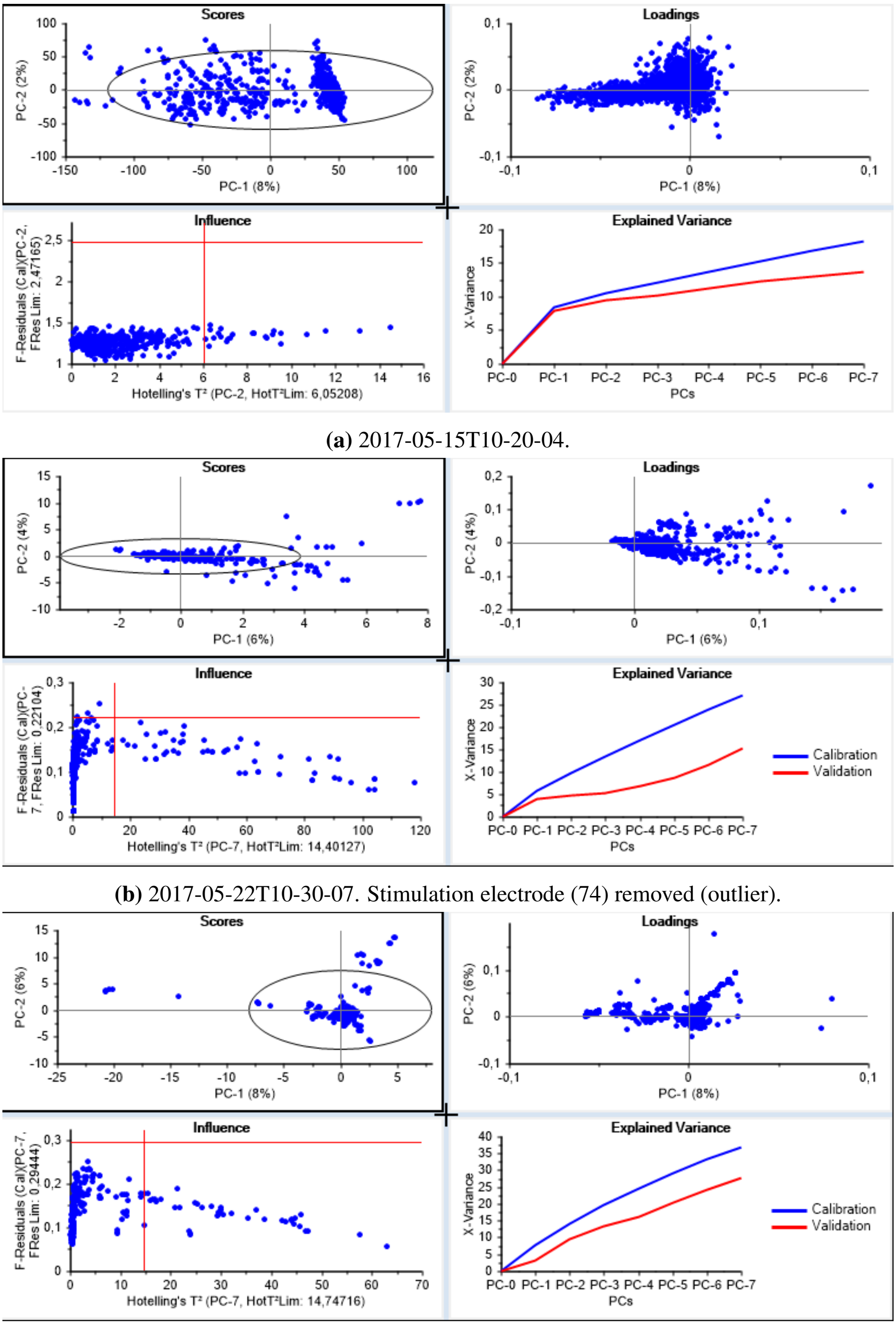


(a) 2017-03-20T10-02-16 (#2).

(b) 2017-04-03T10-16-17.

(c) 2017-05-02T15-18-41 (#3).

Figure 6.1: PCA models from offline preprocessing of Table 2.2, **part 1**. Scores and loadings for PC1 and PC2, influence plot selected for varying number of PCs and explained variance plot for model training (calibration) and validation. Hotelling's T^2 ellipse is shown for 2D scores. Reference electrode (Ref) is kept out for all models.

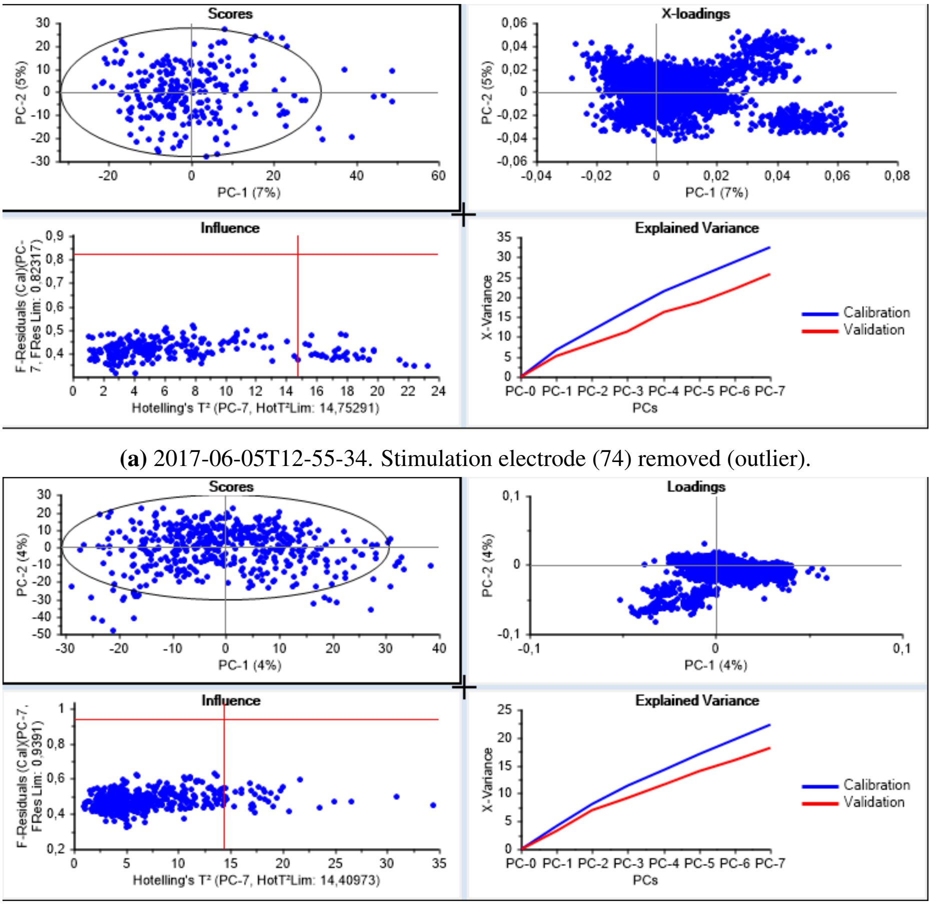


(a) 2017-05-15T10-20-04.

(b) 2017-05-22T10-30-07. Stimulation electrode (74) removed (outlier).

(c) 2017-06-05T12-49-11. Electrode 31 removed (outlier).

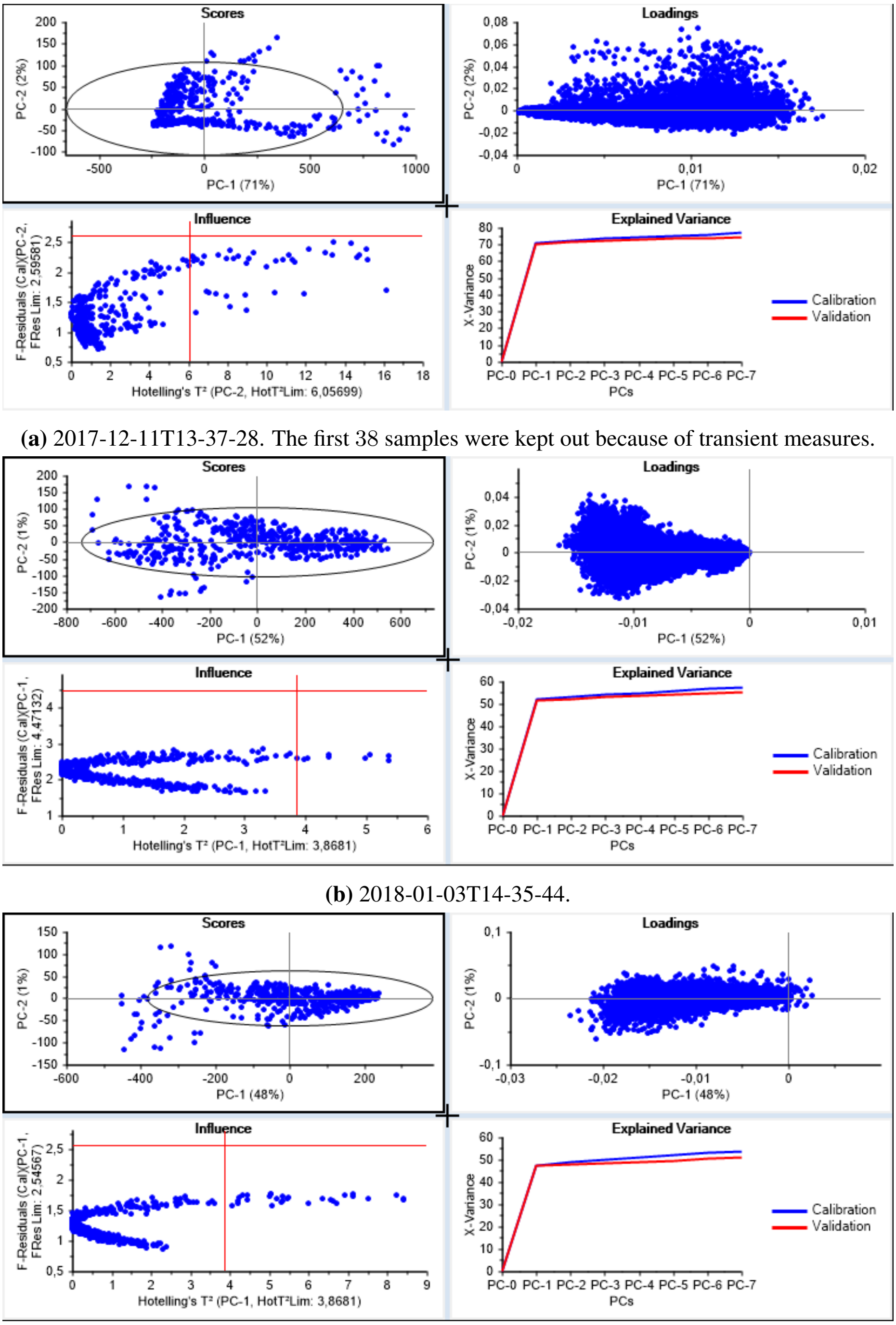
Figure 6.2: PCA models from offline preprocessing of Table 2.2, **part 2**. Scores and loadings for PC1 and PC2, influence plot selected for varying number of PCs and explained variance plot for model training (calibration) and validation. Hotelling's T^2 ellipse is shown for 2D scores. Reference electrode (Ref) is kept out for all models.



(a) 2017-06-05T12-55-34. Stimulation electrode (74) removed (outlier).

(b) 2017-06-12T11-40-48 (#4). Approx. first 12 samples are kept out from the model because of measurement transients.

Figure 6.3: PCA models from offline preprocessing of Table 2.2, **part 3**. Scores and loadings for PC1 and PC2, influence plot selected for varying number of PCs and explained variance plot for model training (calibration) and validation. Hotelling's T^2 ellipse is shown for 2D scores. Reference electrode (Ref) is kept out for all models.

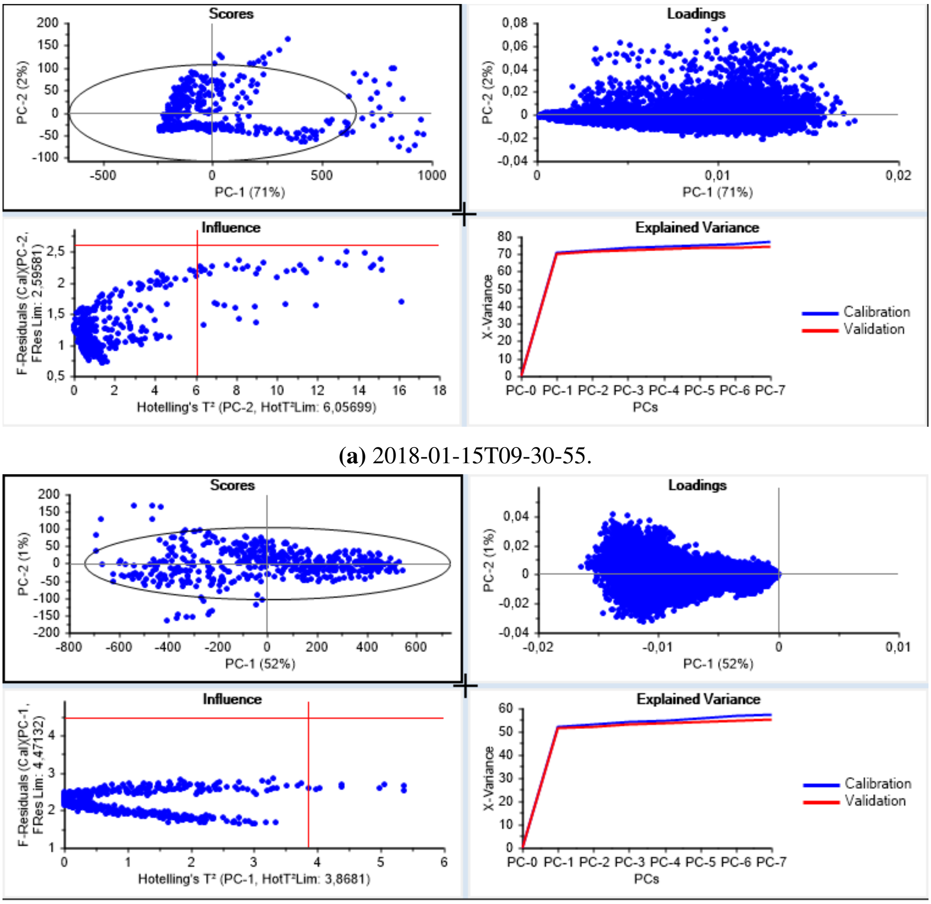


(a) 2017-12-11T13-37-28. The first 38 samples were kept out because of transient measures.

(b) 2018-01-03T14-35-44.

(c) 2018-01-08T13-40-39.

Figure 6.4: PCA models from offline preprocessing of Table 2.3 (preprocessed on not noise reduced data), **part 1**. Scores and loadings for PC1 and PC2, influence plot selected for varying number of PCs and explained variance plot for model training (calibration) and validation. Hotelling's T^2 ellipse is shown for 2D scores. Reference electrode (Ref) is kept out for all models.



(a) 2018-01-15T09-30-55.

(b) 2018-01-22T11-05-33.

Figure 6.5: PCA models from offline preprocessing of Table 2.3 (preprocessed on not noise reduced data), **part 2**. Scores and loadings for PC1 and PC2, influence plot selected for varying number of PCs and explained variance plot for model training (calibration) and validation. Hotelling's T^2 ellipse is shown for 2D scores. Reference electrode (Ref) is kept out for all models.

6.1.2 Comparison of influential variables in PCA models from OP with raw data reveals that synchronized APs in raw data are detected in PCs and they get shorter in duration until they disappear from raw data, as the age of culture increases. Then frequency bins show distinct variation in the form of clusters in loading space.

For a young culture, influential variables in PCA models are shown to relate to the synchronized bursts visible on raw recordings. In old/adult culture, variables in PCA models tend to form clusters in the 2D loading space. It is in this way observed that certain variables/frequency bins become more distinguished in the PCA model when the culture is old/adult, or more than approximately 8 month old (given that they are properly noise reduced) (Figures 6.8 and 6.9 (a)). A closer look on 2D loading space for PC1 and PC2, line plots of scores for PC1 and PC2 and their connection to the most important single electrode raw data are shown for some selected experiments in Figures 6.6, 6.7, 6.8 and 6.9.

Note that a variable is a 10 Hz frequency bin, and each electrode has $(3000 - 300)/10 = 270$ variables in the range $300 - 3000$ Hz. If all electrodes are used in the PCA model, the number of variables is $270 * 60 = 16200$ variables (N). There is a lot of variables compared to the number of samples, which in an exact 10 min recording would be $(10 * 60)/1.2 = 500$ samples (M). The explained variance in PC1 in experiments with clusters in loading space are only around approx. 7 % (experiment #4). However, the variance continues to get explained as the number of PCs are added, and with 7-component PCA models total explained variance end up in approx. 20 % for the models with clusters in 2D PC1-PC2 loading space. It is likely that explained variance will continue to increase as more PCs are added on larger PCA models on this data. If that is not the case, then selecting only the important variables to make a new PCA model would show higher explained variance.

In recordings of a younger culture where no grouping in loading space appear, scores from PC1 tend to express the actual bursts that are visible in raw data on some electrodes (Figures 6.6 and 6.7). Also, the duration of each individual burst gets shorter as the age of the culture progresses until ultimately there exists too little covariance from this type of activity (oscillations in APs across frequency bins, see Figure 2.13 (a)) to be picked up in a single PC.

The PCA model of experiment 2017-05-22T10-30-07 (Figure 6.7 (b)) is a unique model that perhaps expresses a transition phase between burst behavior to specialized behavior of variables/frequency bins (in the form of groups in loading space) for the MEA culture. In this particular model, APs (not the small spike-like APs, they are probably caused by stimulation) in raw data of a significant electrode in the model (35) are expressed somewhat in both PC1 and PC2 score.

For the PCA models of experiments on adult cultures with sufficient preprocessing (Figures 6.8 and 6.9 (a)), there are no visible APs in the raw data of electrodes belonging to influential frequency bins. APs in the frequency bins still occur, but since they not synchronized across the frequency bins they are not visible in raw data. This is when clusters in loading space show up when the preprocessing is working as expected. The preprocessing is working as expected if amplitudes in frequency bins over time exceeding a given threshold, actual represent an AP, and nothing else (such as noise constantly over threshold amplitudes from improper noise reduction).

6.1.3 Frequency bin clustering in loading space seems to be independent of electrode placement in the MEA for adult culture age

Do specific variables or electrodes group in the loading space? Frequency bins belonging to four electrodes, each from each corner of the MEA, were marked one after another in order to see their positions in PC1-PC2 loading space. This was done for the three PCA models with visible groups (Figures 6.10, 6.11 and 6.13). Line loading for the model in Figure 6.11 is show as well in Figure 6.12. The interesting conclusion is that the physical placement of an electrode does not seem to have an impact on which variables/frequency bins are part of a given cluster. Each of the selected electrodes seem to have at least one of their 270 frequency bins in any of the clusters. This could imply that for adult cultures, a real-time analysis on only one electrode could be sufficient in describing the activity.

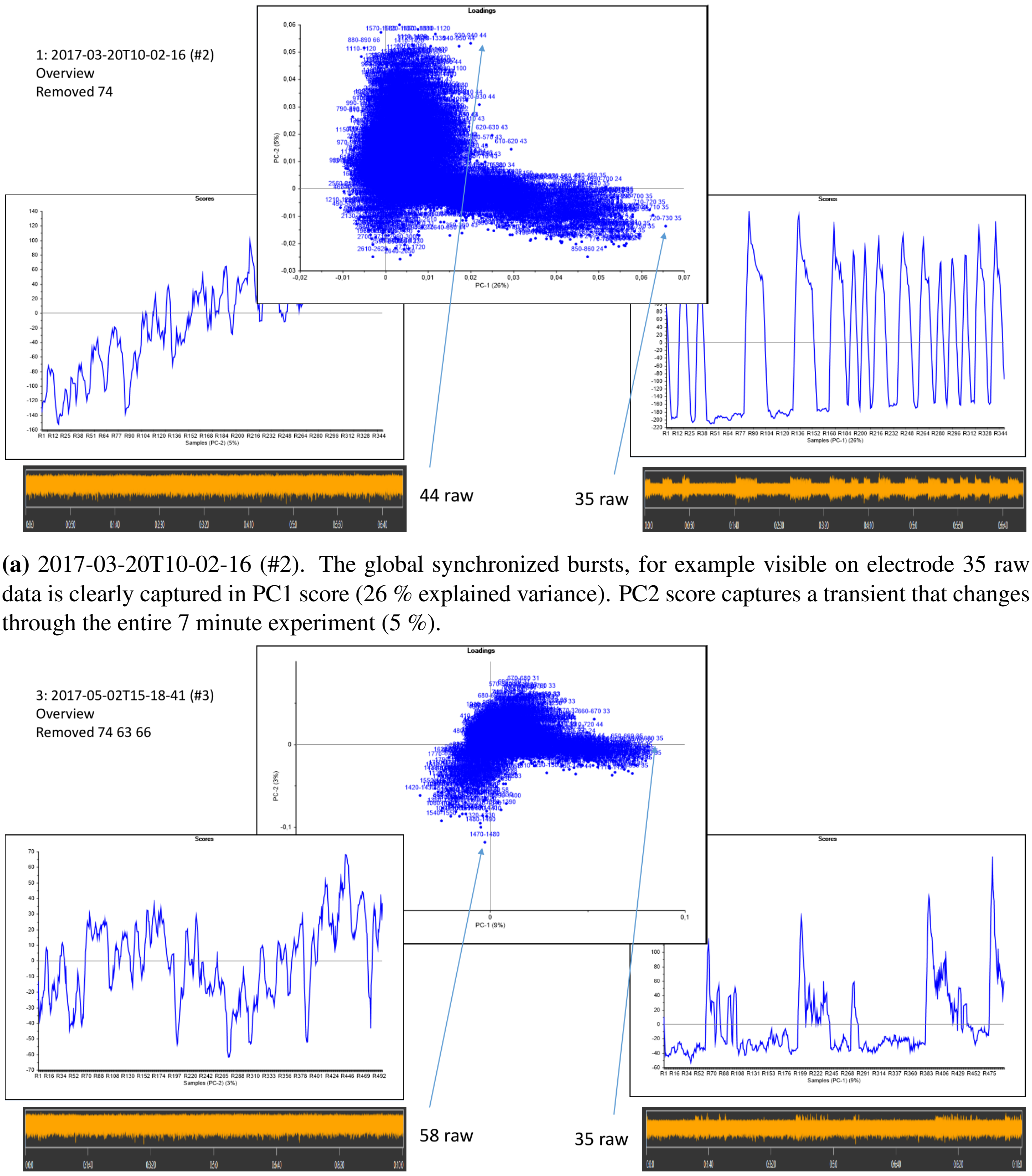
In these PCA models of the adult culture, there is no single electrode that takes a clear leading role in explaining the variance, as in models of the young culture. Rather, specific frequency bins/variables from multiple electrodes with specific frequency range are important in the models. This is perhaps unexpected since stimulation occurs in all experiments except 2017-06-12T11-40-48 (#4) and it would be reasonable that stimulation alters the measurement on some electrodes more than others. However, this is not the case in adult experiments with stimulation. Specific frequency components are important, and they are sustained across electrodes. It must be noted that some electrodes that were very noisy for reasons of transients or constantly larger amplitudes from instrument calibration, were removed from some models.

For younger ages (models of earlier experiments) it is clear that certain electrodes (not certain frequency bins) are more influential than others in describing activity. Compare f. ex. the synchronized activity in Figures 6.6 and 2.15 with Figures 6.9 (a) and 2.17. Maybe the stimulation was part of the cause of distinct electrode activity in young age. Nevertheless, stimulation seems to take a less important role in adult age.

6.1.4 Varying model stability after projecting OP data of experiments into PCA models of other experiments

The offline preprocessed data of some experiments were projected onto score space of offline preprocessed PCA models for some selected experiments (Chapter 5.3) and metrics such as influence plot and explained variance recalculated. Projection of experiment data were made onto score space of an old culture (experiment #4, Figures 6.14 and 6.15) and a young culture (experiment #2, Figures 6.16 and 6.17).

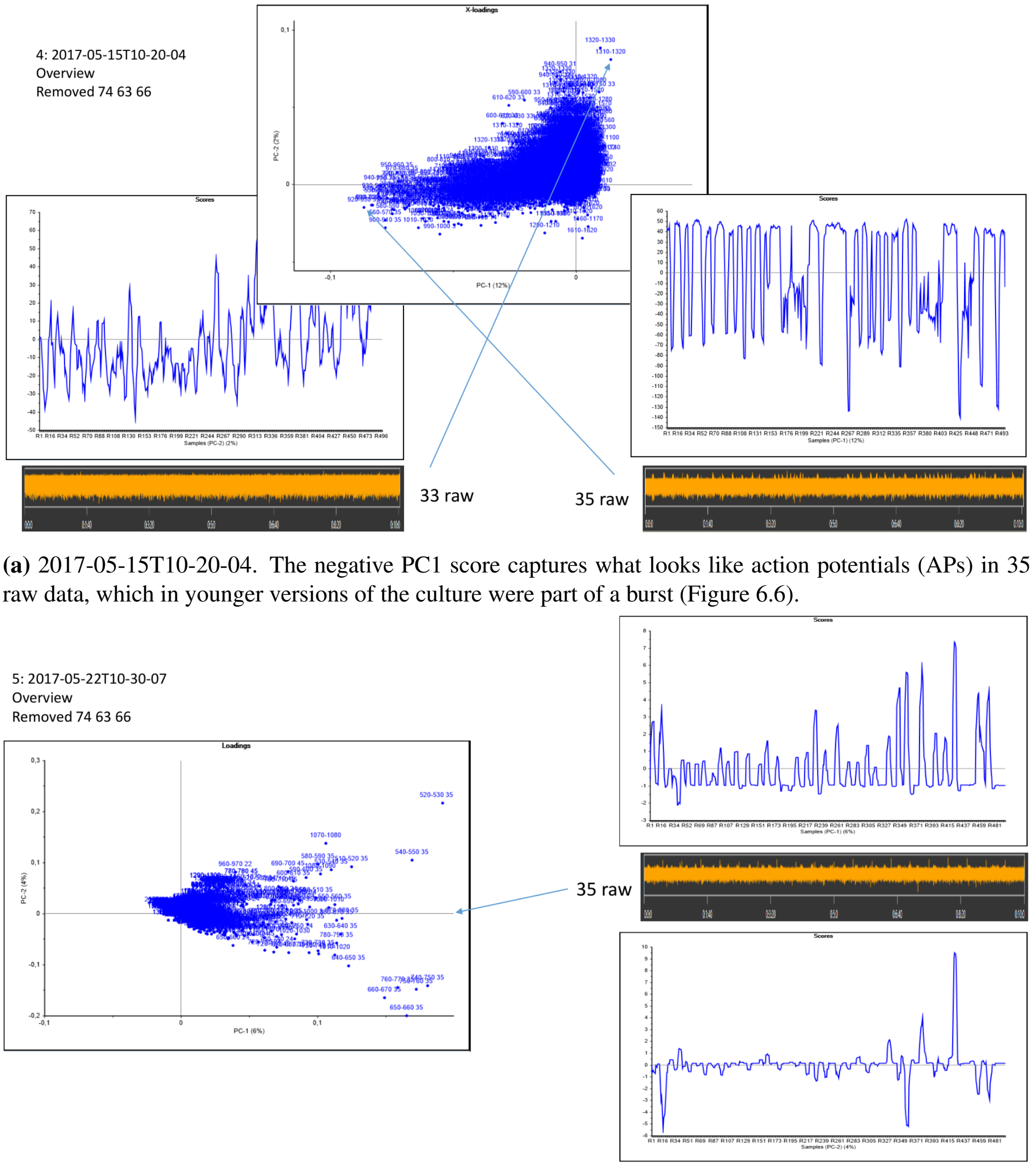
As expected, the projected samples are closer to the original model samples in score space when the two data sets represent similar ages of the culture. External parameters such as preparation before measurement can be slightly different from experiment to experiment. Nevertheless, for the projections onto score space in the selected model of the young culture (Figures 6.16 and 6.17), from line scores (red are the projected samples, same as green in the 2D score space) it can be seen that synchronized oscillations in bursts are somewhat captured by PC1, hence the latent variable describes bursts in both data sets (data for training the PCA models and the projected data for testing it). For the adult culture projections (Figures 6.14 and 6.15), PC2 seems to distinguish the most between the projected and model samples.



(a) 2017-03-20T10-02-16 (#2). The global synchronized bursts, for example visible on electrode 35 raw data is clearly captured in PC1 score (26 % explained variance). PC2 score captures a transient that changes through the entire 7 minute experiment (5 %).

(b) 2017-05-02T15-18-41 (#3). PC1 score still captures what looks like bursts in 35 raw data.

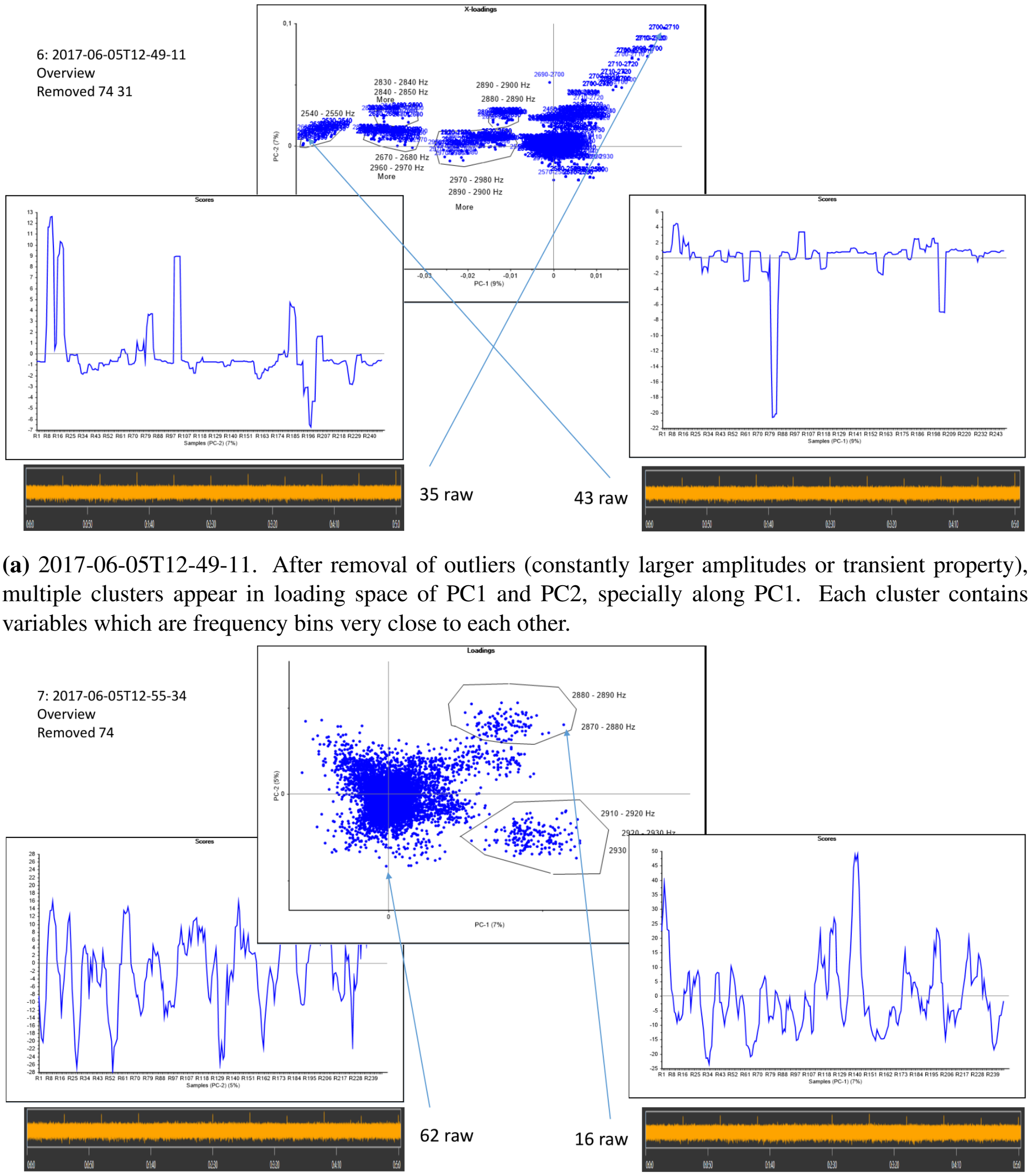
Figure 6.6: Investigating scores, loadings and connection to raw data, **part 1**.



(a) 2017-05-15T10-20-04. The negative PC1 score captures what looks like action potentials (APs) in 35 raw data, which in younger versions of the culture were part of a burst (Figure 6.6).

(b) 2017-05-22T10-30-07. The last experiment with burst-like behavior. The AP behavior is more advanced since it is expressed somewhat in two PC scores (PC1 and PC2 score).

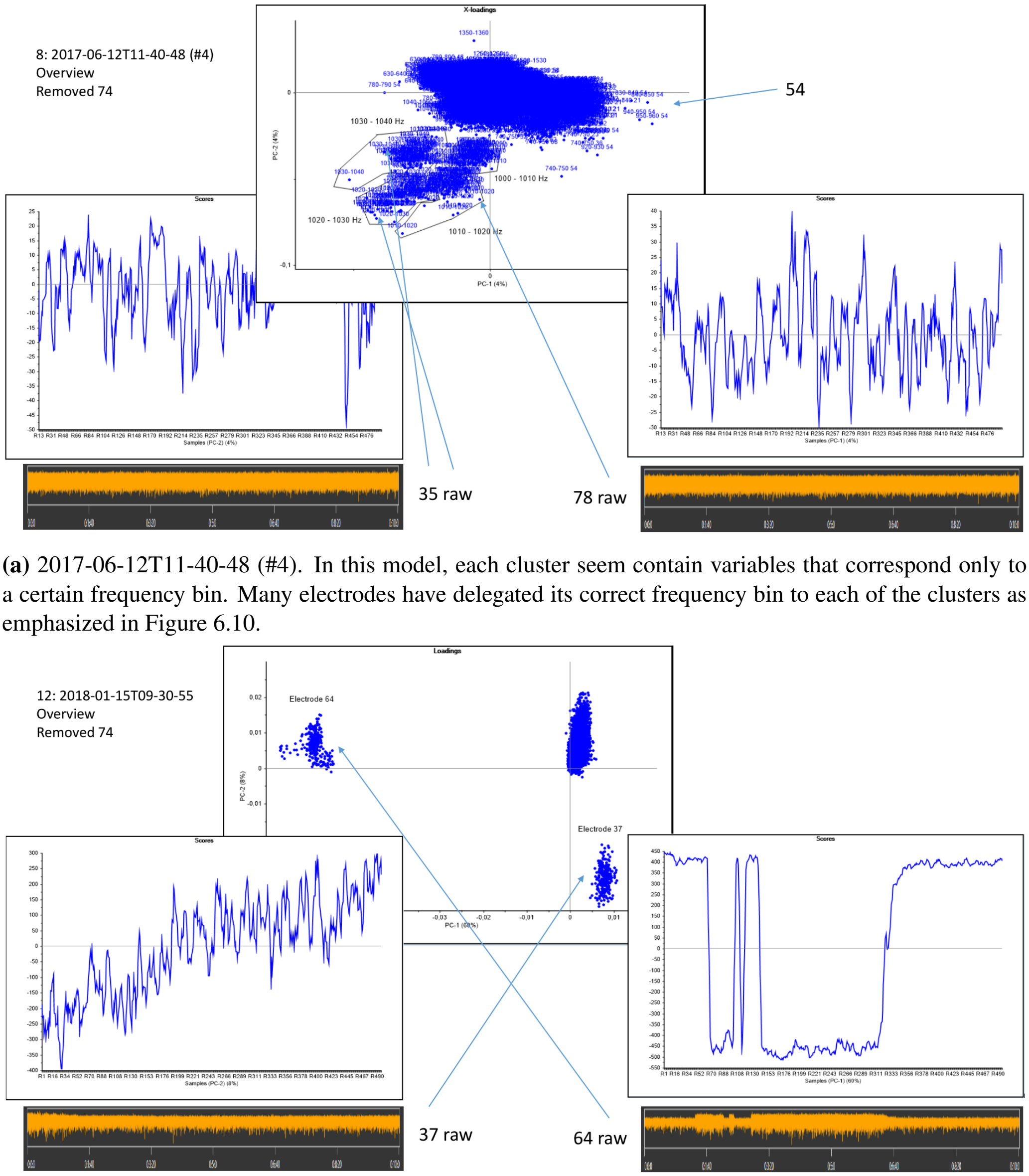
Figure 6.7: Investigating scores, loadings and connection to raw data, **part 2**.



(a) 2017-06-05T12-49-11. After removal of outliers (constantly larger amplitudes or transient property), multiple clusters appear in loading space of PC1 and PC2, specially along PC1. Each cluster contains variables which are frequency bins very close to each other.

(b) 2017-06-05T12-55-34. Similar cluster properties to **(a)**.

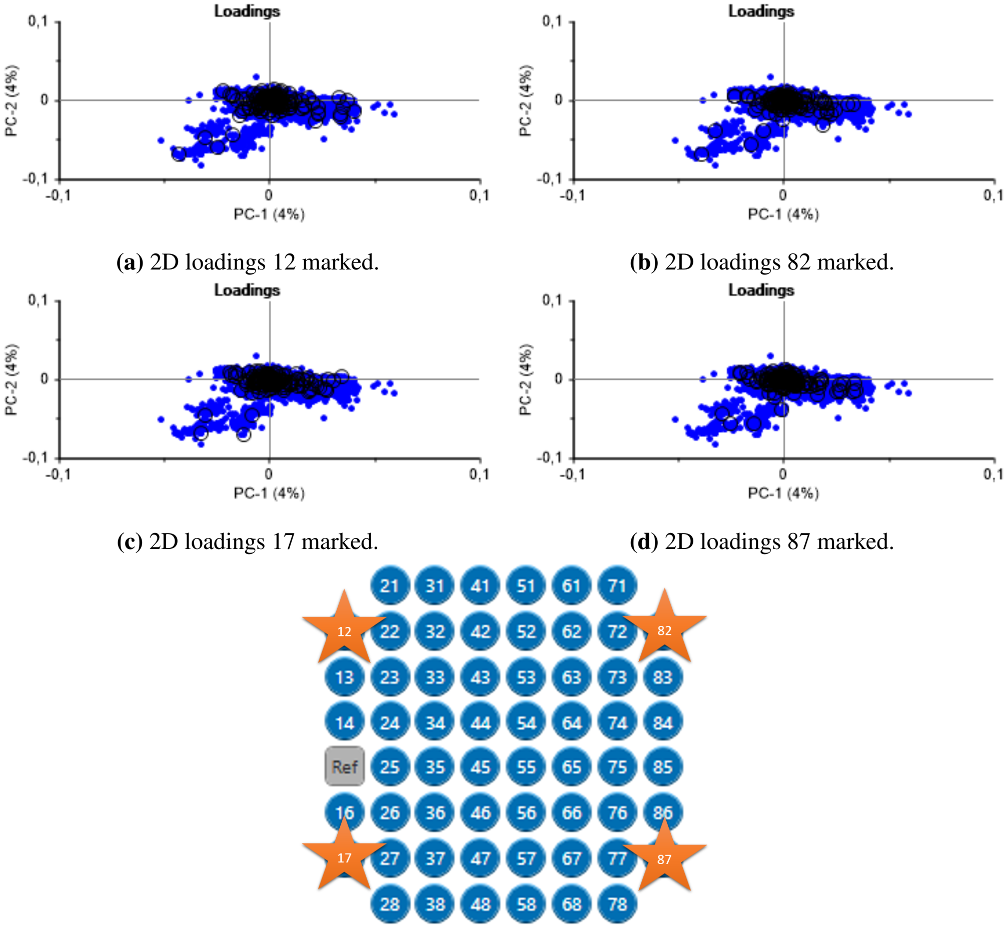
Figure 6.8: Investigating scores, loadings and connection to raw data, **part 3**.



(a) 2017-06-12T11-40-48 (#4). In this model, each cluster seem contain variables that correspond only to a certain frequency bin. Many electrodes have delegated its correct frequency bin to each of the clusters as emphasized in Figure 6.10.

(b) 2018-01-15T09-30-55. The only experiment from the new data set investigated further. In this experiment, PC1 score (60 %) seem to only be cause by a unique behavior from electrode 64, while PC1 from a possible transient on electrode 37.

Figure 6.9: Investigating scores, loadings and connection to raw data, **part 4**.



(a) 2D loadings 12 marked.

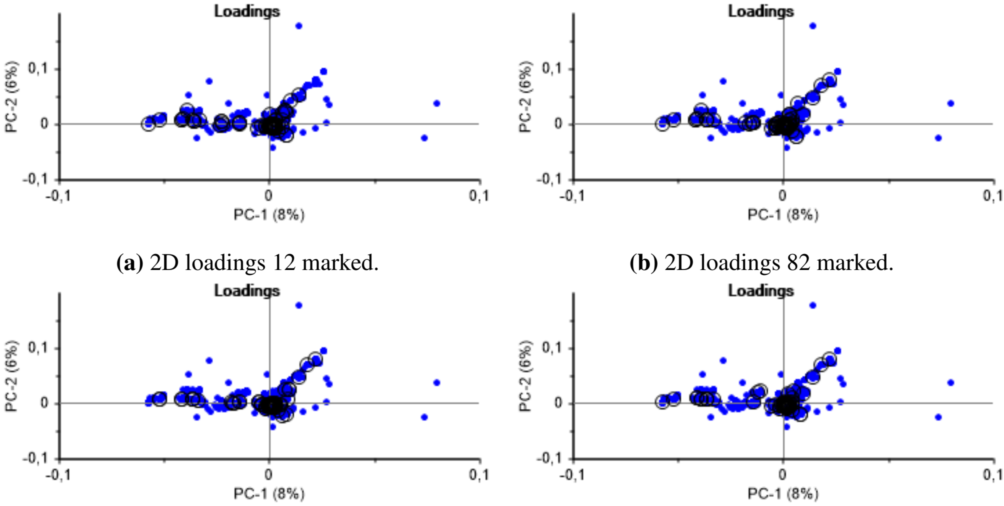
(b) 2D loadings 82 marked.

(c) 2D loadings 17 marked.

(d) 2D loadings 87 marked.

(e) Investigated electrodes (marked in stars) are physically located in each corner of the MEA.

Figure 6.10: Emphasizing that electrode placement doesn't seem to matter for the PCA model of experiment #4. All selected electrodes have exactly one variable in each of the visible (three) clusters in PC1-PC2 loading space.



(a) 2D loadings 12 marked.

(b) 2D loadings 82 marked.

(c) 2D loadings 17 marked.

(d) 2D loadings 87 marked.

Figure 6.11: Same investigation as in Figure 6.10 but for experiment 2017-06-05T12-49-11. Any of the selected electrodes have at least one variable in each of the clusters. The corresponding line loadings for this model in Figure 6.12 show common peaks for the approximate same frequency bin for the selected electrodes.

However, the residual variance (Appendix Chapter 5.4.3) for the projections often does not decrease as the number of PCs are increased. This probably means that the latent variables in the projected data are not expressed in the model. Residual variances for projection are also larger than the calibration and validation residual variance calculations in many of the figures. However, if projection residual variances also decrease as the number of PCs are added, the projected data should still be explained somewhat in the model.

The residual variances for projection are less than the other residual variances in only one projection, 2017-05-22T10-30-07 (data) projected onto 2017-06-05T12-55-34 (#4) (model) score space (Figure 6.15 (a)). 2017-05-22T10-30-07 had stimulation whereas #4 had not (Table 2.2). It could be reasonable to expect that stimulation distorts the projection, even if stimulation electrode was removed to make the model. It doesn't seem to be the case. In all other projections, residual variances (green line) are a lot larger than the ones for the model (red and blue lines). The projected data are very different from the data in the model as also visualized in the influence plot and scores.

Lastly, the total samples after projection are way off the statistical limits both in residual and Hotelling's T^2 in the influence plot in many of the figures. Outside the residual limit gives a hint that the samples are not explained properly in the model, while outside Hotelling's T^2 limit gives indication that the samples are extreme in the model space.

Regarding the explained variance plot, when the accumulated explained variance reaches a plateau as more components are included in the model, it is an indication that noise is being modeled. An additional indication of this is when the validated variance decreases and/or deviates from the calibration variance.

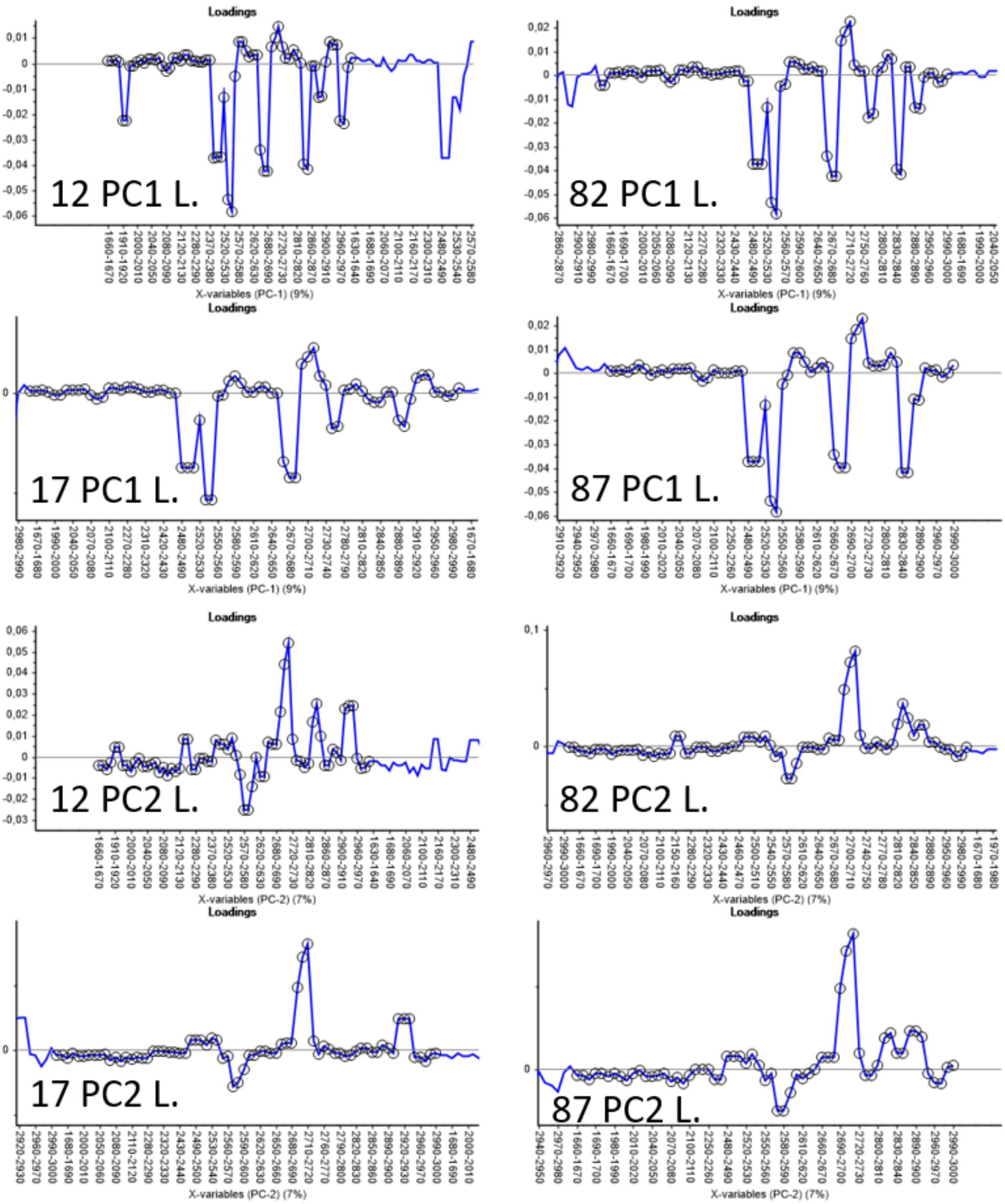
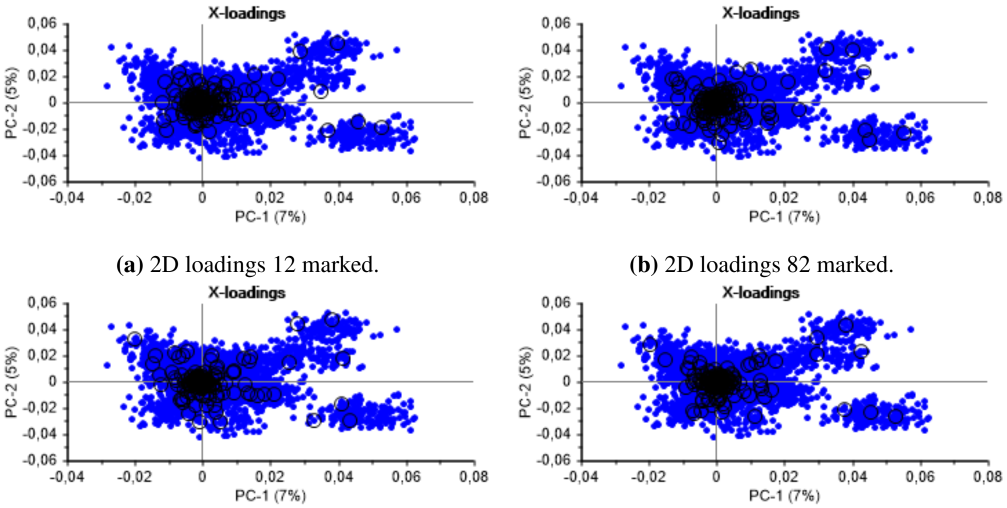


Figure 6.12: The line loadings for the selected electrodes in model Figure 6.11. The peaks occur in approximately the same frequency bins, and lead to clusters in PC1-PC2 loading space.



(a) 2D loadings 12 marked.

(b) 2D loadings 82 marked.

(c) 2D loadings 17 marked.

(d) 2D loadings 87 marked.

Figure 6.13: Same investigation as in Figure 6.10 but for experiment 2017-06-05T12-55-34 .

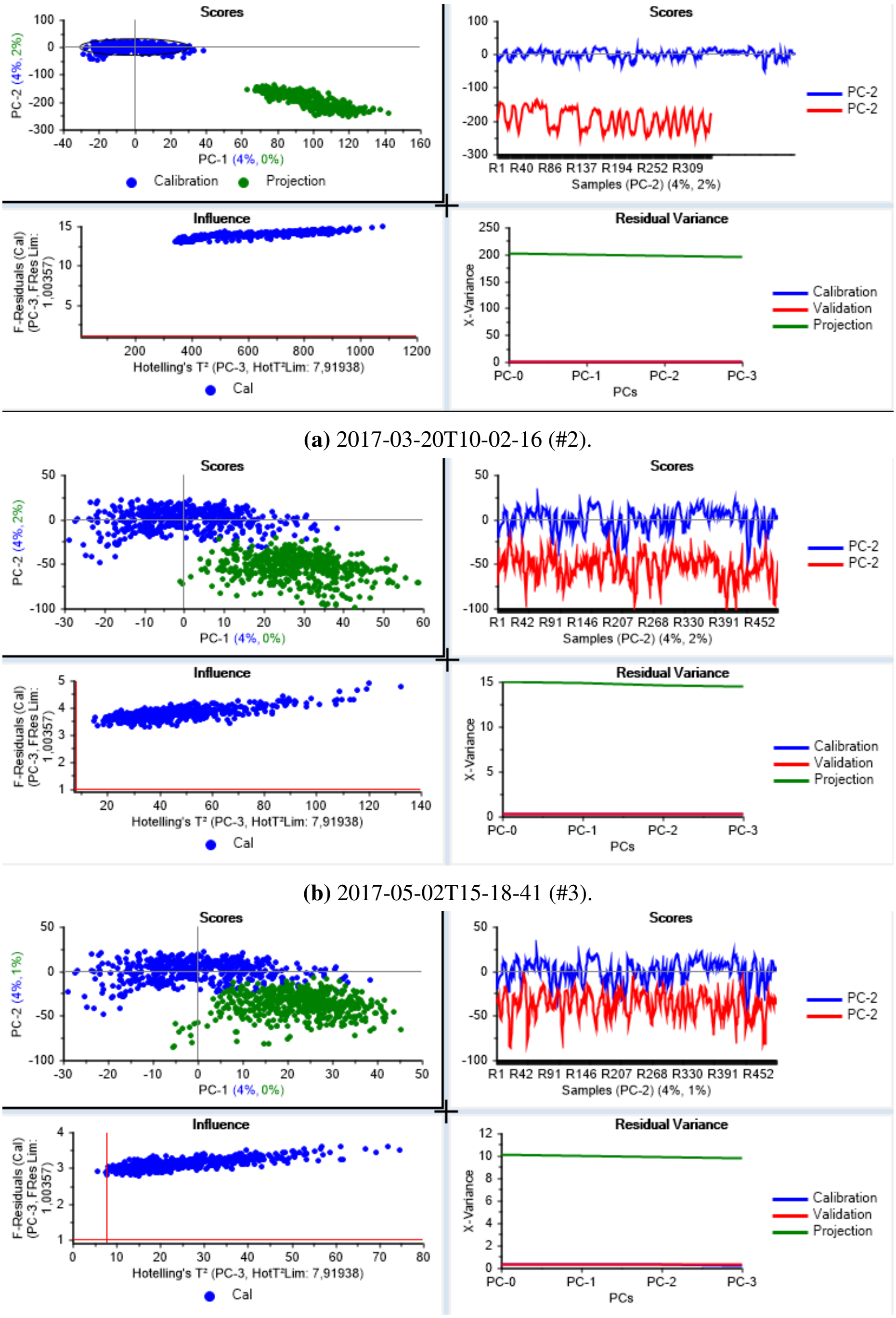
Observations in the last two paragraphs could be indications that a model is no longer stable, but not certain indications. For example, a model from random data will not give any outliers but still the model is not stable.

6.1.5 Clustering in score space in a model of selected real-time preprocessed (RTP) electrode data hints to promising PCA models for projection in a real-time scenario

The cluster formations in loading space in Figures 6.10, 6.11 and 6.13 could be further investigated using real-time preprocessed data (Preprocessing 2). For each experiment 2017-06-05T12-49-11, 2017-06-05T12-55-34 and 2017-06-12T11-40-48 (#4), preprocessed matrices for electrode 12, 17, 82 and 87 were made according to real-time Preprocessing 2, Chapter 4.4, Figure 4.1. However, individual PCA models in each of these electrode matrices did not have similar cluster formations in loading space. This might be because more electrodes might be needed in the PCA model in order to show clusters.

Nevertheless, a PCA model on a combined matrix revealed some new insights about similarities among the selected electrodes across the three experiments in the score space. Each i matrix of the in total $i = 4 * 3 = 12$ matrices with dimension $M_i \times (N = (3000 - 296)/8 = 338)$ were combined into a single matrix $\mathbf{X}$ according to equation 6.1

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_{12} \end{bmatrix} \in \mathbb{R}^{(M_1+M_2+\ldots+M_{12})\times 338} \tag{6.1}$$

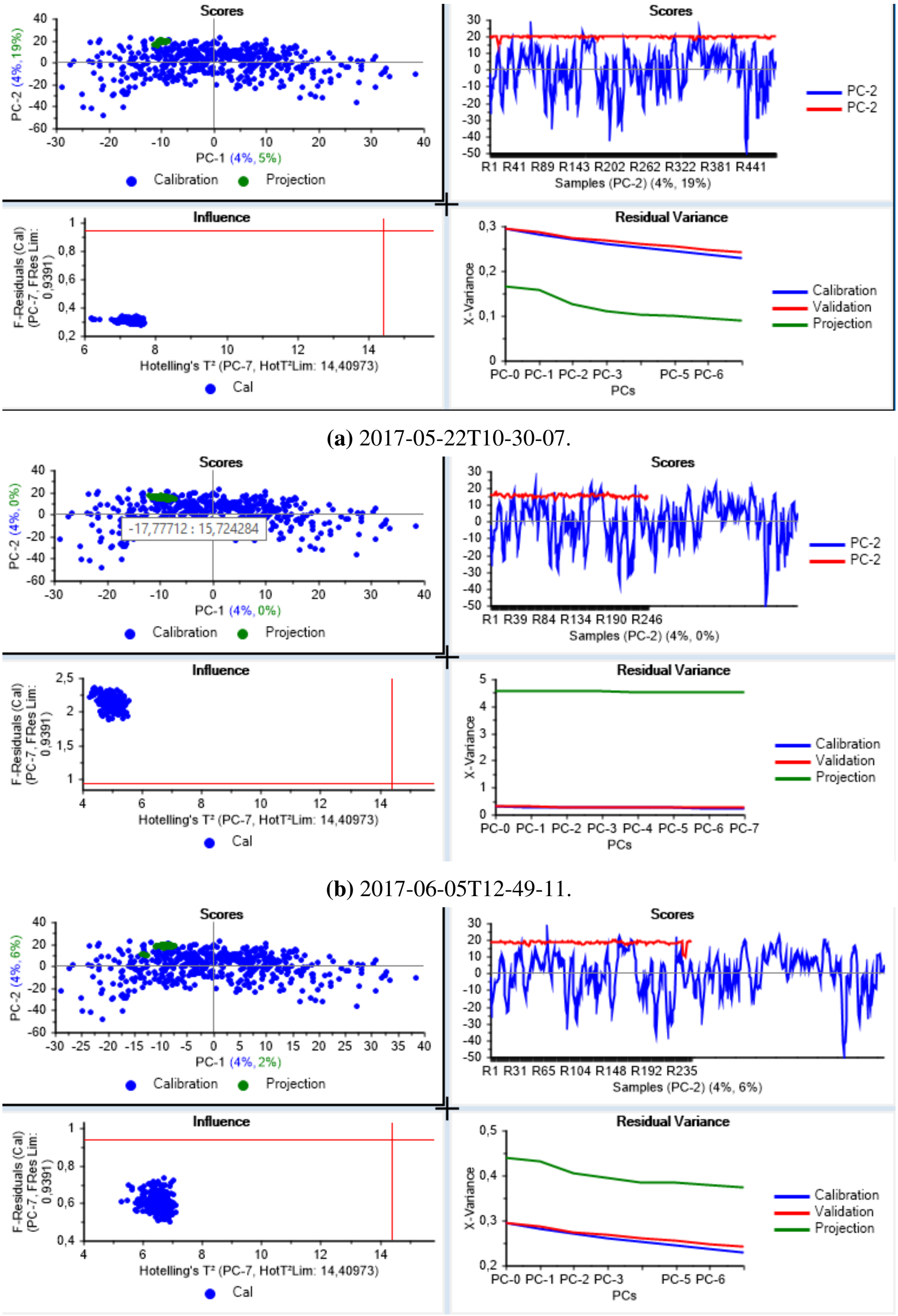


(a) 2017-03-20T10-02-16 (#2).

(b) 2017-05-02T15-18-41 (#3).

(c) 2017-05-15T10-20-04.

Figure 6.14: Projection of data from different experiments onto score space for experiment 2017-06-12T11-40-48 (#4) (old, adult culture), **part 1**.

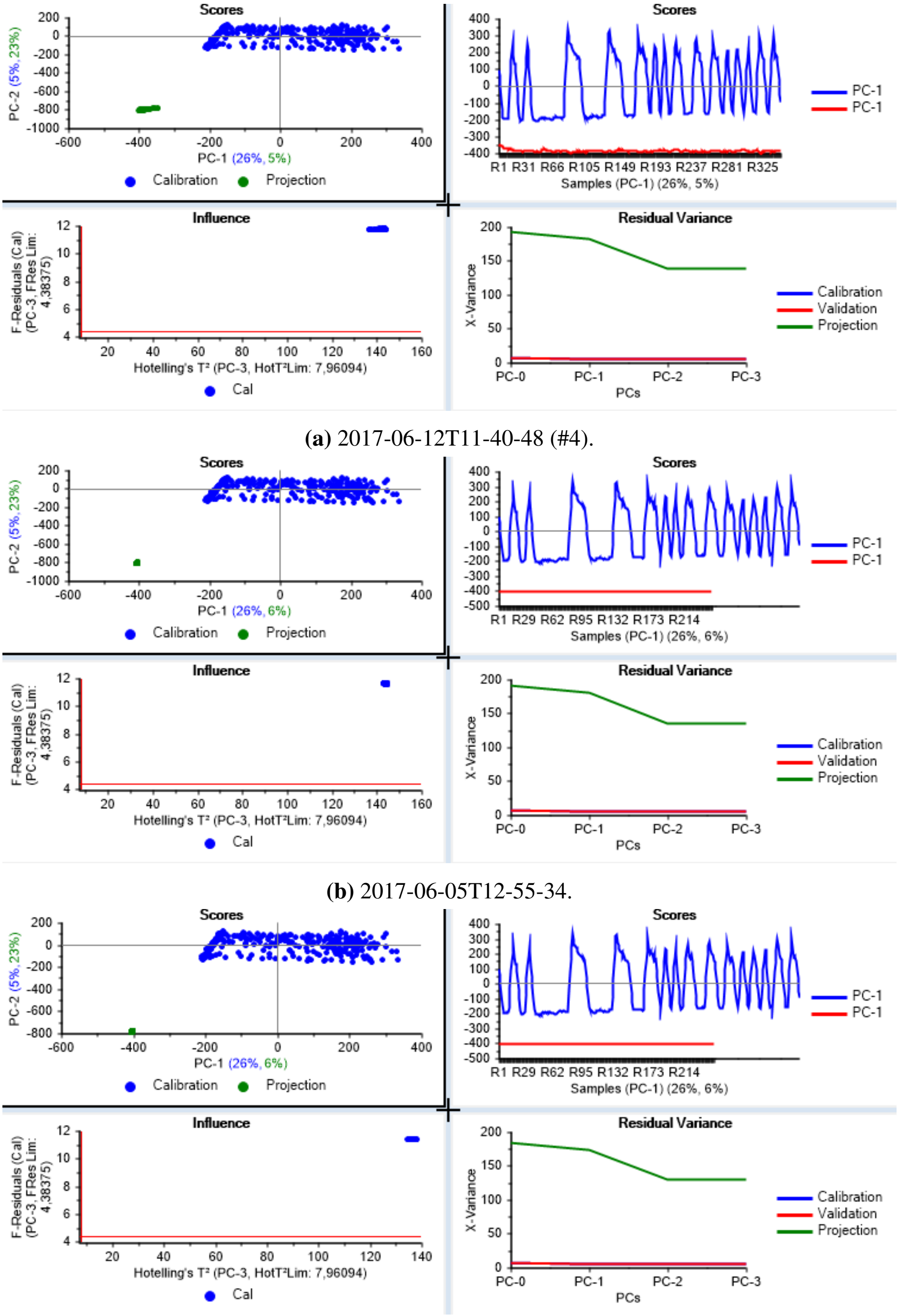


(a) 2017-05-22T10-30-07.

(b) 2017-06-05T12-49-11.

(c) 2017-06-05T12-55-34.

Figure 6.15: Projection of data from different experiments onto score space for experiment 2017-06-12T11-40-48 (#4) (old, adult culture), **part 2**.

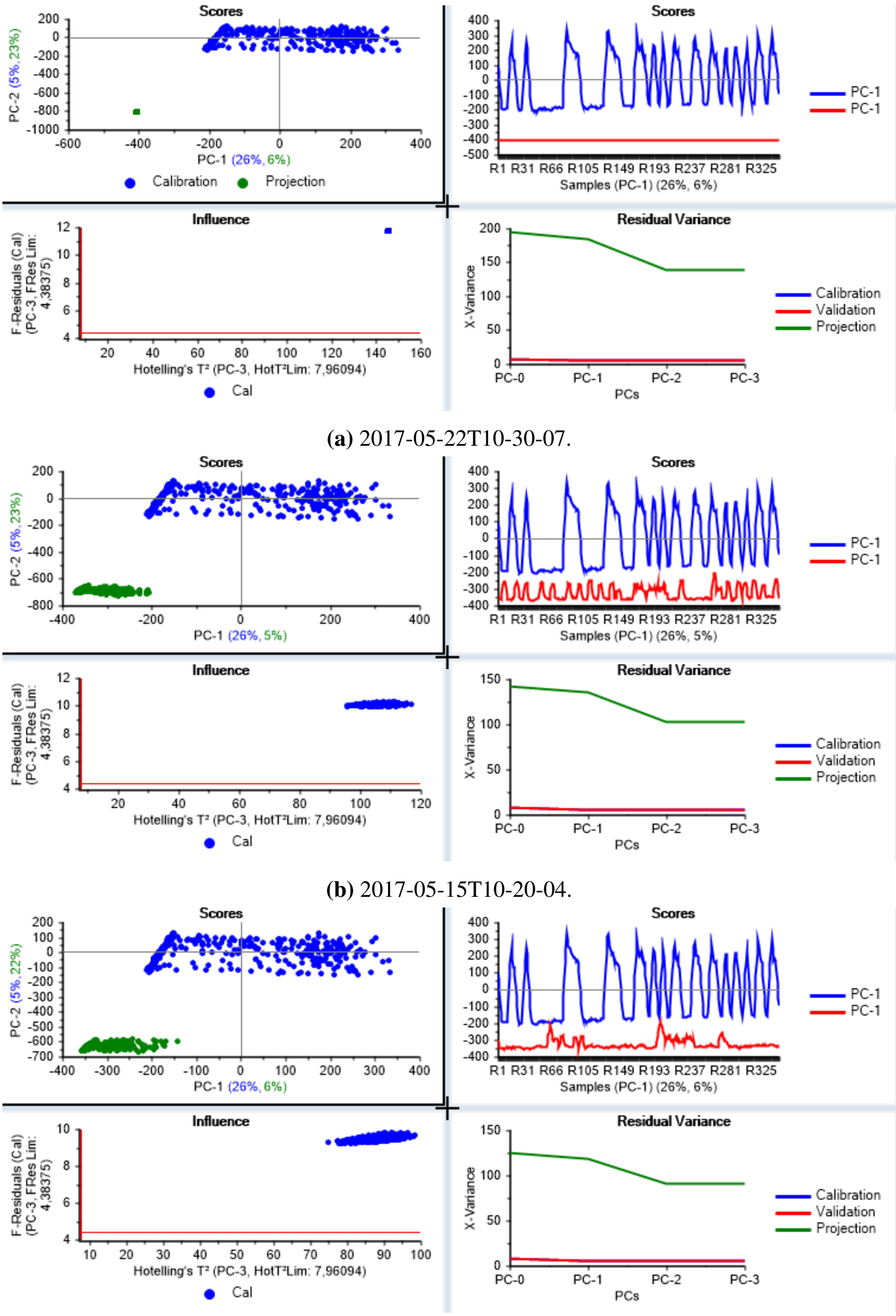


(a) 2017-06-12T11-40-48 (#4).

(b) 2017-06-05T12-55-34.

(c) 2017-06-05T12-49-11.

Figure 6.16: Projection of data from different experiments onto score space for experiment 2017-03-20T10-02-16 (#2) (young culture), **part 1**.



(a) 2017-05-22T10-30-07.

(b) 2017-05-15T10-20-04.

(c) 2017-05-02T15-18-41.

Figure 6.17: Projection of data from different experiments onto score space for experiment 2017-03-20T10-02-16 (#2) (young culture), **part 2**.

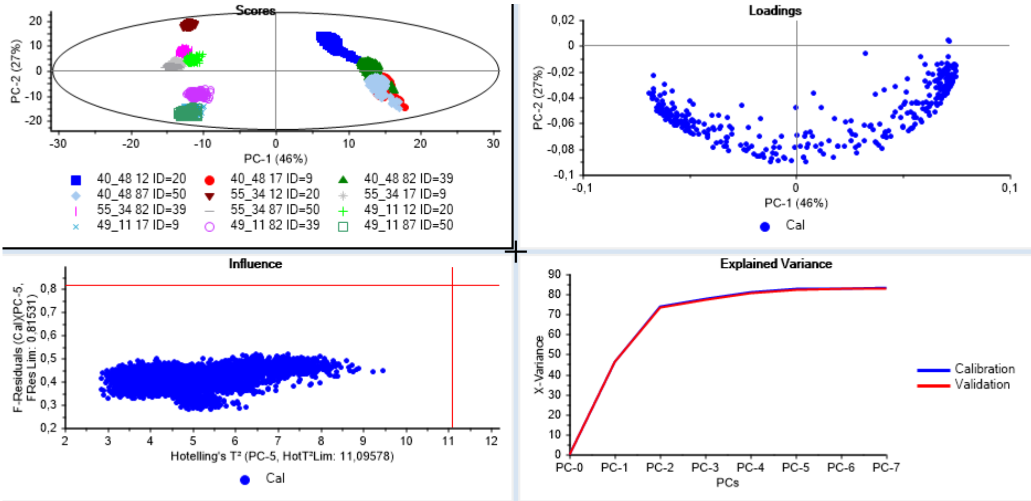


Figure 6.18: Overview PCA on real-time preprocessed electrode 12, 17, 82 and 87 from 2017-06-05T12-49-11, 2017-06-05T12-55-34 and 2017-06-12T11-40-48 (#4) combined using equation 6.1.

7-component PCA on **X** was performed with mean centering, scaling to unit variance and random 20 segment cross-validation. All samples in the PCA models are good below both statistical limits on the influence plot, see Figure 6.18. The groups in score space are the different electrodes, and they are further investigated with different combinations of PCs in Figure 6.19.

Note that opposed to OP, RTP uses individual noise thresholds per frequency bin, per electrode to count an AP. In orther words, the noise thresholds used for an electrode are the same in the preprocessing of that electrode in the three experiments. Distinct noise thresholds per electrode could be an explanation of the PC3-PC4 score clustering in Figure 6.19, where each cluster represents the samples of one of the electrodes across experiments. It is promising that the electrodes group similar regardless of the experiment. Each electrode share the same latent variables (PC3 and PC4) across the experiments, which could be used to track the properties of an electrode across experiments. Perhaps this implies that the model is stable. Projection of samples from a comparable culture onto PC3-PC4 score space of such a model could be interesting in a real-time scenario.

6.2 PLS-DA and (O)PLS-VIP

6.2.1 Successfully discriminating PLS-DA models on (OP) data give insights in differences between experiments

The projections in Chapter 6.1.4 gave some hints on the similarity of experiments. For instance, the projections in Figure 6.14 show that the samples in the experiments are different. Why are they different?

Partial least squares projections to latent structures for discriminant analysis (PLS-DA, Chapter

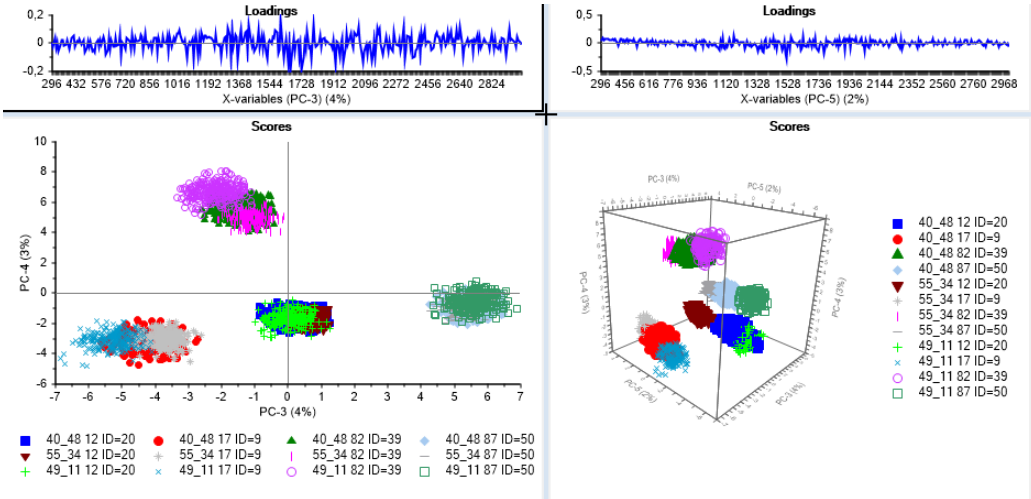


Figure 6.19: Investigation of clusters in score space from the PCA model in Figure 6.18. PC3 and PC4 group electrodes in the same manner regardless of experiment. Each cluster in the lower left plot are not all samples from a single experiment, but samples from the same electrode for all experiments. Hence, there are latent variables that are able to discriminate more than just experiment. PC5 sorts each group of electrodes according to experiment, equally across the different electrodes. PC5 does not correspond to age (the oldest, experiment #4, is in the middle). The x-axis on the two line loading plots are frequency (bins). Making such a model offline, and then using it for projecting real-time (interesting samples from selected electrodes) could perhaps be worth a try in the NTNU Cyborg project.

5.2) was used to further analyze sets of two experiments. The approach introduces a ”dummy” variable $\mathbf{Y}_{PLS}$ that categorizes each sample in a combined matrix $\mathbf{X}_{PLS}$ into either one of two classes depending on which of the two experiments the sample belongs to. In this way, a PLS-DA model can further investigate latent features in $\mathbf{X}_{PLS}$ that are important in order to discriminate the two experiments from each other.

For the offline preprocessed data matrices of a set of two experiments A and B, $\mathbf{X}_A \in \mathbb{R}^{M_1 \times N}$, $\mathbf{X}_B \in \mathbb{R}^{M_2 \times N}$, combined matrices $\mathbf{X}_{PLS}$ and $\mathbf{Y}_{PLS}$ were made for the PLS-DA according to equation 6.2 and 6.3

$$\mathbf{X}_{PLS} = \begin{bmatrix} \mathbf{X}_A \\ \mathbf{X}_B \end{bmatrix} \in \mathbb{R}^{(M_1+M_2)\times N} \quad , \quad \mathbf{Y}_{PLS} = \begin{bmatrix} \mathbf{Y}_A \\ \mathbf{Y}_B \end{bmatrix} \in \mathbb{R}^{(M_1+M_2)\times 1} \tag{6.2}$$

and

$$\mathbf{Y}_A = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{M_1 \times 1} \quad , \quad \mathbf{Y}_B = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \in \mathbb{R}^{M_2 \times 1} \tag{6.3}$$

Prior to modelling, $\mathbf{X}_{PLS}$ and $\mathbf{Y}_{PLS}$ were mean-centered. PLS-DA models were made for both scaled to unit variance (Chapter 5.1) and non-scaled versions of $\mathbf{X}_{PLS}$ and $\mathbf{Y}_{PLS}$, resulting in 6

Data set	Investigation	Conclusion
2017-05-02T15-18-41 (#3), 2017-06-12T11-40-48 (#4)	Stimulation	Discrimination from energy differences. Figures 6.20 and 6.23
2017-06-05T12-49-11, 2017-06-05T12-55-34	Distinct activity across equal ages	Almost all electrodes are important in discriminating. Electrode 35 show some unique behavior Figures 6.21 and 6.24
2017-05-02T15-18-41 (#3), 2017-05-22T10-30-07	Age	Age discrimination is based on energy differences. Can be an effect of long-term stimulation. Figures 6.22 and 6.25

Table 6.1: Summary of PLS-DA analysis.

PLS-DA models. (The previous PCA models were not scaled to unit variance). All 7-component PLS-DA models were generated using NIPALS (Nonlinear Iterative Partial Least Squares, Appendix 5.2) algorithm for finding the eigenvectors, and random cross-validation with 20 segments was used in model training and testing. It is noted that 7 was a good amount of more components than needed for discriminating, especially for the PLS-DA models with non-scaled data. The first component is correctly used to classify experiment samples in all PLS-DA models, but some models reveal additional latent features as well.

The PLS-DA models on sets of two experiments are presented in Figures 6.20, 6.21 and 6.22 for scaled matrices and in Figures 6.23, 6.24 and 6.25 for non-scaled variants. Electrodes belonging to variables that were manually removed from the data sets prior to modeling are specified in the figure descriptions. Additionally, Unscrambler automatically kept out a varying amount of additional variables prior to modeling (columns expected to have all 0s). Table 6.1 contains a summation of the results from the PLS-DA analysis.

6.2.2 Influential variables in the PLS-DA models - variable influence on projection (VIP) - further describe differences between experiments

The author had the opportunity to try out a method for investigating influential variables in the PLS-DA models. The variable influence on projection (VIP) method was presented by (Galindo-Prieto et al., 2014) (as OPLS-VIP), but was adapted to PLS-DA in this case. For each PLS-DA model, the importance of the frequency bins of all electrodes (variables of the PLS-DA model) was evaluated by calculating their variable influence on projection (VIP) values (Method explained in Chapter 5.5).

Figure 6.26 gives a summary of the VIP plots. The VIP plots for the non-scaled PLS-DA models are very similar to the loadings plot for first component in the respective PLS-DA models. VIP plots on scaled data PLS-DA models show similar trends of electrodes, but are less exaggerated. An important observation is that lower frequency bins seem to be more important than higher

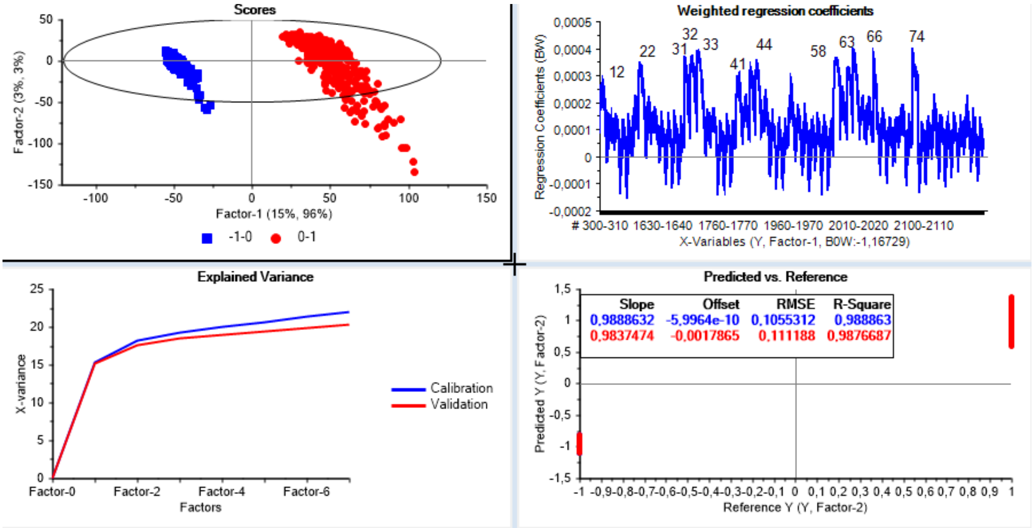


Figure 6.20: PLS-DA 2017-05-02T15-18-41 (#3) (red circles, A) and 2017-06-12T11-40-48 (#4) (blue squares, B) scaled to unit variance. A young and old experiment where stimulation seems to be important in discrimination (74). Explained variance from calibration and cross-validation diverge slightly, so no optimal model exists. Three components seems to be a reasonable model dimension. Weighted regression coefficients (that are used for predicting $\mathbf{Y}$) show that electrodes previously known to have high energy (not much spiking activity, see Figure 2.12), are important in discriminating between the two experiments. Weighted regression coefficients are regression coefficients working on mean centered and scaled (to unit variance) data. Weighted regression coefficients for unit scaled data are independent of the variance in the original variables. #3 had stimulation on electrode 74 whereas #4 had no stimulation. It is thus proposed that the PLS-DA model discriminates based on energy differences from with and without stimulation. However, from results in Figure 6.22, stimulation might not (only) be the discriminating factor but also age.

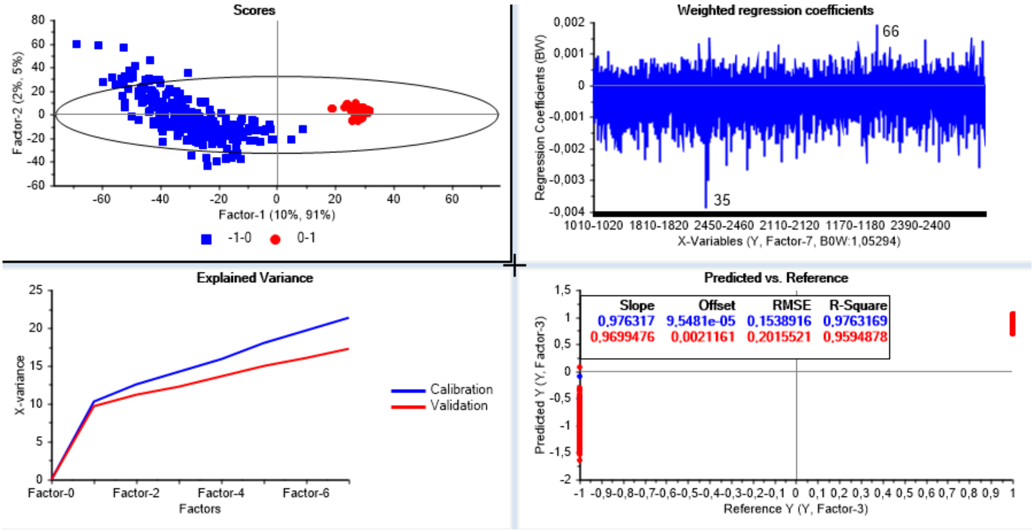
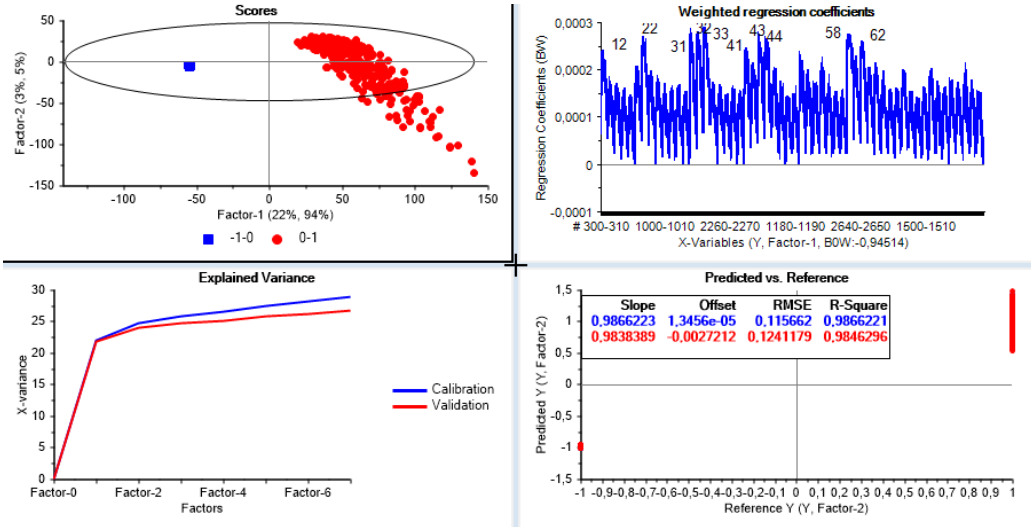


Figure 6.21: PLS-DA 2017-06-05T12-49-11 (red circles, A) and 2017-06-05T12-55-34 (blue squares, B) scaled to unit variance, 31 and 74 removed from data set. Two experiments recorded only a couple of minutes apart (same day/age). The least optimal model and the one being most vulnerable to overfitting by fast diverging calibration and variance measures in explained variance plot. However manages to discriminate with first component. It is interesting that the samples can still be correctly separated/discriminated according to experiment. Even more interesting are the variables belonging to electrode 35, where many of them have very negative weighted regression coefficients when all 7 components are used. Electrode 35 is known from earlier to have a lot of spiking activity, even perhaps the electrode with the most synchronized bursts from younger versions of the culture (Figures 2.13 (a), 6.6 and 6.7). This PLS-DA shows that this very same electrode might play an important role in discriminating between the experiments when more components are added to the model.



Slope	Offset	RMSE	R-Square
0,9866223	1,3456e-05	0,115662	0,9866221
0,9838389	-0,0027212	0,1241179	0,9846296

Figure 6.22: PLS-DA 2017-05-02T15-18-41 (#3) (red circles, A) and 2017-05-22T10-30-07 (blue squares, B) scaled to unit variance, 63, 66 and 74 removed from data set. A young and old (not as old as in Figure 6.20) experiment where age seems to discriminate. Three components seem to be a reasonable model dimension, and risk of overfitting if the number of components are increased. Opposed to the PLS-DA model in Figure 6.20, both experiments received the same stimulation. However, weighted regression coefficients are very similar in both PLS-DA models (74 was removed here). Since the same stimulation was applied on all experiment except #4, it might be reasonable that the common regression coefficients in this figure and Figure 6.20 are important for predicting $\mathbf{Y}$ based on its age, and not directly stimulation. However, stimulation might have been important in formulating the age. Many of the electrodes for variables that have large weighted regression coefficients were also previously shown to have high variance in energy (PSD over frequencies) but little variance in number of detected spikes (ISI-histograms). In the project work, it was suggested that large variance in energy activity was an effect of the stimulation. Hence, stimulation might have had an intrinsic importance in making the culture learn energy behavior that can be used to determine the age of the experiment. This learnt behavior is not in the form of spiking activity, but in overall measured electrical energy. The two measures might describe the same overall development of the culture, but it might also be cause by electrode measurements that have generally become more noisy along with age.

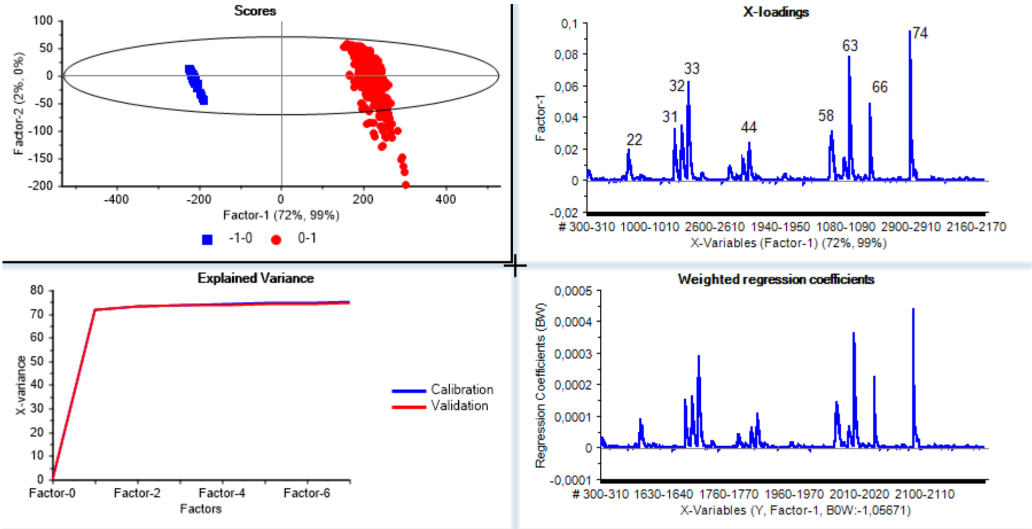


Figure 6.23: PLS-DA 2017-05-02T15-18-41 (#3) (red circles, A) and 2017-06-12T11-40-48 (#4) (blue squares, B) non-scaled matrices. Better discrimination than in Figure 6.20. Two components is optimal model dimension, and no signs of overfitting. Non-scaled matrices lead to very similar weighted regression coefficients and loading for component 1. For only the first component, it can be seen that electrodes that have an overall high variance in energy (high PSD variation from project work) contribute the most to separating the experiments.

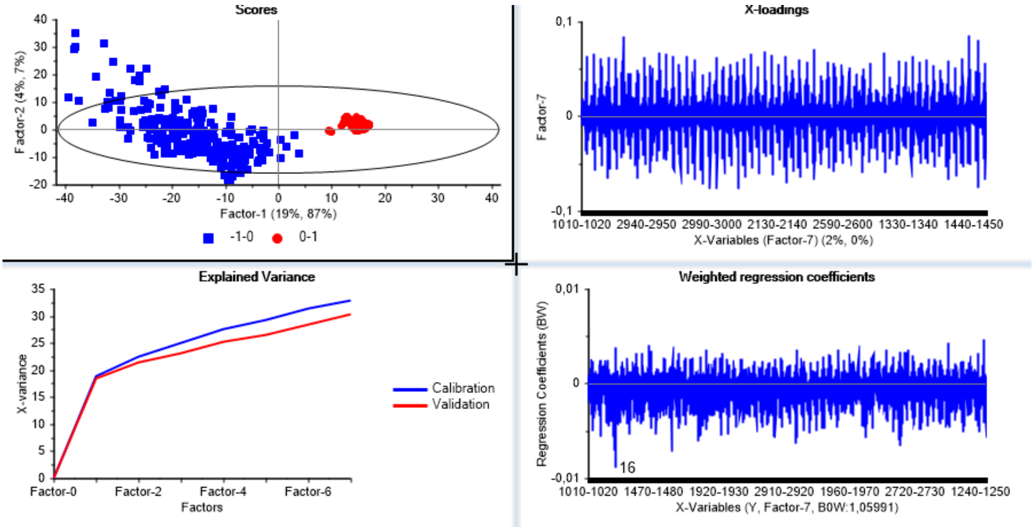


Figure 6.24: PLS-DA 2017-06-05T12-49-11 (red circles, A) and 2017-06-05T12-55-34 (blue squares, B) non-scaled matrices, 31 and 74 removed from data set. Two experiments recorded only a couple of minutes apart (same day/age). Not any significant improvement in discrimination, but less risk of overfitting than the scaled data model version in Figure 6.21. Perhaps two components is the best model dimension. Opposed to Figure 6.21, 35 is no longer showing distinct behavior. 16 has the most negative regression coefficients.

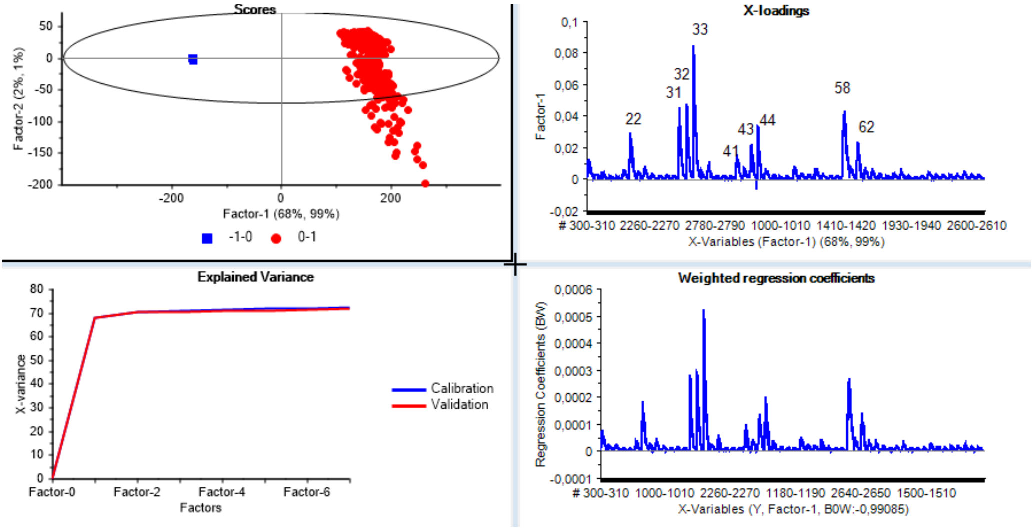


Figure 6.25: PLS-DA 2017-05-02T15-18-41 (#3) (red circles, A) and 2017-05-22T10-30-07 (blue squares, B) non-scaled matrices, 63, 66 and 74 removed from data set. Better discrimination than in Figure 6.22. Optimal model dimension is two components. As in Figure 6.23, shows that significant electrodes in the discrimination are many of the same electrodes to have an overall high variation in energy in previous analysis.

frequency bins for discriminating the experiments in the PLS-DA models that have experiments with large distance in age, while for both PLS-DA models on the two experiments with near equal culture age, higher frequency components are slightly more influential than lower frequency components in discriminating in the models. Again, the bins are in the range $300-3000$ Hz. Note that for Figure 6.26 (c), stimulation variables for stimulation electrode as well as for some other electrodes were kept out (see Figure 6.22 description). Some additional bins were kept out as well, depending on the PLS-DA model (a varying amount of automatically kept out variables occurred the PLS-DA NIPALS algorithm in Unscrambler, and the VIP plots were made for comparison to these models).

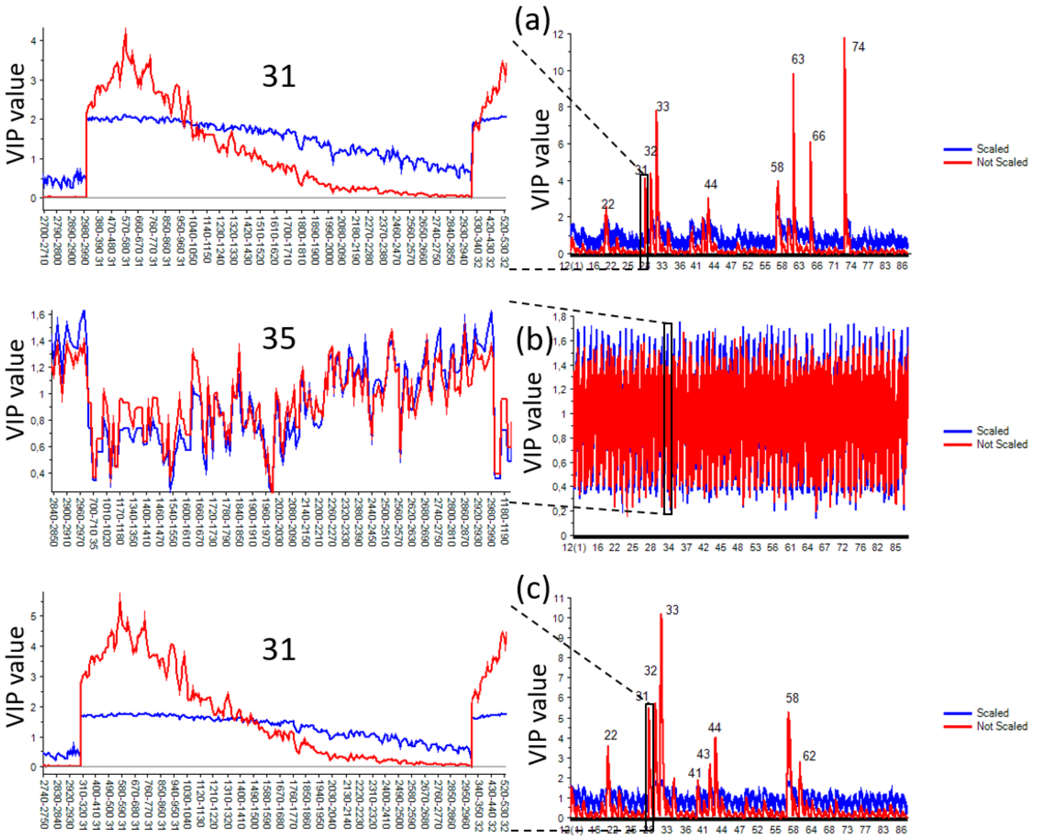


Figure 6.26: Summary of the 6 VIP plots, based on scaled to unit variance and non-scaled data PLS-DA models of the experiments **(a)** 2017-05-02T15-18-41 (#3) and 2017-06-12T11-40-48 (#4), **(b)** 2017-06-05T12-49-11 and 2017-06-05T12-55-34 and **(c)** 2017-05-02T15-18-41 (#3) and 2017-05-22T10-30-07. Of the investigated frequency range 300 − 3000 Hz, lower frequency components (around 700 Hz) seem to be more influential in discriminating between the cultures with large age difference than higher frequency components. Furthermore when discriminating two experiments with same age, higher frequency components (roughly 2000 Hz) are more influential than the lower frequency components. This is further displayed in Figure 6.27 using (a) and (b).

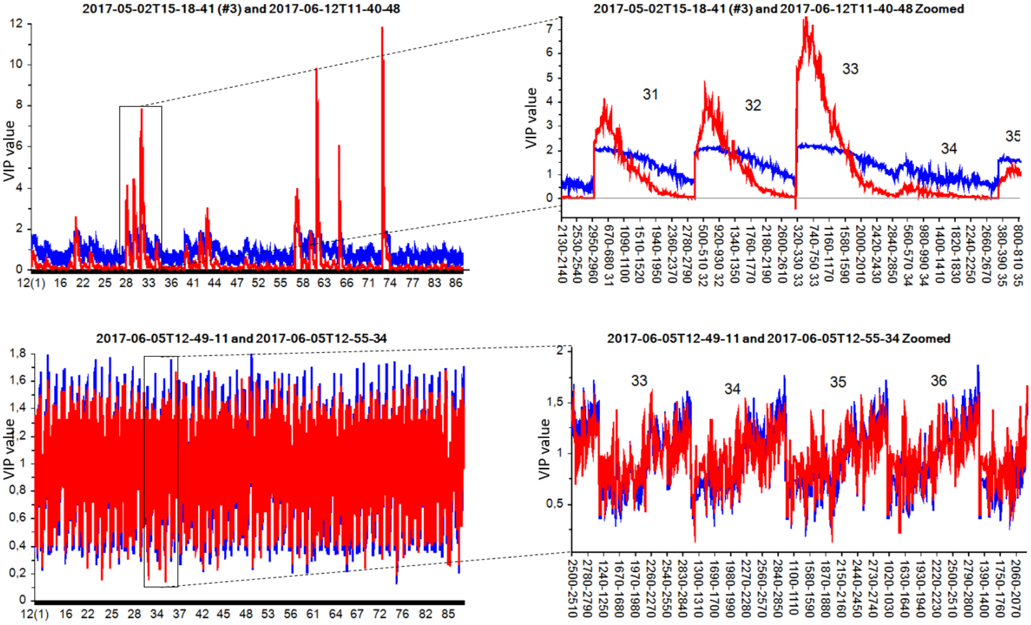


Figure 6.27: Investigating long term (age) discriminating and short term (spontaneous activity) discriminating frequency bins in VIP plots based on scaled to unit variance (blue) and non-scaled (red) data PLS-DA models of the experiments 2017-05-02T15-18-41 (#3) and 2017-06-12T11-40-48 (#4) (upper two plots), and of 2017-06-05T12-49-11 and 2017-06-05T12-55-34 (lower two plots). Higher frequencies are more influential in classifying experiment samples to correct experiment if they are from a culture with the same age. For two experiments with large difference in age, lower frequencies are more influential in separating the experiment samples in a PLS-DA model. VIP plots from non-scaled data PLS-DA models show clearer influential low frequency components if the experiments have different ages. If the age is similar, VIP plots from scaled data and non-scaled data PLS-DA models show the same trend, which is that higher frequencies are more influential than lower frequencies in the discriminatory PLS.

Chapter 7

Discussion

The preprocessing has been a vital part of this work, and assumptions have been taken based on what the author wanted to look for in the data. This might lead to a preprocessing that is not working correctly on other types of neural data. The MVA models lead to some open questions which will be discussed in this chapter.

7.1 Neural envelope frequency need to be $<\approx 8$ Hz for OP and $<\approx 6$ and $<\approx 20$ Hz for RTP 1 and 2 respectively

It is assumed that for other types of neurons than the ones analyzed, neural oscillations in a frequency bin in the spectrogram could be very different (f. ex. higher frequency oscillations). In the young MEA2 culture (for example experiment #2, Figures 2.13 (a) and 2.15), the oscillations were typically around 3 Hz), while in adult culture (for example #4, Figures 2.13 (b) and 2.17) around approx. $0.3 - 8$ Hz). The OP was designed based capturing activity in in the ranges spanned by #2 and #4 together. Thus the OP will probably not work correctly on cultures of other types of neurons with radically different oscillations in the frequency bins than in #2 and #4, f. ex. envelope frequencies above 8 Hz.

For the offline preprocessing (OP), the frequency limit for counting APs comes from hard-coded parameters in the noise reduction plug-in in Audacity. In each of the frequency bins, the Fourier based filter has hard-coded an attack time of 20 ms, and decay time of 100 ms. This is called time smoothing in its algorithm, again see description on how it works and source code). The fixed attack and decay parameters imply that the fastest envelope frequency correctly picked up in the noise reduction algorithm is $1/(120 * 10^-3) \approx 8$ Hz.

The real-time preprocessing (RTP) 2 does not have such an upper envelope frequency limit and should correctly count any frequency bin oscillation below $40/2 = 20$ Hz in the 40 Hz PSD stream. However, the low pass filter in RTP 1 was set to have a cut-off frequency of 6 Hz, so RTP 1 has a similar constraint to the OP. This cutoff frequency was based on testing the filter on an artificial 3 Hz oscillation simulating a burst. Hence, RTP 1 was designed to best function on data of the young culture.

7.2 Are envelope frequencies comparable to EEG bands and estimated frequencies from interspike-intervals?

The neural firing creating the oscillations in the frequency bins in this Masters's thesis has been called APs. However, these oscillations might be better considered oscillations in the local field potential (LFP), since APs are typically found on the millisecond scale in neuronal data (APs (EAPs) should more correctly be the small spikes with high amplitudes occurring together with larger LFP oscillations as in raw data of electrode 35 in Figure 6.7 (b)) (See Chapter 2.2).

Recent research provide evidence that LFPs have an important role in spike timing from traditional APs in human temporal cortex (Zanos et al., 2012) and monkey motor cortex (Denker et al., 2011). The question is whether the timing information captured in this prepossessing is comparable to timing information in traditional APs. As explained in the Chapter 2.2, single-unit activity (SUA) and multi-unit activity (MUA) are regarded as discrete spiking events from APs coming from activations of single and populations of neurons respectively. They are extracted by high-pass filtering (> 300 Hz) a MEA signal and then performing classical spike detection by thresholding (Appendix Chapter 8.1). ISI histograms can then be constructed from the SUAs and MUAs in order to aggregate timing information in the form of shapes of the histograms.

Since the preprocessing works on data in the frequency range $\approx 300 - 3000$ Hz, perhaps the timing information extracted from counting activations in the frequency bins are comparable to timing information captured in ISI histograms? Spiking frequencies from traditional APs have been typically grouped in the EEG bands for EEG signals (delta ($0.5 - 4$ Hz), theta ($4 - 8$ Hz), alpha ($8 - 13$ Hz), beta ($13 - 30$ Hz) and gamma ($30 - 120$ Hz), Chapter 2.3). EEG signals are typically sampled with something like 200 Hz sampling rate, so higher frequency oscillations than the EEG bands have not been available for analysis, opposed to when analyzing neuronal populations with MEAs with a much higher sampling rate (at least 10000 Hz, frequencies up to 5000 Hz can be analyzed).

Lastly, estimated frequencies from ISI histograms can be in a comparable frequency range to EEG bands ($1.005 - 200$ Hz in the project work). The point is that, if the types of neurons analyzed are roughly comparable (f. ex dopaminergic neurons) across the different preprocessing methods, high power oscillations in any of the frequencies from envelope frequencies, spiking frequencies grouped in EEG bands or estimated spiking frequencies from ISI histograms, could likely carry much of the same information. MEA recordings are considered electrophysiological extracellular recordings of local field poentials (LFPs), extracellular action potentials (EAPs) and summations of EAPs, and all frequency interpretations are based on these signals. (Obien et al., 2015; Waldert et al., 2009) (Chapter 2.2). It is thus proposed that spiking frequencies from traditionally detected spikes from APs, estimated spiking frequencies from ISI histograms and envelope frequencies in this preprocessing represent roughly the same neural oscillations and are thus comparable for a given MEA recording.

7.3 Preprocessing bias from selection of noise segment in OP and noise thresholds in RTP

Only a small segment of noise between third and second burst of electrode 87 in experiment #2 (young culture) was used as the noise profile for noise reduction in OP. This is obvious a weakness of the OP, and could lead to less or more than necessary noise reduction on other electrodes that have a different noise profile. However, the OP noise reduction can be easily improved by more carefully construction of a noise segment. For example, multiple noise segments/types from various electrodes could be concatenated into one noise segment to use as the noise profile.

In the RTP case, the selection of a proper MEA culture with visible noise segments on most electrodes is very important in order to construct good noise thresholds per electrode PSD. In this work, the entire experiment #2 was used to make the PSD noise thresholds. The type of activity in this culture should be as similar as possible to the activity in the cultures that is going to be streamed for real-time analysis.

7.4 Other differences between OP and RTP

It is worth mentioning that the sound conversion together with the noise reduction steps in OP lead to less differences in amplitudes in bins in STFT than in the PSD stream in RTP. It is thus proposed that there can be more amplitude information in RTP than OP.

Also, RTP has 8 Hz frequency resolution in the spectrogram, whereas OP has 10 Hz.

7.5 Why are cultures so different in projections and PLS-DA models?

Chapter 6.1.4 presented projections of OP data for one experiment into the score space in the PCA model of another experiment, for multiple combinations of experiments. The new residuals and Hotelling's distances with the projected samples in the model varied significantly over the sets of experiments. This raises questions of whether the PCA models are comparable at all.

There can be multiple reasons for why two experiments are so different that samples from the one experiment cannot be projected onto the model of the other. One cause could be differences in the preprocessed data (ex. different levels of noise noise in raw data remaining after noise reduction). More interesting causes could be the difference in age of the culture between the experiments, or whether stimulation was applied or not (experiment #4 was the only experiment without the same stimulation).

Nevertheless, comparisons of line scores for the projected samples (red line) and model samples (blue line) in projections such as Figure 6.17 (b) show that some models share PCs that are latent variables across experiments. In the case of Figure 6.17 (b), the latent variable seem to express the synchronized bursting in the young culture. In some other projections, the projected samples span very little variation in the score space (f. ex. Figures 6.17 (a) (model of young culture) and

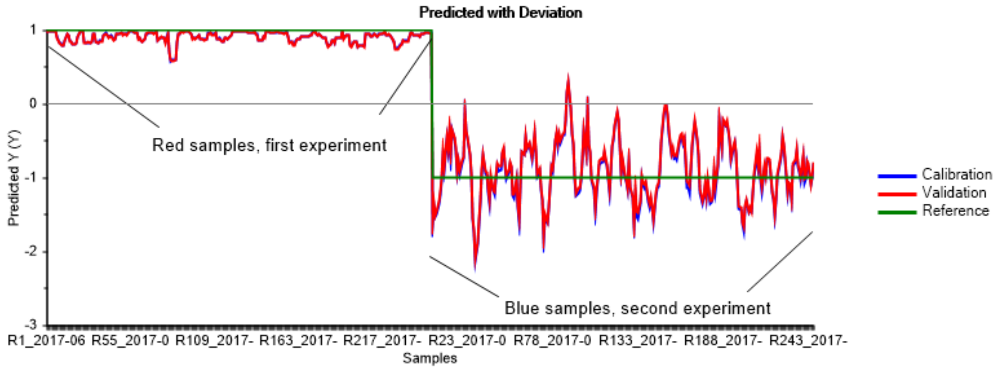


Figure 7.1: Prediction of $\mathbf{Y}$ in PLS-DA of 2017-06-05T12-49-11 (red samples) and 2017-06-05T12-55-34 (blue samples) for the first factor. See Figure 6.21.

6.15 (a) (model of old culture)). A likely reason for much of this behavior is that the **data in a PCA model was not scaled no unit variance. It was done on purpose from the fact that it gave meaningful PCA models of individual experiments with the type of preprocessing used. There was no reason to scale the data when it originated from a single experiment**. Mean centering was applied.

The PLS-DA models for cultures with large differences of age were very sensitive to whether or not $\mathbf{X}$ and $\mathbf{Y}$ were scaled to unit variance. Without scaling, loadings of the first component revealed important electrodes in the discrimination, while with scaling much of the same information about important electrodes were shown in weighted regression coefficients instead. However, this was not the case for the PLS-DA models with the set of cultures with near equal age. VIP plots of the models further showed much of the same influential variables as was seen in loading plots of unscaled PLS-DA models.

For the PLS-DA models in Figures 6.20, 6.23, 6.22 and 6.25, it is not very surprising that discrimination is accomplished. All models use two experiments that were recorded approximately a month in between, and differences between the experiments such as age, spiking and energy activity progression, stimulation (for Figure 6.20) or different MEA preparing conditions or instrument calibration would occur more likely.

Nonetheless, the two experiments in the PLS-DA models in Figures 6.21 and 6.24 were only recorded a couple of minutes apart, and it is thus reasonable to assume that at least some latent factors express the same structure/property across these experiments. Still, like the other PLS-DA models, only the first factor is required to separate the experiments. The prediction in of $\mathbf{Y}$ using factor 1 in the scaled PLS-DA version is displayed in Figure 7.1, and 3D score plots and weighted regression coefficients for the first three factors/components are displayed in Figure 7.2.

Still referring to the PLS-DA models of the experiments with culture of equal ages (Figures 6.21, 6.24), the blue experiment is from a culture that has just been recorded already for five minutes (for making the red experiment). Comparison of Figures 6.21 and 7.2 with results from earlier MVA points in a direction that more advanced correlations in frequency bins occurred in the two experiments (Figures 6.2 (c), 6.8 (a), 6.11 (red experiment) and 6.3 (a), 6.8 (b), 6.13 (blue experiment)). More samples are within Hotelling's limit in the blue experiment than in the red

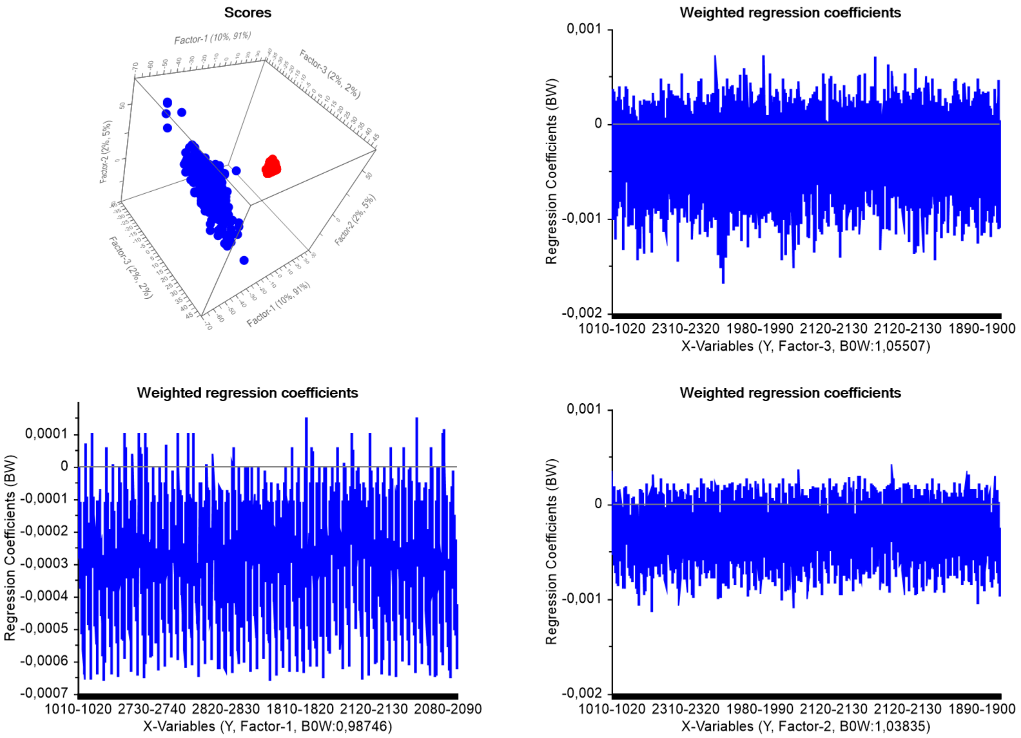


Figure 7.2: 3D score plot and weighted regression coefficients for three first factors of the PLS-DA model of 2017-06-05T12-49-11 (red samples) and 2017-06-05T12-55-34 (blue samples). See Figure 6.21.

experiment in the individual PCA models, so the PCA model of the blue experiment might be more stable than that of the red.

Electrode 35 has shown overall much spiking activity in earlier analysis on the young culture (Figures 6.6, 6.7, 2.12 (Hovden, 2017)) and it is interesting that many of the frequency bins (the variables) belonging to the same electrode have large weighted negative regression coefficients starting to be expressed from factor 3 and higher (Figures 6.21 and 7.2). This is a latent feature well expressed in the blue experiment but little expressed in the red experiment (See Figure 7.2, PC3 score in 3D plot).

Despite this, the PLS-DA model on the non-scaled data (Figure 6.24) do not point out electrode 35 in any way in the regression coefficients or loading plots, and the discrimination is just as good. Note however, as commented in Figure 6.20 that weighted regression coefficients for unit scaled data are independent of the variance in the original variables (not idependent of the variance in the original variables when data is not unit scaled, as in Figure 6.24). This might be the reason that 35 is not visible as special electrode in this model. As with the scaled data PLS-DA model, almost every electrode have variables that contribute similar to each other. Hence, there must be some common difference between the two recordings that are observable on all electrodes. VIP plots show that higher frequency components are influencing the discrimination in the PLS-DA model more than the lower frequency components. **It is beyond the knowledge of the author to discuss the possible biological factors that make it possible to discriminate the experiments in this case (Figures 6.21, 7.1, 7.2, 6.24)**. In the other two cases, it is more evident what can be the latent variable(s) used in discrimination. **It is probable that the overall long-term change in electrical power in the culture is the physical interpretation of the latent variable(s) used in discrimination. It might be used as a measure of the age of the culture (Figures 6.20, 6.23, 6.22, 6.25, 6.26 (a) (c) and 6.27 top two figures**.

7.6 Are synchronized bursts a sign of young culture?

In expert terminology, the synchronization in a burst can also be regarded as a form of coherence. Spike trains that form of an effect of repetitive bursting with many second pauses can also also be called a neuronal avalanche. Coherent avalanches could in many cases be related to synaptic plasticity, in other words the general development of the neural network. (Gireesh and Plenz, 2008) claimed from in-vivo and in-vitro MEA experiments on rat cortical layer 2/3 that neuronal avalanches organize as nested theta and beta/gamma oscillations during development of cortical layer 2/3. The correspondence between nested oscillations and neuronal avalanches required activation of the dopamine D1 receptor. (Shew et al., 2009) reported from MEA experiments on coronal slices from rat somatosensory cortex that such cortical networks that generate neuronal avalanches benefit from a maximized dynamic range, i.e., the ability to respond to the greatest range of stimuli. Lastly, (Benchenane et al., 2010) reported coherent theta oscillations between hippocampus and medial prefrontal cortex on in-vivo microdrive-array tetrode recordings of rat learning a maze.

The above research might not be relatable at all to the cell culture analyzed, but the reward related dopamine is often relatable in tasks such as reward-based learning. As described in the background Chapter 2, the neuronal cells analyzed should correspond to a type of dopaminergic neurons found in substantia nigra pars compacta (SNc), which is a part of substantia nigra, which is located in the human midbrain. The core pathology in Parkinson's disease is the degeneration of the dopaminergic neurons in the SNc of the midbrain that project to the striatum. The latter is the major portal of the basal ganglia, receiving inputs from the cerebral cortex and thalamus, and projecting to the pallidonigral system (Hammond et al., 2007). The same source also refers to multiple research claiming high beta synchronicity as a pathological sign in PD patients. Note that this was also presented in Chapter 2.5.1, (Hovden, 2017) and in (Hovden and Gulbrandsen, 2017a).

As observed in earlier analysis, high beta synchronicity decays with age of the culture. In this analysis, however, coherence in envelope frequencies in the beta band $13 - 30$ Hz was not analyzed. Rather coherence for frequencies up to approx. 6 Hz was analyzed. This includes the theta band. Perhaps there is a link between less coherence in theta band (that was observed here) and the decrease in power in beta band observed earlier. Thus, that the young culture was widely aiming for development at the same time that it was pathogenic as in PD, then later differentiated into more specialized dopaminergic neurons with beta-band coherence/synchronized oscillations.

Consequently, it is unlikely that synchronized bursts would occur in the adult culture. No data of old culture showed synchronized bursts. However, this is clearly not enough evidence to claim that synchronized bursts is a sign of young culture. It might be more a property of a developing dopaminergic neuronal cell culture.

Chapter 8

Conclusion and future work

8.1 Conclusion

This study has investigated spatiotemporal variation in detected action potentials across frequency components of multiple combinations of electrode signals. The approach was multivariate analysis on collections of spike count histograms from each of the 60 electrode signals on 13 MEA recordings of a developing dopaminergic neuronal culture from the time span 2017-03-20 to 2018-01-22. Note that there was an approximately 5 month gap where data from the adult culture was not available.

The synchronized bursts (around 3 Hz) with pauses of several seconds (around 10 seconds) clearly visible in raw data of the young culture develop into gradually shorter bursts with shorter pauses as the age of the culture increases. The end result is complete de-synchronization where no oscillatory activity can be detected from raw data without analyzing in frequency domain. During the same age increase, the overall power in the culture has been shown in earlier analysis to increase.

When the preprocessing is working correctly and number of action potentials are counted in the 6 second sliding 80 % overlapping windows, results from Chapter 6 provide evidence of a culture were certain frequency components start to show distinct behavior as the age of the culture increases. In some microelectrode array experiments of the adult culture, oscillatory patterns in certain high-frequency ranges (f. example $2540 - 2550$ Hz, Figure 6.8 **(a)**) are observable on most electrodes. However, the distinct frequency ranges vary with experiment.

Comparisons of MVA models of sets of two experiments have two main findings:

- If the two experiments are from the culture with different age, power spectral density variation over frequency on a small set of electrodes are the main difference between the experiments. Those electrodes contain the significant energy differences between the experiments.

  Additionally, for each of these electrodes, lower frequencies from the range $300 - 3000$ Hz are more influential than higher frequencies in this range in activity that are distinct in separating the experiments (energy difference).

- The electrode activity patterns to distinguish between two experiments with almost equal age are contained in almost all electrodes.

Additionally, for all electrodes, higher frequencies from the range $300 - 3000$ Hz are more influential than lower frequencies in this range in activity that are distinct in separating the experiments.

8.2 Future work

An immediate suggestion is to analyze more experiments with the methods described in the thesis. The experiment data could also be of better quality, f. ex. in terms of known or constant signal-to-noise-ratio or more meta data available.

Furthermore, it is necessary to experiment with different types of stimulation as already described in the introductory Chapter 1.2. Something like intraburst-based stimulation in (Wagenaar et al., 2005) would be worth looking into, or even optogenetic neuromodulation (Yaroslavsky et al., 2002; Laxpati et al., 2014) if possible/relevant. The real-time preprocessing likely is a step in the correct direction into applying multivariate analysis that can be used to determine feedback stimulation.

The following sections present some suggestions for improvement of the preprocessing and multivariate analysis in order to get closer to the NTNU Cyborg goal of making a closed-loop controlled bio-mechanic robot. Some alternative methods for spectrogram-like decomposing of signals, as well as some alternative neural activity analysis are then discussed and finalize the chapter.

8.2.1 Possible modifications of the preprocessing

Discussions in 7.1, 7.2 and 7.3 lead to some suggestions for improvement of the preprocessing methods:

- Preprocessing should count oscillations with higher envelope frequencies, maybe even up to 200 Hz, so that results can be related to existing research such as to properties related to EEG. For this specific dopaminergic culture, it would be interesting if APs in beta-band envelope frequencies ($13 - 30$ Hz) could be counted. However, this would require a much larger sampling rate if this is to be done real-time.

- The length of the sliding window could be dynamically adjusted according to some measure (for example entropy), in order to remember an optimal, varying past of neural activity. The author has no reason except for observing approx. 10 second bursts in the young culture, that 6 seconds is an optimal time window for decoding aggregated neural codes in the neural network.

- Offline preprocessing should have a better way to select noise segments for noise reduction. A possible workaround to represent common noise could be concatenating noise segments from multiple electrodes and use this as the noise segment in the offline preprocessing. For the real-time preprocessing, an automatic program to construct the noise thresholds would be a nice feature.

- Interspike-interval (ISI) histograms can be created from each frequency bin as a way to possibly analyze even finer grained spatiotemporal patterns. If the ISI histogram contains

100 frequencies of interspike-intervals, it would multiply the number of variables in the existing MVA by 100. This was not discussed in the above mentioned sections.

8.2.2 Improvement of the MVA and complementing analysis

The existing MVA could be complemented with additional analysis such as:

- Additional variable contribution measures that work on batch data can be just what is needed in the real-time MVA analysis (Westerhuis et al., 2000). Could work on a small set of selected influential electrodes from adult culture.
- Additional methods for measuring model stability such as using jack-knife estimates and stabilised PLS models in (Martens et al., 2001).
- Further measures on neuronal synchrony such as entropy measures on f. ex. scores from the bilinear models to investigate neural codes such as those in (Borst and Theunissen, 1999).
- It must be noted that the overall (first) goal of the multivariate analysis in the Cyborg project would be something like determining actionable feedback electrical stimulation to the biological neural culture (possibly with a combination of other methods/analyses and algorithms).

  Actionable stimulation is in this multivariate context regarded as stimulation that leads to clearly visible variational pattern changes visible in the latent component space in multivariate models of the preprocessed data stream.

  Determining appropriate feedback stimulation to the culture could perhaps be regarded as an optimization problem where given environmental constraints (such as blocking walls) the goal is to let the neuronal activity represent and control the robot movement and position, as well as act reasonable in accordance to the robot's behavior and sensors, as tightly as possible.
- Multivariate analysis such as PLS is perhaps more naturally fit to validate the realization of such a system described in the last point. For instance, logs of turning left or right of the controlled robot (in f. ex. the simulation environment) can be recorded in the form of a two column data matrix $\mathbf{Y}$.

  The two columns in $\mathbf{Y}$ could represent left- and right-turns turn respectively, where the column with the largest of the two values in any of the rows determine the relative turn direction/angle (based on whatever analysis the described system in the last point would use. This is in fact how the robot simulator SHODAN gives right and left turns as of today (2018-05-31)).

  The independent $\mathbf{X}$ block could contain the binned spike count histograms from real-time preprocessing method 2 in this thesis. A PLS model of this $\mathbf{X}$ and $\mathbf{Y}$ as independent and dependent blocks could then give a measure of the predictive steering ability of the system with respect to actual neural activity.

8.2.3 Other methods for spectrum-like decomposition

There exists many other methods for decomposing a signal into spectral components for time-frequency or spectrum-like analysis than the windowed Fourier transform used in the preprocessing. Some are the wavelet transform, and (the already used on the same data) empirical mode decomposition (Hovden and Gulbrandsen, 2017a). There exist also multivariate empirical mode decomposition (Rehman and Mandic, 2009) and a mixture of empirical mode decomposition and wavelet transform called variational mode decomposition (Dragomiretskiy and Zosso, 2014). Two other similar approaches are the empirical wavelet transform (Gilles, 2013) and the synchrosqueezed wavelet transform (Daubechies et al., 2011).

Lastly, there exist something called dynamic mode decomposition, which can somewhat be related to PCA and Fourier analysis (Chen et al., 2012; Schmid, 2010). In traditional control systems, model reduction involves finding a minimum realizable transfer function describing the dynamics of the system (much like the transfer function for the filter in Appendix Chapter 8.4.14) that is stable, by using methods such as balanced truncation or Hankel-norm approximation (Zhou et al., 1996). (Proctor et al., 2016) extends dynamic mode decomposition to being useful in input-output models in control theory. It is thus a newer approach in finding the optimal model dimension for input-output control of a dynamical system, which leads to finding a balanced reduced-order model of observable and controllable subspaces.

8.2.4 Some other methods for neural activity analysis

There are too many alternative approaches of analyzing neural data to be mentioned in a single section, but some are:

- Established effective connectivity analysis such as Dynamic Casual Modeling (DCM, generative model, Friston et al. (2003)), Granger Casuality (GC, Roebroeck et al. (2005)) and Transfer entropy (TE, Vicente et al. (2011); Orlandi et al. (2014); Zuo et al. (2013); Li (2015)). A part of the analysis of almost the same neural data (also from Sandvig lab, but not necessarily MEA2) in the specialization project of MSc student Helge-Andre Langåker are transfer entropy calculations as a way to find effective connectivity (directional flow of information) in the MEA experiments (Langåker, 2017). (Friston, 2011) compares functional (predictive model) with effective (generative model) connectivity analysis.
- Trying to fit the activity to existing computational models (Bassett and Bullmore, 2017; Kuznetsova et al., 2010; Cohen and Esposito, 2016).
- Using one of the many popular artificial neural network architectures for specific hard tasks, such as the multiplicative recurrent neural network used for decoding neural information in a brain-computer interface and keeping it robust to future neural variability in (Sussillo et al., 2016). Keyword: Hessian-free optimization.

Bibliography

Aaser, P. (2017). Investigating in-vitro neuron cultures as computational reservoir. `https://github.com/PeterAaser/semproj/blob/master/semesterrapport%20peter%20aaser.pdf`.

Aaser, P. (2018). Implementing and testing a closed-loop system for the ntnu cyborg hybrid computer-neuro system.

Aaser, P., Knudsen, M., Ramstad, O. H., van de Wijdeven, R., Nichele, S., Sandvig, I., Tufte, G., Bauer, U. S., Halaas, Ø.; Hendseth, S., Sandvig, A., and Valderhaug, V. D. (2017). Towards making a cyborg: A closed-loop reservoir-neuro system.

Allen, J. (1977). Short term spectral analysis, synthesis, and modification by discrete fourier transform.

Arenas, E., Denham, M., and J., C. V. (2015). How to make a midbrain dopaminergic neuron.

Aylin, A. (2009). Comparison of pls algorithms when number of objects is much larger than number of variables.

Bai, Q. and Wise, K. (2001). Single-unit neural recording with active microelectrode arrays.

Barker, R. A., Drouin-Ouellet, J., and Parmar, M. (2015). Cell-based therapies for parkinson disease —past insights and future potential.

Barter, J. W., Li, S., Lu, D., Bartholomew, R. A., Rossi, M. A., Shoemaker, C. T., Salas-Meza, D., Gaidis, E., and Yin, H. H. (2015). Beyond reward prediction errors: the role of dopamine in movement kinematics.

Bassett, D. S. and Bullmore, E. T. (2017). Small-world brain networks revisited.

Beaulieu, J. M. and Gainetdinov, R. R. (2011). The physiology, signaling, and pharmacology of dopamine receptors.

Benchenane, K., Peyrache, A., Khamassi, M., Tierney, P. L., Gioanni, Y., Battaglia, F. P., and Wiener, S. I. (2010). Coherent theta oscillations and reorganization of spike timing in the hippocampal- prefrontal network upon learning.

Beninger, R. J. (1983). The role of dopamine in locomotor activity and learning.

Bernheimer, H., Birkmayer, W., Hornykiewicz, O., Jellinger, K., and Seitelberger, F. (1973). Brain dopamine and the syndromes of parkinson and huntington.

Borst, A. and Theunissen, F. E. (1999). Information theory and neural coding.

Broersma, H. (2018). Evolution in nanomaterio: the nascence project.

Brown, P., Oliviero, A., Mazzone, P., Insola, A., Tonali, P., and Di Lazzaro, V. (2001). Dopamine dependency of oscillations between subthalamic nucleus and pallidum in parkinson's disease.

Burkow, A. V. (2016). Exploring physical reservoir computing using random boolean networks.

Buzsáki, G., Anastassiou, C. A., and Koch, C. (2012). The origin of extracellular fields and currents — eeg, ecog, lfp and spikes.

Bylesjö, M., Rantalainen, M., Cloarec, O., Nicholson, J. K., Holmes, E., and Trygg, J. (2007). Opls discriminant analysis combining the strengthsof pls da and simca classificationopls discriminant analysis: combining the strengthsof pls-da and simca classification.

CAMO (2017a). The unscrambler x method references. `http://www.camo.com/helpdocs/The_Unscrambler_Method_References.pdf`.

CAMO (2017b). The unscrambler x product page. `http://www.camo.com/rt/Products/Unscrambler/unscrambler.html`.

CAMO (2017c). The unscrambler x user manual. `http://www.camo.com/files/TheUnscramblerXv10.3-UserManual.zip`.

Chen, K. K., Tu, J. H., and Rowley, C. W. (2012). Variants of dynamic mode decomposition: Boundary condition, koopman, and fourier analyses.

Cochran, W. T., Cooley, J. W., Favin, D. L., Helms, H. D., Kaenel, R. A., Lang, W. W., Maling, G. C., Nelson, D. E., Rader, C. M., and Welch, P. D. (1967). What is the fast fourier transform?

Cohen, J. R. and Esposito, M. D. (2016). The segregation and integration of distinct brain networks and their relationship to cognition.

Daubechies, I., Lu, J., and Wu, H. T. (2011). Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool.

de Jong, S. (1993). Simpls: An alternative approach to partial least squares regression.

Deignan, J., Luján, R., Bond, C., Riegel, A., Watanabe, M., Williams, J. T., Maylie, J., and Adelmana, J. P. (2012). Sk2 and sk3 expression differentially affect firing frequency and precision in dopamine neurons.

Denker, M., Roux, S., Lindén, H., Diesmann, M., Riehle, A., and Grün, S. (2011). The local field potential reflects surplus spike synchrony.

Doi, D., Samata, B., Katsukawa, M., Kikuchi, T., Morizane, A., Ono, Y., Sekiguchi, S., Nakagawa, B., Parmar, M., and Takahashi, J. (2014). Isolation of human induced pluripotent stem cell-derived dopaminergic progenitors by cell sorting for successful transplantation.

Dragomiretskiy, K. and Zosso, D. (2014). Variational mode decomposition.

Friston, K. (2011). Functional and effective connectivity: A review.

Friston, K. J., Harrison, L., and Penny, W. (2003). Dynamic causal modelling.

Gadagkar, V., Puzerey, P. A., Chen, R., Baird-Daniel, E., Farhang, A. R., and Goldberg, J. H. (2015). Dopamine neurons encode performance error in singing birds.

Galindo-Prieto, B. (2017). Novel variable influence on projection (vip) methods in opls, o2pls, and onpls models for single- and multi-block variable selection.

Galindo-Prieto, B., Eriksson, L., and Trygg, J. (2014). Variable influence on projection (vip) for orthogonal projections to latentstructures (opls).

Galindo-Prieto, B., Eriksson, L., and Trygg, J. (2015). Variable influence on projection (vip) for opls models and its applicability in multivariate time series analysis.

Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: a tutorial.

Gershenson, C. (2004). Introduction to random boolean networks.

Gilles, J. (2013). Empirical wavelet transform.

Gireesh, E. D. and Plenz, D. (2008). Neuronal avalanches organize as nested theta- and beta/gamma-oscillations during development of cortical layer 2/3.

Gold, C., Henze, D. A., Koch, C., and Buzsáki, G. (2006). On the origin of the extracellular action potential waveform: A modeling study.

Golub, G. and Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix.

Golub, G. H. and Reinsch, C. (1970). Singular value decomposition and least squares solutions.

Hammond, C., Bergman, H., and Brown, P. (2007). Pathological synchronization in parkinson's disease: networks, models and treatments.

Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete fourier transform.

Holroyd, C. B. and Coles, M. G. H. (2002). The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity.

Hotelling, H. (1931). The generalization of student's ratio.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components.

Hovden, I. T. (2017). Principal component analysis on a time series of micro electrode-array recordings of dopaminergic neurons. `http://folk.ntnu.no/ivarthov/PCA%20on%20a%20time%20series%20of%20MEA%20recordings%20of%20Dopaminergic%20neurons%20-%20ivarthoklehovden%20-%2020171217.pdf`.

Hovden, I. T. and Gulbrandsen, H. (2017a). Long-term frequency band characteristics of a stimulated stem cell-derived dopaminergic neuronal culture. `http://folk.ntnu.no/ivarthov/Long-term%20frequency%20band%20characteristics%20of%20a%20stimulated%20stem%20cell-derived%20dopaminergic%20neuronal%20culture_Hovden_TTK7_IEEEtran_journal_paper_corrected_20171211.pdf`.

Hovden, I. T. and Gulbrandsen, H. (2017b). Microelectrode array recordings. `http://folk.ntnu.no/ivarthov/TTK19%20Presentation.pdf`.

Howe, M. W. and Dombeck, D. A. (2016). Rapid signalling in distinct dopaminergic axons during locomotion and reward.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C., and Liu, H. H. (1998). The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis.

Ikemoto, S., Yang, C., and Tan, A. (2015). Basal ganglia circuit loops, dopamine and motivation: A review and enquiry.

Im, C. and Seo, J. M. (2016). A review of electrodes for the electrical brain signal recording.

Jay, M., De Faveri, F., and McDearmid, J. R. (2015). Firing dynamics and modulatory actions of supraspinal dopaminergic neurons during zebrafish locomotor behavior.

Jenkinson, N. and Brown, P. (2011). New insights into the relationship between dopamine, beta oscillations and motor function.

Jo, J., Xiao, Y., Sun, A. X., Cukuroglu, E., Tran, H. D., Göke, J., Tan, Z. Y., Saw, T. Y., Tan, C. P., Lokman, H., Lee, Y., Kim, D., Ko, H. S., Kim, S. O., Park, J. H., Cho, N. J., Hyde, T. M., Kleinman, J. E., Shin, J. H., Weinberger, D. R., Tan, E. K., Je, H. S., and Ng, H. H. (2016). Midbrain-like organoids from human pluripotent stem cells contain functional dopaminergic and neuromelanin-producing neurons.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Science+Business Media, LLC.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey.

Kirkeby, A., Nelander, J., and Parmar, M. (2013). Generating regionalized neuronal cells from pluripotency, a step-by-step protocol.

Kirkeby, A., Nolbrant, S., Tiklova, K., Heuer, A., Kee, N., Cardoso, T., Ottosson, D. R., Lelos, M. J., Rifes, P., Dunnett, S. B., Grealish, S., Perlmann, T., and Parmar, M. (2017). Predictive markers guide differentiation to improve graft outcome in clinical translation of hesc-based therapy for parkinson's disease.

Kleinfeld, D. (2018). A practical guide to decibels. `https://neurophysics.ucsd.edu/courses/physics_120/A%20Practical%20Guide%20to%20Decibels.pdf`.

Knudsen, M. (2016). Ntnu cyborg: A study into embodying neuronal cultures through robotic systems.

Kuznetsova, A. Y., Huertas, M. A., Kuznetsov, A. S., Paladini, C. A., and Canavier, C. C. (2010). Regulation of firing frequency in a computational model of a midbrain dopaminergic neuron.

La Manno, G., Gyllborg, D., Codeluppi, S., Nishimura, K., Salto, K., Zeisel, A., Borm, L. E., Stott, S. R. W., Toledo, E. M., Villaescusa, C. J., Lönnerberg, P., Ryge, J., Barker, R. A., Arenas, E., and Linnarsson, S. (2016). Molecular diversity of midbrain development in mouse, human, and stem cells.

Langton, G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation.

Langåker, H. (2017). Investigating the network dynamics of in vitro neural cells with the use of exploratory pca and transfer entropy on mea data.

Laxpati, N. G., Mahmoudi, B., Gutekunst, C. A., Newman, J. P., Townson, R. Z., and Gross, R. E. (2014). Real-time in vivo optogenetic neuromodulation and multielectrode electrophysiologic recording with neurorighter.

Levy, R., Ashby, P., Hutchison, W. D., Lang, A. E., Lozano, A. E., and Dostrovsky, J. O. (2002). Dependence of subthalamic nucleus oscillations on movement and dopamine in parkinson's disease.

Li, J. (2015). Reconstructing neuronal connectivity from calcium imaging data using generalized transfer entropy.

Lindgren, F., Geladi, P., and Wold, S. (1993). The kernel algorithm for pls.

Lisman, J. and Buzsáki, G. (2008). A neural coding scheme formed by the combined function of gamma and theta oscillations.

Martens, H. (2016). Three algorithms for principal component analysis. Dept. Engineering Cybernetics, NTNU.

Martens, H., Høy, M., Westad, F., Folkenberg, D., and Martens, M. (2001). Analysis of designed experiments by stabilised pls regression and jack-knifing.

Martens, H. and Martens, M. (2001). *Multivariate Analysis of Quality. An Introduction*. Measurement Science and Technology, Volume 12, Number 10.

Maynard, E. M., Nordhausen, C. T., and Normann, R. A. (1997). The utah intracortical electrode array: a recording structure for potential brain-computer interfaces.

MultiChannelSystems (2017a). Mea2100-system manual. http://www.multichannelsystems.com/sites/multichannelsystems.com/files/documents/manuals/MEA2100-System_Manual.pdf.

MultiChannelSystems (2017b). Multichannel systems analyzer manual. http://www.multichannelsystems.com/sites/multichannelsystems.com/files/documents/manuals/Multi%20Channel%20Analyzer_Manual.pdf.

MultiChannelSystems (2017c). Multichannel systems datamanager manual. http://www.multichannelsystems.com/sites/multichannelsystems.com/files/documents/manuals/Multi%20Channel%20DataManager_Manual.pdf.

MultiChannelSystems (2017d). Multichannel systems experimeter manual. http://www.multichannelsystems.com/sites/multichannelsystems.com/files/documents/manuals/Multi%20Channel%20Experimenter_Manual.pdf.

NationalInstruments (2017). The fundamentals of fft-based signal analysis and measurement in labview and labwindows/cvi. http://www.ni.com/white-paper/4278/en/. From National Instruments.

Nicolas-Alonso, L. F. and Gomez-Gil, J. (2012). Brain computer interfaces, a review.

Nácher, V., Ledberg, A., Deco, G., and Romo, R. (2013). Coherent delta-band oscillations between cortical areas correlate with decision making.

Obien, M. E. J., Deligkaris, K., Bullmann, T., Bakkum, D. J., and Frey, U. (2015). Revealing neuronal function through microelectrode array recordings.

Orlandi, J., Stetter, O., Soriano, J., Geisel, T., and Battaglia, D. (2014). Transfer entropy reconstruction and labeling of neuronal connections from simulated calcium imaging.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space.

Pine, J. (1980). Recording action potentials from cultured neurons with extracellular microcircuit electrodes.

Proakis, J. G. and Manolakis, D. K. (2014). *Digital Signal Processing*, chapter 4, 7, 5, 2, pages 252, 474, 750, 751, 62. Pearson Education Limited.

Proctor, J. L., Brunton, S. L., and Kutz, N. (2016). Dynamic mode decomposition with control.

Rehman, N. and Mandic, D. P. (2009). Multivariate empirical mode decomposition.

Richardson, M. (2009). Principal component analysis.

Risvik, H. (2007). Principal component analysis (pca) nipals algorithm.

Roebroeck, A., Formisano, E., and Goebel, R. (2005). Mapping directed influence over the brain using granger causality and fmri.

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data.

Scipy (2017). scipy.signal.periodogram. `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.signal.periodogram.html`. From the Scipy Community.

Seeman, P., Bzowej, N. H., Guan, H. C., Bergeron, C., Reynolds, G. P., Bird, E. D., Riederer, P., Jellinger, K., and Tourtellotte, W. W. (1987). Human brain d and d dopamine receptors in schizophrenia, alzheimer's, parkinson's, and huntington's diseases.

Shew, W. L., Yang, H., Petermann, T., Roy, R., and Plenz, D. (2009). Neuronal avalanches imply maximum dynamic range in cortical networks at criticality.

Siemens (2017a). The autopower function... demystified! `https://community.plm.automation.siemens.com/t5/Testing-Knowledge-Base/The-Autopower-Function-Demystified/ta-p/362368`. From Siemens Product Lifecycle Management Community.

Siemens (2017b). Digital signal processing: Sampling rates, bandwidth, spectral lines, and more.... `https://community.plm.automation.siemens.com/t5/Testing-Knowledge-Base/Digital-Signal-Processing-Sampling-Rates-Bandwidth-Spectral/ta-p/402991`. From Siemens Product Lifecycle Management Community.

Siemens (2017c). What is a power spectral density (psd)? `https://community.plm.automation.siemens.com/t5/Testing-Knowledge-Base/What-is-a-Power-Spectral-Density-PSD/ta-p/360969`. From Siemens Product Lifecycle Management Community.

Smith, L. I. (2002). A tutorial on principal components analysis.

Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I., and Shenoy, K. V. (2016). Making brain–machine interfaces robust to future neural variability.

van Kerkoerle, T., Self, M. W., Dagnino, B., Gariel-Mathis, M. A., Poort, J., van der Togt, C., and Roelfsema, P. R. (2014). Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex.

Vicente, R., Lindner, M., and Pipa, G. (2011). Transfer entropy—a model-free measure of effective connectivity for the neurosciences.

Viventi, J., Kim, D., Vigel, L. and, F. E. S., Blanco, J. A., Kim, Y. S., Avrin, A. E., Tiruvadi, V. R., Hwang, S. W., Vanleer, A. C., Wulsin, D. F., Davis, K., Gelber, C. E., Palmer, L., Van der Spiegel, J., Wu, J., Xiao, J., Huang, Y., Contreras, D., Rogers, J. A., and Litt, B. (2012). Flexible, foldable, actively multiplexed, high-density electrode array for mapping brain activity in vivo.

Wagenaar, D. A., Madhavan, R., Pine, J., and Potter, S. M. (2005). Controlling bursting in cortical cultures with closed-loop multi-electrode stimulation.

Waldert, S., Pistohl, T., Braun, C., Ball, T., Aertsen, A., and Mehring, C. (2009). A review on directional information in neural signals for brain-machine interfaces.

Weinberger, M., Mahant, N., Hutchison, W. D., Lozano, A. M., Moro, E., Hodaie, M., Lang, A. E., and Dostrovsky, J. O. (2006). Beta oscillatory activity in the subthalamic nucleus and its relation to dopaminergic response in parkinson's disease.

Weisstein, E. (2017a). Eigen decomposition theorem. `http://mathworld.wolfram.com/EigenDecompositionTheorem.html`. From MathWorld–A Wolfram Web Resource.

Weisstein, E. (2017b). Singular value decomposition. `http://mathworld.wolfram.com/SingularValueDecomposition.html`. From MathWorld–A Wolfram Web Resource.

Weisstein, E. (2018). Eigen decomposition theorem. `http://mathworld.wolfram.com/NyquistFrequency.html`. From MathWorld–A Wolfram Web Resource.

Westad, F. and Marini, F. (2015). Validation of chemometric models – a tutorial.

Westerhuis, J. A., Gurden, S. P., and Smilde, A. K. (2000). Generalized contribution plots in multivariate statistical process monitoring.

Wold, S., Martens, H., and Wold, H. (1983). The multivariate calibration problem in chemistry solved by the pls method.

Wold, S., Esbensen, K. and Geladi, P. (1987). Principal components analysis.

Wu, Z. and Huang, N. E. (2009). Ensemble empirical mode decomposition: A noise assisted data analysis method.

Yaroslavsky, A. N., Schulze, P. C., Yaroslavsky, I. V., Schober, R., Ulrich, F., and Schwarzmaier, H. J. (2002). Optical properties of selected native and coagulated human brain tissues in vitro in the visible and near infrared spectral range.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S., and Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets.

Zanos, A., Zanos, T. P., Marmarelis, V. Z., Ojemann, G. A., and Fetz, E. E. (2012). Relationships between spike-free local field potentials and spike timing in human temporal cortex.

Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and optimal control*, chapter 7, 8, pages 156, 174. Prentice Hall.

Zuo, K., Zhu, J., Bellanger, J. J., and Jeannès, R. L. B. (2013). Adaptive kernels and transfer entropy for neural connectivity analysis in eeg signals.

Appendix

Supplementary background information.

8.3 Further recording procedures

8.3.1 MCS Experimeter to detect spikes from raw data

MultiChannel Experimeter can also detect spikes in an electrode recording based on thresholding. This spike detection is not used in the thesis, but is included here since it was used in the project report and lead to some results that are compared to the results in the thesis.

A spike on sample n on the time domain signal $x(n)$ is detected if

$$x(n) > 5\sqrt{Var(x(n))} \quad or \quad x(n) < -5\sqrt{Var(x(n))} \tag{8.1}$$

where the variance was calculated in MultiChannel Systems Analyzer (MultiChannelSystems, 2017b) using a selected sample segment N. In the PCA spike recordings, $N = TimeDuration[s] \times SampleRate[\frac{1}{s}] = 10[s] \times 10000[\frac{1}{s}] = 100000$.

8.3.2 MCS Experimeter to generate Interspike-interval histograms from detected spikes

The ”Spike Analyzer” module in MultiChannel Analyzer (MultiChannelSystems, 2017b) can further analyze the detected spikes from the ”Spike Detector” module (both MCS Analyzer and MCS Experimeter). The module has the ability to compute ISI histograms on each electrode from the raw data of the (7-10 min) experiments.

Interspike interval (ISI) histograms show the distribution of ISI in fixed 10 ms bins for each channel. The maximum ISI displayed is 1000 ms (MultiChannelSystems, 2017b). Hence, the histograms show spike counts over a distribution of 100 bins.

8.4 Mathematical background

8.4.1 Estimation of spike frequency from ISI bins

In the project work (Hovden, 2017), an estimation of spiking frequency was calculated for each 10 ms bin in Microsoft Excel using the formula:

$$BinFrequency_i[Hz] = \frac{1}{Avg(BinStartTimeMs_i, BinEndTimeMs_i) \times 10^-3} \tag{8.2}$$

where $BinEndTimeMs_i - BinStartTimeMs_i = 10$ ms and $i \in 100$. For example, for the last bin, $BinFrequency_{100} = \frac{1}{((990+1000)/2) \times 10^{-3}} = 1.005$ Hz. The resulting frequency band was $f \in [1.005 - 200]$ Hz with uneven frequency spacing. See Figure 8.1a.

8.4.2 ISI histogram interpolation

In the project work (Hovden, 2017), the frequency spacing needed to be even in order to group the frequencies using the average PSD powers from earlier (Figure 2.11). This was achieved by importing the modified histograms (conversion to Hz as row header) into The Unscrambler®and using the ”Interpolate...” transform with a step size of 2 on the imported data. The procedure was done on ISI histograms for each of the experiments. The results for #4 is seen in Figure 8.1.

8.4.3 The covariance matrix $\mathbf{C_X}$

Given the $M \times N$ data matrix with M objects and N variables

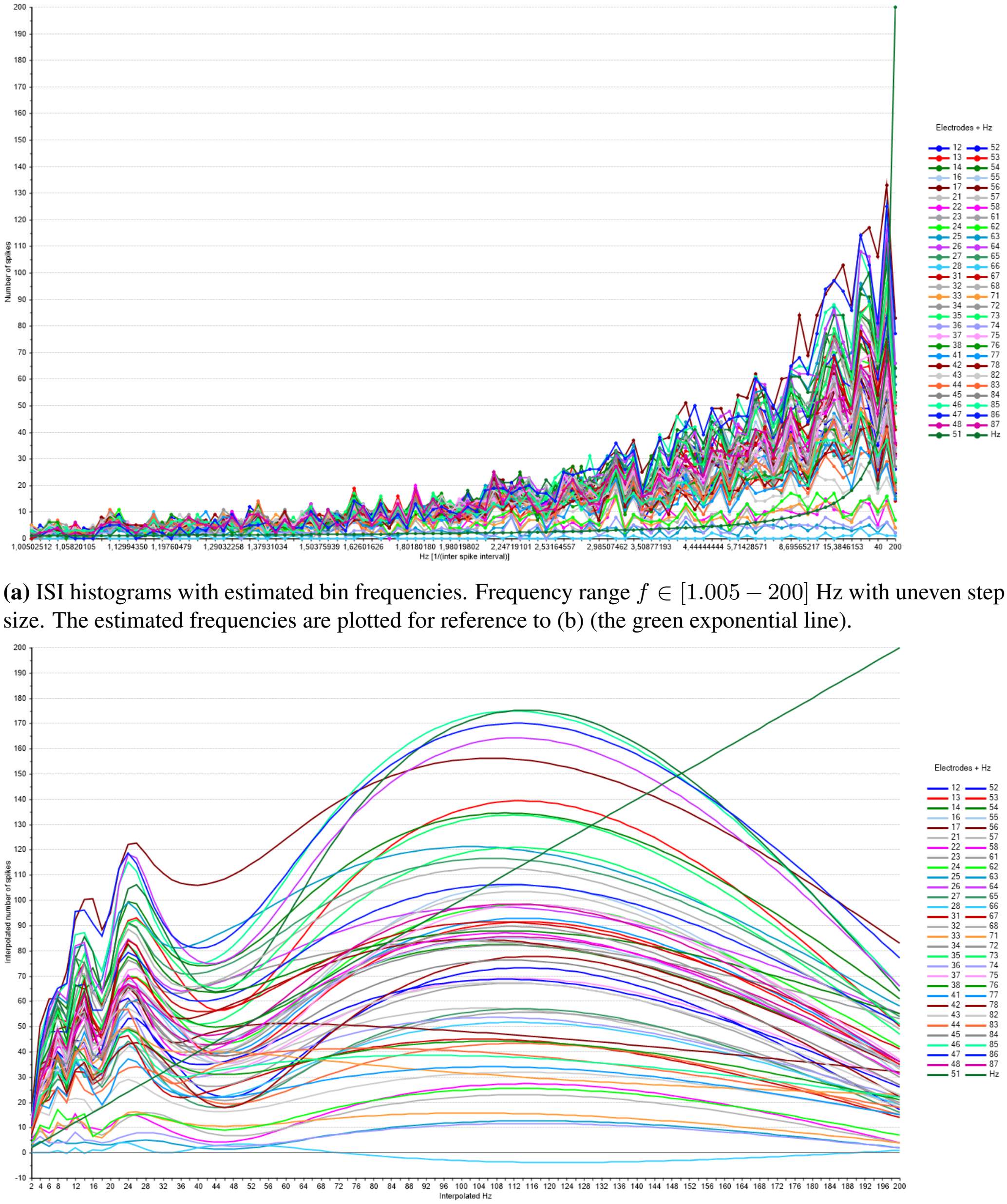
$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{bmatrix} \in \mathbb{R}^{M \times N} \tag{8.3}$$

the covariance matrix is defined to be (Richardson, 2009):

$$\mathbf{C_X} = \frac{1}{M-1}\mathbf{X}^T\mathbf{X} = \frac{1}{M-1} \begin{bmatrix} \mathbf{x}_1\mathbf{x}_1^T & \mathbf{x}_1\mathbf{x}_2^T & \dots & \mathbf{x}_1\mathbf{x}_N^T \\ \mathbf{x}_2\mathbf{x}_1^T & \mathbf{x}_2\mathbf{x}_2^T & \dots & \mathbf{x}_1\mathbf{x}_N^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_N\mathbf{x}_1^T & \mathbf{x}_N\mathbf{x}_2^T & \dots & \mathbf{x}_N\mathbf{x}_N^T \end{bmatrix} \in \mathbb{R}^{N \times N} \tag{8.4}$$

which expresses covariances between all the different variables. Hence,

$$\mathbf{C_X} = \begin{bmatrix} Cov(\mathbf{x}_1, \mathbf{x}_1) & Cov(\mathbf{x}_1, \mathbf{x}_2) & \dots & Cov(\mathbf{x}_1, \mathbf{x}_M) \\ Cov(\mathbf{x}_2, \mathbf{x}_1) & Cov(\mathbf{x}_2, \mathbf{x}_2) & \dots & Cov(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(\mathbf{x}_N, \mathbf{x}_1) & Cov(\mathbf{x}_N, \mathbf{x}_2) & \dots & Cov(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \in \mathbb{R}^{N \times N} \tag{8.5}$$



(a) ISI histograms with estimated bin frequencies. Frequency range $f \in [1.005 - 200]$ Hz with uneven step size. The estimated frequencies are plotted for reference to (b) (the green exponential line).

(b) The interpolated histograms. Frequency range $f \in [2 - 200]$ Hz with 2 as step size. By comparision with (a), most information lies in $f \in [2 - 32]$ Hz. Note that in (b) the height of the histograms are no longer spike counts, but stretched-out interpolations for integer frequencies. This was done in order to group objects using average electrode power from PSD in score and influence plot after PCA on the interpolated ISI histograms. A step size of 2 was the solution for having correct interpolation in The Unscrambler®software. The straight line of the plotted Hz verifies that the interpolation was done correctly.

Figure 8.1: The original and interpolated interspike interval histogram of #4 used in (Hovden, 2017).

8.4.4 Covariance

The covariance between the two variable sequences of length N, $\mathbf{x}_1$ and $\mathbf{x}_2$, is defined as

$$Cov(\mathbf{x}_1, \mathbf{x}_2) = E[(\mathbf{x}_1 - E[\mathbf{x}_1])(\mathbf{x}_2 - E[\mathbf{x}_2])] \tag{8.6}$$

Assuming equiprobable outcomes, the covariance formula becomes (Smith, 2002)

$$Cov(\mathbf{x}_1, \mathbf{x}_2) = E[(\mathbf{x}_1 - \bar{\mathbf{x}_1})(\mathbf{x}_2 - \bar{\mathbf{x}_2})] = \frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{x}_{1i} - \bar{\mathbf{x}_1})(\mathbf{x}_{2i} - \bar{\mathbf{x}_2}) \tag{8.7}$$

8.4.5 Variance

The variance is a measurement of variation on a single variable sequence $\mathbf{x}$, or the covariance between $\mathbf{x}$ and $\mathbf{x}$

$$Var(\mathbf{x}) = Cov(\mathbf{x}, \mathbf{x}) = \frac{1}{M-1}\sum_{i=1}^{M}(\mathbf{x}_i - \bar{\mathbf{x}})^2 \tag{8.8}$$

8.4.6 Mean

The mean $\bar{\mathbf{x}}$ of $\mathbf{x}$ is defined as (Smith, 2002)

$$\bar{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i \tag{8.9}$$

8.4.7 Eigen Decomposition Theorem

Let $\mathbf{E}$ be a matrix of eigenvectors of a given square matrix $\mathbf{A}$ and $\mathbf{D}$ be a diagonal matrix with the corresponding eigenvalues on the diagonal. Then, as long as $\mathbf{E}$ is a square matrix, $\mathbf{A}$ can be written as an eigen decomposition

$$\mathbf{A} = \mathbf{E}\mathbf{D}\mathbf{E}^{-1} \tag{8.10}$$

where $\mathbf{D}$ is a diagonal matrix. Furthermore, if $\mathbf{A}$ is symmetric, then the columns of $\mathbf{E}$ are orthogonal vectors.

If $\mathbf{E}$ is not a square matrix, then it cannot have an inverse and eigen decomposition is not possible. However, if $\mathbf{E} \in \mathbb{R}^{M \times N}$ where $\mathbf{M} > \mathbf{N}$, Singular Value Decomposition can be used instead to decompose $\mathbf{A}$.

Source: (Weisstein, 2017a)

8.4.8 Singular Value Decomposition (SVD)

Let $\mathbf{A} \in \mathbb{R}^{M \times N}$ be a real matrix with $\mathbf{M} > \mathbf{N}$. Then $\mathbf{A}$ can be written on the form

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}} \tag{8.11}$$

where $\mathbf{U} \in \mathbb{R}^{M \times M} \quad \mathbf{D} \in \mathbb{R}^{M \times N} \quad \mathbf{V} \in \mathbb{R}^{N \times N}$

and $\mathbf{U}$ and $\mathbf{V}$ have orthogonal columns so that

$$\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I} \tag{8.12}$$

and

$$\mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I} \tag{8.13}$$

where $\mathbf{I}$ is the identity matrix. The two identity matrices can have different dimensions.

Source: (Golub and Kahan, 1965; Golub and Reinsch, 1970; Weisstein, 2017b)

8.4.9 Parseval's relation for discrete periodic signals

Parseval's relation is an application of the Plancherel theorem on the Fourier transform. The discrete version yells for the Discrete Fourier Transform (DFT), used in the project, and tells the relation of total energy in a discrete signal in time and frequency domain, where the frequency domain is the DFT of the time domain signal. (Proakis and Manolakis, 2014)

$$E_N = \sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \tag{8.14}$$

where $X(k) = DFT(x(n))$.

8.4.10 Power Spectral Density (PSD)

PSD is a scaled version of the Fast Fourier Transform (FFT), which represents power in the discrete signal independently of the sampling rate. Said differently, PSD is the auto power spectrum of $x(n)$ scaled by the frequency resolution, $\frac{1}{\Delta f} = N \times \Delta t = \frac{N}{F_s}$ (this equation is called the golden equation of digital signal processing).

$$PSD = \frac{N}{Fs} S_{xx} = \frac{1}{Fs \times N} |X(k)|^2 \quad and \quad S_{xx} = \frac{1}{N^2} |X(k)|^2 \tag{8.15}$$

where Fs is the sampling rate, $X(k)$ is the FFT of $x(n)$, S_{xx} is the auto power spectrum and N is the length of the discrete signal, following from 8.14 (Siemens, 2017b,a,c; NationalInstruments, 2017; Scipy, 2017).

8.4.11 DFT, FFT, STFT, spectrogram and window functions

The Discrete Fourier Transform (DFT) is the frequency-amplitude spectrum of discrete periodic time-domain signals (Proakis and Manolakis, 2014).

The Fast Fourier Transform is an efficient software implementation of DFT (Cochran et al., 1967).

The Short-Time Fourier Transform (STFT) is a time-frequency analysis of time-domain signals. It computes multiple DFTs from a small sliding window, resulting in a frequency-amplitude-time representation (Allen, 1977). This is also more broadly called a spectrogram.

There exist many types of windows that can be used in STFT (Harris, 1978). A typical window is the ”Hamming” window. This window is used in the electrode sound STFT analysis.

8.4.12 Cross-validation

A common way to evaluate the fitness of the model during training (or statistical modeling) by dividing the training data to N (possibly random selected) segments, then training the model N times on $N-1$ segments, each time validating the model with the single segment that was not used for training at that time.

8.4.13 Nyquist frequency

In order to recover all Fourier components of a periodic waveform, it is necessary to use a sampling rate F_s at least twice the highest waveform frequency. The Nyquist frequency, also called the Nyquist limit, is the highest frequency that can be coded at a given sampling rate in order to be able to fully reconstruct the signal, i.e.,

$$f_{nyquist} = \frac{1}{2} F_s \tag{8.16}$$

Source: (Weisstein, 2018)

8.4.14 IIR and FIR filters

Infinite impulse response (IIR) and finite impulse response (FIR) filters are two different ways of achieving filtering of a discrete-time signal $x(n)$ based on a desired magnitude response (low pass, highpass, band-pass etc.) specified in equation 8.17

$$|H(z)| = \left| \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_N z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + b_M z^{-M}} \right| \tag{8.17}$$

The a a and b constants are the filter coefficients and N and M are the filter order in numerator and denominator. $H(z)$ is a transfer function in the discrete z-domain analog to the Laplace s-transform for continuous signals (f. ex. in the transfer function describing a dynamical system in control theory). z^{-1} corresponds to a discrete time shift (delay) of of 1 of an input amplitude. For a FIR filter, $a_0 = 1$ and $a_n = 0$ for $n > 0$.

With an unit impulse $\delta(n)$ as input to a filter, the response in the time-domain is modelled as $h(n)$. $H(z)$ is the z-transform of $h(n)$.

Building upon this, the casual time response for a discrete input sequence $x(n)$ is a discrete convolution with the unit impulse response $h(n)$ for that filter configuration according to equation 8.18

$$y(n) = \sum_{k=0}^{\inf} h(k)x(n-k) \tag{8.18}$$

In the FIR case, no denominator in $H(z)$ leads to a finite impulse response duration of equation 8.18, while for IIR the reponse is infinite.

FIR filters have linear phase displacement over frequencies (phase delay) while IIR have nonlinear phase delay, so FIR is often chosen in applications where having no phase distortion is important. However, IIR filters are more computationally efficient to implement and require less ”taps” than FIR filters to achieve comparable magnitude response.

Source: (Proakis and Manolakis, 2014)

Matlab’s Filter Designer Tool (fdatool) was used to design the magnitude response and filter coefficients were extracted and used in the real-time preprocessing algorithm using the scalasignal.

8.5 Software used

8.5.1 MultiChannel Systems Lab Software

- MultiChannel Experimeter 2.6.0.17256, on-line MEA recorder and analyzer. Used to record the data at Sandvigs’ lab at St. Olav University Hospital.
- MultiChannel Analyzer 2.6.0.17256, off-line analyzer. Used to re-run the recorded data to do spike detection with new standard deviation estimation for spike detection 8.1.
- MultiChannel DataManager 1.9.4.0, Used to convert the proprietary raw format to comma-separated (.csv) format.

8.5.2 The Unscrambler®version 10.5

Licensed access to a multivariate analysis software called ”The Unscrambler®10.5” (CAMO, 2017b) by CAMO Software, Oslo Science Park, Oslo, Norway. was given for use in a course in the master’s studies (TTK19 Big Data Cybernetics) as well as in project works for related students.

It is typically used in the Chemometrics industries, and provides with a fast way to analyze well-preprocessed multivariate data.

Although neuronal data is a little heavy in this program, it presents all the plots and model calibration and validation metrics in an organized way. The PCA and PLS and associated plots in the report were made using this software.

8.5.3 Microsoft Excel and Microsoft Powerpoint

Excel was used for some simple tasks such as constructing .csv header row for the real-time preprocessed data.

Powerpoint was used to compose some of the plots.

8.5.4 MATLAB

MATLAB's (The MathWorks Inc., Matick, MA, USA) Filter Designer Tool was used to design and get filter coefficients for the FIR and IIR filters tested for use in the real-time preprocessing.

MATLAB version R2017b was also used for calculating variable influence on projection (VIP) values.

8.5.5 Open-source scientific computing and visualization software

- Scala 2.11 with Scala Build Tool 1.1.0, with roughly the libraries
  - scalasignal (forked). Github link
  - Functional Streams 2 (fs2). Github link.
  - Vegas plotting library for Scala.
- Docker on linux for running the below docker image
- all-spark-notebook Docker image from docker-stacks (forked). Github link
- Python 3 with the following python libraries used for the offline preprocessing.
  - os
  - glob
  - numpy
  - pandas
  - scipy
  - matplotlib
- Jupyter Notebook, a programming web server. Used during programming to plot data.
- Audacity, an audio editor. Used for noise reduction for converted audio signals from electrodes. Version 2.1.2 (Linux) and 2.2.2 (Windows) was used.

- Baudline 1.08, a scientific spectrogram analyzer. Used to analyze the noise reduced audio signals.
- Ubuntu Linux 16.04 LTS, used with all mentioned software except The Unscrambler®, MultiChannel Software and Microsoft Excel and Powerpoint.