Machine Learning Methods for the Analysis of MEG Data



Evan James Roberts

Kellogg College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2023

Acknowledgements

Firstly, Prof. Mark Woolrich. You are knowledgeable, perceptive and above all, a kind and compassionate person. It has taken a bit longer than either of us expected for me to get to this point, but you have supported me every step of the way. Our meetings have always been equal parts science and laughter and I will miss them dearly. Your group is a testament to your passion and attitude – I hope your new students will continue appreciating the environment you foster as much as I have.

Now to my industrial supervisor, Dr Stanislaw Adaszewski. Your contributions and suggestions through the years have always been insightful and focussing. Our meetings have been a great opportunity to condense my thoughts and keep track of the course we're taking. We have been fortunate to have your involvement over the last few years, something which I hope will continue.

Dr Chetan Gohil, I don't think I've ever met anyone who works so hard, and it shows. This project would never have progressed as far as it has without you. Your generosity with your time is appreciated by everyone who works with you. Wherever your career takes you, I'm sure you'll continue to excel.

The good people of OHBA are unimaginably special. I can't believe that anywhere will ever match up to the energy and community that you all create. Lucy "Coach Jobbo" Jobbins, Grace Gillis, Lara Bolte, Rhianna Watt – my post-pandemic pals. We may all be in different places, but our brief time together has been truly wonderful. Andrew Quinn, Ryan Timms, Alex Skates, Cam Higgins and Mats van Es – you have all taught me so much over the years. To be both brilliant scientists and delightful people is a gift which you have all put to good use. And the rest of the community: Richard, Jemma, Oscar, Tom, Oliver, Verena, Dani, Sarah, Irene, Cara, Lauren, Sage, Dejan, Diego, Lara, Louise, Hadida, Neb, Layla, Ludo, Anna, Nicky, Anna-Katharina, Evan, Marco, Laurence, Nils, Karen, Kenneth. Thank you for your collective effort to maintain my baseline sanity (whatever that might be) and for making coming to work a joy.

Listing all of the folks who've enriched my life though rowing would be ambitious. To keep it brief, the members of Christ Church Boat Club, City of Oxford Rowing Club and OURCs have made my time in Oxford all the more special. I've never been much of an athlete, but through years of early mornings and shared

pain, I have truly felt like I've belonged. Special mention to my fellow trouble-makers, Pete, Marion and Marin, and to Arthur Wotton – may you never cease to be verbose.

My partner, Dr Anna James-Bott. You have travelled with me through my highs and lifted me from my lows. Seeing you thrive in your career both fills me with joy and makes me excited to take my first steps into the world beyond academia. There was no way I could have left this DPhil unfinished with your unique style of motivation to drive me on. I'm also grateful to you for bringing Pam, Mike and Alex into my life. Their kindness and hospitality are beyond anything I could ever have expected.

Finally, to my family. Mum and Dad, you've always supported me in everything I've done. It is a rare gift to advise without judgement and to guide without pressure. I will be forever grateful for every opportunity you have given me and for always being there when I needed you, even if it meant tolerating my living room "office". Mog and Pyp – having family nearby has been lovely. Thank you both for your company, and may your board game cupboard continue to run over.

This work was made possible by the generosity of the EPSRC and Roche through the SABS DTC. The computational facilities were provided by the BMRC.

Abstract

Neuroimaging data is often high-dimensional and difficult to interpret. Methods have been developed which can be applied to datasets to make them more malleable and comprehensible to researchers. This process is critical for improving our understanding of the nature of the brain. The developments made through neuroscientific research have the potential to transform the lives of many people, and making progress in the development of new methods is an important part of the discovery pipeline.

In this thesis, a novel recurrent neural network model, Dynamic Network Modes (DyNeMo), is introduced to improve on the neuroimaging analysis performance (in particular magnetoencephalography (MEG) data) compared to the current gold standard analysis method, the hidden Markov model. hidden Markov models (HMMs) depend on the Markovian property and mutual exclusivity of activating states, ignoring long-range structure within data. DyNeMo removes these limitations to allow for a potentially deeper analysis.

The validity of DyNeMo was assessed with a series of simulations to test the performance of DyNeMo in comparison to the HMM. As expected, the HMM and DyNeMo both performed well for HMM and hidden semi-Markov model (HSMM) simulated data. For a more complex simulation of a linear mixture of several modes, DyNeMo outperformed the HMM. The HMM was unable to detect co-activation of states and learn long-range dependencies of data, whereas DyNeMo could.

DyNeMo was applied to real-world MEG data from 51 subjects. DyNeMo revealed a new interpretation of the data. The modes of DyNeMo were found to have modelled the structure of a task performed by the participants which was hidden to the model during training. It also successfully captured the spectral content of the data.

In an alternative regime, DyNeMo was applied to each channel of the dataset in isolation. By selecting functions which optimise some aspect of the data, low dimensional modes can be created by combining these single-channel modes. It was found that different interpretations of the data could be produced from the same models. This approach also has the advantage of a reduced memory requirement during training, making it more accessible to researchers.

DyNeMo outperforms its predecessors. Its adoption could improve uptake of MEG in research and clinical settings and ultimately improve patient outcomes and lives in numerous neurological conditions.

Contents

List of Figures **xi**

Acronyms **xv**

1 Introduction **1**
1.1 Structure . . . 3
1.2 Collaboration . . . 4

2 Static and Dynamic Brain Network Analysis **7**
2.1 The Brain . . . 7
2.2 Form and Function . . . 8
2.3 Neuroimaging . . . 9
2.3.1 Electroencephalography . . . 10
2.3.2 Magnetic Resonance Imaging . . . 11
2.3.3 Magnetoencephalography . . . 14
2.3.4 Comparison of Neuroimaging Methods . . . 16
2.4 Physiology of MEG . . . 17
2.5 Functional Connectivity . . . 20
2.6 Brain Spaces . . . 20
2.6.1 Volumes . . . 20
2.6.2 Surfaces . . . 21
2.6.3 Dipoles . . . 21
2.7 Source Reconstruction . . . 21
2.7.1 Lead Fields . . . 22
2.7.2 Leakage Correction . . . 23
2.8 Time-delay Embedding . . . 23
2.9 Analysis Techniques . . . 24
2.9.1 Static Network Analysis . . . 24
2.9.2 Dynamic Network Analysis . . . 24
2.9.2.1 Sliding Windows . . . 25
2.9.2.2 Independent Component Analysis . . . 25
2.10 Machine Learning . . . 26

2.10.1 Hidden Markov Models . . . 27
2.10.2 Deep Learning . . . 30
2.10.2.1 Multi-Layer Perceptrons . . . 30
2.10.2.2 Convolutional Neural Networks . . . 31
2.10.2.3 Recurrent Neural Networks . . . 31
2.10.2.4 Why Deep Learning? . . . 33

3 DyNeMo 35
3.1 Mathematical Formulation . . . 36
3.2 Inference . . . 38
3.2.1 Bayesian Inference and Variational Bayes . . . 38
3.2.1.1 Amortised Inference . . . 40
3.2.1.2 The Reparameterisation Trick . . . 41
3.2.2 Free Energy . . . 41
3.3 Prediction . . . 43
3.3.1 Mitigating Edge Effects . . . 43
3.4 Software . . . 43
3.4.1 Implementation . . . 44
3.4.2 Data Structures . . . 45
3.4.3 Models . . . 45

4 Simulation Results 47
4.1 HMM Simulation . . . 47
4.2 HSMM Simulation . . . 48
4.3 Co-activating Modes Simulation . . . 51

5 Multichannel DyNeMo 57
5.1 Introduction . . . 57
5.2 The Data . . . 58
5.2.1 Acquisition . . . 58
5.2.1.1 Task . . . 59
5.2.2 Preprocessing . . . 59
5.3 Inferred Networks . . . 61
5.4 Background Activity . . . 66
5.5 HMM Comparison . . . 66
5.6 Recovery of Resting State Networks . . . 68
5.7 Mixing Coefficients . . . 73
5.8 Evoked Response . . . 74
5.8.1 Comparison with HMM . . . 78
5.9 Time-Frequency Reconstruction . . . 78
5.10 Summary . . . 80

6 Single-channel DyNeMo **83**
6.1 Introduction . . . 83
6.1.1 Potential Benefits . . . 84
6.1.1.1 Model Memory Requirement . . . 84
6.1.1.2 Post-hoc Mode Aggregation . . . 85
6.1.1.3 Observation Model Constraints . . . 85
6.1.2 Potential Drawbacks . . . 85
6.1.3 Chapter Overview . . . 86
6.2 Methods . . . 86
6.2.1 Data Preparation . . . 86
6.2.2 Parameter Choices . . . 87
6.2.3 Modes with Distinct Autocorrelation/Power Spectral Profiles 87
6.2.4 Implementation of Covariance Matrix Constraints . . . 90
6.2.5 Multi-channel DyNeMo . . . 91
6.2.6 Single DyNeMo . . . 92
6.2.6.1 DyNeMo Run Selection . . . 92
6.2.6.2 Post-hoc Analysis: Meta-modes . . . 93
6.2.6.3 Permutation Framework . . . 93
6.2.6.4 Simulated Annealing . . . 93
6.2.6.5 Alpha Correlation . . . 96
6.2.6.6 Dummy Variables . . . 97
6.2.6.7 Frequency Clustering . . . 97
6.2.6.8 Post-hoc Trial Matching . . . 98
6.3 Results . . . 99
6.3.1 Multi-Channel DyNeMo . . . 99
6.3.2 Single DyNeMo . . . 102
6.3.2.1 Run Selection . . . 102
6.3.2.2 Mode-specific Autocovariance Functions . . . 102
6.3.2.3 Permutation . . . 102
6.3.2.4 Alpha Correlation . . . 106
6.3.2.5 Dummy Variables . . . 108
6.3.2.6 Frequency Clustering . . . 112
6.3.2.6.1 Regressed PSDs . . . 114
6.3.2.6.2 Autocorrelation Functions . . . 119
6.3.2.7 Alpha RMSD . . . 119
6.3.2.8 Post-hoc Trial Matching . . . 121
6.4 Summary . . . 130

7 Conclusions **133**
7.1 Future Work . . . 135

References **137**

List of Figures

2.1 Images showing modern EEG cap and EEG trace 12
2.2 EEG aparatus 12
2.3 Structural MRI 14
2.4 MEG scanner and trace 16
2.5 Neuron potentials and magnetic fields 18
2.6 Sliding window analysis 26
2.7 Multi-layer perceptron 30
2.8 RNN topology 32

3.1 DyNeMo generative model 39

4.1 HMM simulation covariance recovery 49
4.2 HMM simulation state time course recovery 50
4.3 HSMM simulation state time course recovery 51
4.4 HSMM simulation state lifetime recovery 52
4.5 Co-activating modes simulation 55

5.1 Task structure 59
5.2 Multichannel DyNeMo PSDs 62
5.3 Multichannel DyNeMo power maps 63
5.4 Multichannel DyNeMo connectivity plots 65
5.5 Comparison of inter-mode correlation between DyNeMo and HMM 67
5.6 Multichannel HMM PSDs 68
5.7 Multichannel HMM power maps 69
5.8 Multichannel HMM power maps, demeaned 70
5.9 Multichannel HMM connectivity plots 71
5.10 Resting state DyNeMo power maps 72
5.11 Resting state DyNeMo PSDs 73
5.12 Normalised mixing coefficients 74
5.13 Barcode plot of DyNeMo alphas 76
5.14 Epoched DyNeMo alphas 77
5.15 Epoched HMM states 79
5.16 Barcode plot of HMM states 80

5.17 Time-frequency reconstruction of DyNeMo and HMM 81

6.1 Toeplitz matrices 89
6.2 Calculation of Toeplitz deviation 91
6.3 Meta-mode generation process 94
6.4 Multi-channel DyNeMo power maps for single-channel comparison . 100
6.5 Multi-channel DyNeMo epoched alphas for single-channel comparison 101
6.6 Multi-channel DyNeMo PSDs for single-channel comparison 101
6.7 Run selection methods for single-channel DyNeMo 103
6.8 Sorted Toeplitz deviations for single-channel DyNeMo 104
6.9 Exemplar correlation matrices for single-channel DyNeMo 105
6.10 Exemplar autocorrelation functions and PSDs for single-channel DyNeMo 105
6.11 Single-channel DyNeMo permutation framework 107
6.12 Metamode correlation matrices for single-channel DyNeMo with alpha correlation 108
6.13 Structured alpha correlation for single-channel DyNeMo with alpha correlation 109
6.14 Metamode alphas for single-channel DyNeMo with alpha correlation 110
6.15 Power maps for single-channel DyNeMo with alpha correlation . . . 111
6.16 Power spectral density for single-channel DyNeMo with alpha correlation 112
6.17 Alpha correlation matrices for single-channel DyNeMo with dummy variables 112
6.18 Structured alpha correlation for single-channel DyNeMo with dummy variables 113
6.19 Metamode alphas for single-channel DyNeMo with dummy variables 114
6.20 Alpha correlation power maps for single-channel DyNeMo with dummy variables 115
6.21 PSDs of metamodes for single-channel DyNeMo with dummy variables 116
6.22 Clustered parcel-mode PSDs for single-channel DyNeMo 116
6.23 Parcel-mode alphas for single-channel DyNeMo grouped by frequency cluster 117
6.24 Correlation between meta-mode alphas for single-channel DyNeMo grouped by frequency cluster 118
6.25 PSDs of meta-modes for single-channel DyNeMo grouped by frequency cluster 118
6.26 Power maps of meta-modes for single-channel DyNeMo grouped by frequency cluster 120
6.27 Clustered parcel-mode autocorrelation functions for single-channel DyNeMo 121

6.28 Parcel-mode alphas for single-channel DyNeMo grouped by frequency cluster (autocorrelation functions) 122
6.29 The correlation between the meta-mode alphas for single-channel DyNeMo grouped by frequency cluster (autocorrelation functions). The meta-modes are positively correlated with their spectral neighbours and anti-correlated with other meta-modes. 123
6.30 PSDs of meta-modes for single-channel DyNeMo grouped by frequency cluster (autocorrelation functions) 123
6.31 Power maps of meta-modes for single-channel DyNeMo grouped by frequency cluster (autocorrelation functions) 124
6.32 Alpha RMSD power maps for single-channel DyNeMo 125
6.33 Structured alpha RMSD for single-channel DyNeMo 126
6.34 Metamode alphas for single-channel DyNeMo 127
6.35 Power spectral density for single-channel DyNeMo 127
6.36 Clustered parcel-mode alpha trial responses for single-channel DyNeMo 128
6.37 Parcel-mode alphas for single-channel DyNeMo grouped by trial response 129
6.38 Metamode alphas for single-channel DyNeMo grouped by trial response 130
6.39 Power maps for single-channel DyNeMo grouped by trial response . 131
6.40 PSDs of meta-modes for single-channel DyNeMo grouped by trial response 132

Acronyms

AAL automated anatomical labelling.

API application programming interface.

BET brain extraction tool.

CNN convolutional neural network.

CNS central nervous system.

CSF cerebrospinal fluid.

DyNeMo Dynamic Network Modes.

EEG electroencephalography.

EMG electromyography.

FC functional connectivity.

FE free energy.

fMRI functional magnetic resonance imaging.

GMM Gaussian mixture model.

GPU graphics processing unit.

HMM hidden Markov model.

HSMM hidden semi-Markov model.

ICA independent component analysis.

KLD Kullback-Leibler divergence.

LCMV linearly constrained minimum variance.

LL log-likelihood.

LSTM long short-term memory.

MAGE multi-dynamic adversarial generator encoder.

MEG magnetoencephalography.

ML machine learning.

MLP multilayer perceptron.

MRI magnetic resonance imaging.

MVN multivariate normal.

NMR nuclear magnetic resonance.

OSL OHBA Software Library.

PCA principal component analysis.

PET positron emission tomography.

PSD power spectral density.

PyPI Python Package Index.

RF radio frequency.

RMSD root mean square deviation.

RNN recurrent neural network.

SGD stochastic gradient descent.

SNR signal-to-noise ratio.

SQUID superconducting quantum interference device.

STC state time course.

sTD symmetric Toeplitz deviation.

TDE time delay embedding.

VAE variational autoencoder.

VB variational Bayes.

1 Introduction

The ability to discern the way in which people's brains behave allows us to better understand how and why the brain performs different tasks. Comparing between people reveals heterogeneity in the way that the brains of different people operate. In a practical sense, being able to recognise differences in the behaviour of brains under a variety of conditions can enable us to better understand and recognise pathologies more quickly and effectively.[1]

An improved knowledge of brain activity in patients with common medical conditions such as major depressive disorder may allow a more informed choice of medication for doctors. This could in turn improve patient outcomes in an area in which a considerable amount of trial and error is currently involved.[2] In addition, research suggests that neurological conditions such as schizophrenia, bipolar disorder and epilepsy are related to abnormal connectivity in the brain[3] and as such, investigating neural connectivity should be seen as a priority. The potential for aiding development of personalised medicine is of interest to patients, healthcare providers and the pharmaceutical industry.

Neuroimaging data exists in a high-dimensional space. Data may be recorded using hundreds of channels, each measuring a different region of the brain. This presents challenges for both the analysis and understanding of these data. Large quantities of data can take a long time to process, and designing algorithms which

can perform at these scales is a non-trivial task. Humans brains are also not particularly good at comprehending data with a high dimensionality. This means that any methods used need to collect and simplify the data to a number of dimensions which can be easily understood.

A commonly used technique for the analysis of dynamic connectivity is the hidden Markov model (HMM) developed by Vidaurre et al., hereon referred to as "the HMM". It has been used in a large number of published works, both within OHBA and externally.[4–11] While the HMM has been an exceptionally successful tool in the search for new insights into neuroimaging data, there is a potential flaw. When using the HMM, we rely on the assumption that the Markovian properties of memorylessness and mutual exclusivity are reasonable constraints for modelling brain activity. These are potentially unrepresentative of the function of the brain, which performs multiple tasks concurrently[12] and exhibits behaviours which exist on the order of seconds and minutes, as well as milliseconds.[13] A potential solution to these issues is to use a model that is less tightly constrained by short-range dependencies and allows for analysis over a larger temporal scale. It is also desirable to allow for coincident activation of spatial networks by defining mixing factors which control the activation of each network individually. The suggested solution presented in this thesis is the use of recurrent neural networks (RNNs), a type of deep learning algorithm, in a model called Dynamic Network Modes (DyNeMo).

There are three main contributions that this thesis focuses on:

- In collaboration with others, the development and implementation of Dynemo,
- The validation of Dynemo using simulations and task data,
- A novel application of Dynemo for doing single channel analyses.

In the validation work, DyNeMo will be assessed using simulated data, comparing its performance to the HMM in a set of representative experiments. The aim of these tests is to ensure that DyNeMo suffers no loss of performance when training on data which lends itself to analysis using the HMM. Further simulations are used

to explore beyond these limits and demonstrate the ability of DyNeMo to learn data patterns which are not well estimated by the HMM. The use of simulations is an important step in building trust in a new model as it provides a base from which further validation and testing can be performed.

DyNeMo will then be tested using magnetoencephalography (MEG) data recorded in Nottingham as part of the UK MEG Partnership. The dataset comprises resting state data and task data, with the latter allowing for analysis within a small window with a repetitive structure. When performing a task analysis, we already understand some of the underlying structure which we may expect to be visible in the outputs of the model. This can provide a higher degree of certainty into the success of any given inference. Comparing this analysis to the equivalent using the HMM allows us to assess any changes, improvements or deteriorations in the results generated.

A novel application of Dynemo that is investigated in thesis is the single-channel regime, which initially analyses each channel separately. This approach will be tested to demonstrate a more flexible set of potential analyses which can be performed with reduced hardware requirements. With large models such as DyNeMo taking a long time to train, even on high-end hardware, the opportunity to train once on consumer-grade equipment and having the ability to modify parameters of the analysis after the fact is a potential boon. Reducing the hardware requirements of training vastly increases the number of researchers who have the potential to use DyNeMo. This in turn increases the potential for more new and exciting work to be performed using this model.

In its totality, the aim of this thesis is to develop mathematical modelling techniques to provide the clearest possible spatial, temporal and spectral descriptions of brain network dynamics.

1.1 Structure

- Chapter 2 reviews background information concerning the brain, neuroimaging and existing associated analysis techniques. It includes information re-

garding the preparation and preprocessing of data for analysis and an introduction to the concepts of machine learning.

- Chapter 3 introduces DyNeMo: the structure of the model, how it is parameterised and the inference methods used to train it. It also concerns software development, including frameworks used to prepare the data, train the model and analyse the results.
- Section 5.2 describes the data on which the model is trained. This includes the acquisition of the data, preparation steps like source reconstruction and the preprocessing performed to ready the data for analysis.
- Chapter 4 uses illustrative simulations to document the abilities of DyNeMo and compare them to the HMM when the ground truth is known.
- Chapter 5 uses DyNeMo to analyse the task portion of the Nottingham dataset. The nature of the inference is examined and differences from the HMM are explained.
- Chapter 6 uses the single-channel training regime to explore the data from a different perspective and validate this alternative methodology.
- Chapter 7 presents a summary of the research conducted and suggests future directions for continued study.

1.2 Collaboration

The work in this thesis was produced as part of a collaborative effort involving many researchers. I was responsible for much of the initial work of developing and packaging DyNeMo. My work was tightly coupled with a parallel project developed by Ryan Timms which used the same methods to improve source reconstruction. He contributed to the development of the software and the theoretical underpinnings of the project. Improvements to both DyNeMo and the analysis components of the package were developed and implemented by Chetan Gohil and Rukuang Huang.

The scale of the development required to bring these ideas to fruition would not have been possible without the collaboration of these researchers.

2

Static and Dynamic Brain Network Analysis

2.1 The Brain

Life on Earth is a perpetual competition, and the brain has proven to be a highly effective tool in navigating and succeeding in this challenge. Its evolutionary journey traces back to the chemical and light-sensing abilities of ancient single-celled organisms. Multicellular organisms required not only external sensing to detect changes in their environment, but also internal communication to coordinate their activities. As these organisms grew in size and complexity, more intricate signalling and stimuli processing was required. Electrical signals proved an effective method for internal transmission. In Cnidaria (a phylum of animals that includes jellyfish and corals), we see extremely rudimentary nervous systems called nerve nets.[14] These are networks of neurons which relay electrical signals around the body, triggering reactions such as the contraction of muscle sheets.

The next signalling evolutionary milestone, saw the emergence of the central nervous system (CNS).[15] The CNS is a hub of neurons which processes information and determines appropriate actions, rather than only passing signals. Specialisation of neurons in the central nervous system led to the development of proto-brains.[16] Even these early brains had regions of cells which were involved

in different tasks. Some cells were dedicated to controlling movement, others to detecting visual stimuli. Through evolution, the brains of animals acquired new abilities. Some of the structures they developed are still present in the brains of humans today. Mammalian brains show a massive expansion in both size and complexity when compared with reptiles and amphibians.[17] The additional capacity allows for better visual processing, detailed smell perception and more complex social relationships, like those seen in primates. In humans, the brain represents just two percent of the body by mass, but twenty percent of its energy consumption.[18] 85 billion neurons and their accompanying support cells[19–21] create a dense, complex, interconnected network which is responsible for everything from the original neural functions of sensory perception and motor control to more nebulous notions such as the conscious experience.

2.2 Form and Function

With many parts of the body, function is tied closely to form, and both form and function are readily accessible to a researcher. Apply an electric potential to a muscle, and you can make it contract. Consider the structure of the lungs under a microscope and their enormous surface area is clear. The brain, however, is more enigmatic: a gelatinous mass with many folds comprising its surface. Some features of the brain are visible upon dissection such as white and grey matter, the axons and soma of neurons, are distinct to the naked eye.[22] Some components of the brain, such as the cerebellum and hippocampus, are also distinct structures. However, simply seeing these structures does not give any indication as to their functionality.

During much of the history of neuroscience as a field, insights came mostly through unfortunate accidents and ethically questionable surgeries. Perhaps the most widely known example is Phineas Gage, a railroad worker who was impaled by a large iron rod.[23] Despite the rod going through his left cheek and exiting the top of his skull, Gage survived. His partial recovery brought changes in mood which were assumed to be due to the parts of the brain which were destroyed. By

the end of his life, many of the changes he experienced had been partly reversed,[24] which demonstrates the resilience and plasticity of the brain.

In 1861, a year after Phineas Gage's death, Paul Broca gave a presentation at the Anthropological Society of Paris.[25,26] He had been studying a subject experiencing 'aphemia' (a term of his own devising), the interruption of the ability to speak. By observing that the subject was able to understand language and voluntarily move the muscles involved in speech, he determined that the disruption was specific to spoken language. Broca conjectured that if lesions were found in a few specific locations in many subjects, the "seat of the faculty of spoken language" could be deduced.

In 1863, while continuing to express the need for further study, Broca presented his analysis of the lesions of over 25 subjects displaying aphemia.[27] He confirmed the localization of their lesions to the third left frontal convolution, a region now called the Broca area. In 1874, Carl Wernicke analysed another region of the brain and found that lesions in this location had different effects on language.[28,29] The region he described is now known as the Wernicke area and damage can result in a condition known as Wernicke's aphasia. In this case, the sufferer is unable to understand either written or spoken language, but is able to speak fluently, albeit nonsensically. The study of the localization of function to regions of the brain has remained a focus of neuroscience.

2.3 Neuroimaging

While the insights provided by lesion studies were invaluable, they were limited in scope. Many pathologies are not as obvious as aphasias and their causes are less easily identified than lesions. It is also notable that lesions are most easily studied in the dead, a category of people whose brains are not overly representative of the general population. To avoid these limitations, new techniques were required. In the effort to understand the activity of the brain without the need for dramatic intervention, neuroscientists looked to create instruments capable of measuring the activity of the brain non-invasively. Specifically, they looked to measure changes in

the electric field it produced using electroencephalography (EEG). This was not the only method for measurement. Later advances would measure magnetic fields using MEG, the changes of the dipole orientation of protons using magnetic resonance imaging (MRI), the radiation emitted during the annihilation of positrons using positron emission tomography (PET), and various other methodologies.

2.3.1 Electroencephalography

Throughout the late 19th and 20th centuries, a variety of techniques were devised which allowed scientists to measure changes in the brain. These techniques exploit many differing phenomena and each has its own strengths and weaknesses. One of the first developed was EEG. The first steps towards EEG were taken in 1875 in seminal experiments by Caton.[30] He used a galvanometer to measure electrical currents in the brains of rabbits and monkeys. He found that currents were most noticeable between the external surface of the grey matter (the cortex) and the deeper matter beneath it. Through knowledge of localised neural function, Caton was able to see that the currents he measured were affected by the activity of the animal being studied.

In the early 20th century, Hans Berger went in search of a more practical method of measuring electrical activity in the brain.[31] He expected that bioelectrical phenomena would be detectable in the central nervous system and looked for a way to detect them. In 1910, while experimenting on dogs, Berger was concerned that the measurements he found were from peripheral stimuli, and he gave up on the experiments for some time. Almost two decades later, he returned to his experiments using a more sensitive instrument. Inserting electrodes into the brain of a dog, he saw large fluctuations in the readings from the device. To ensure that no external stimuli were affecting the readings, the dog was anaesthetised. He also drained it of blood to ensure that the readings were not due to the electrical activity of the heart or movements from the rest of the body. It is worth noting that most people would find this procedure distinctly uncomfortable and its diagnostic value

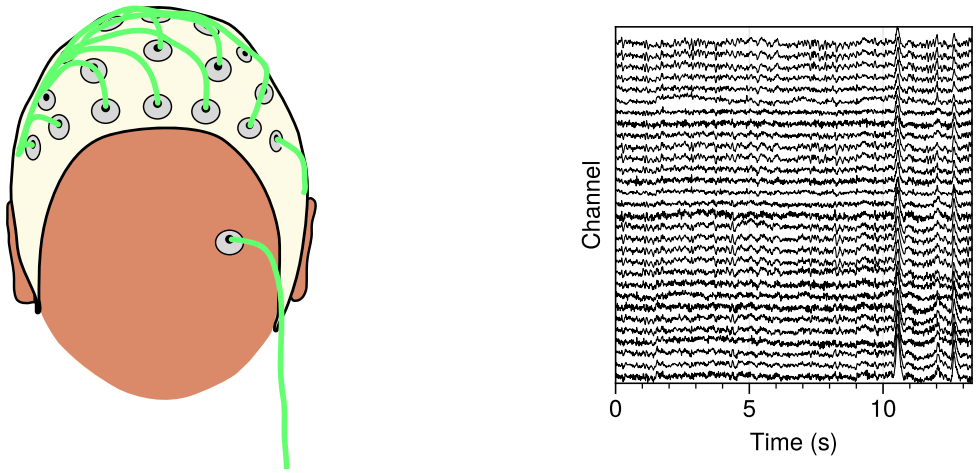
questionable. Berger then went on to test this method in human brains, using people who had undergone trepanation for unrelated reasons as subjects. He was able to confirm from this study that he could also detect currents from human brains with his method. Even without anaesthetics and blood letting, due to the invasiveness of the electrodes his method would not have been very clinically practical. Further, its diagnostic value had not yet been proven. Berger set to work developing a method to record the electrical activity of the brain from the outside of the skull.

The setup devised was reasonably simple. A rubber bathing cap was placed on the head of the subject. The bathing cap contained pieces of lead foil which acted as flexible electrodes to maintain good contact with the scalp. The electrodes were covered with a saline soaked flannel to ensure good electrical contact was maintained. Using this technique on one of the trepanned patients, Berger was able to see a consistent pattern of electrical activity. The pattern of elasticated cap, electrodes and conductive solution is recognizable in modern EEG setups as shown in Fig. 2.1a. Development of this technique has continued since Berger's initial work, and it is now a common clinical tool.

2.3.2 Magnetic Resonance Imaging

Another non-invasive technique for neuroimaging is MRI. The field is built upon the work of Isidor Rabi, who devised a new method of measuring the magnetic moment of nuclei. In 1938, Rabi presented a short paper describing an experiment in which a beam of molecules travels through a strong magnetic field. Another magnetic field is then applied perpendicular to the first, but much smaller in magnitude. The oscillations of the second field can scan through a range of frequencies. When the oscillation frequency approaches the frequency of procession (Lamor frequency) of the magnetic moment of the nuclei, the moments will realign, causing the beam of molecules to be deflected from its base course.

Measuring the intensity of the beam after the deflection, Rabi was able to determine the nuclear magnetic moment of the lithium chloride molecules he used in his experiment.[32]



(a) EEG cap **(b)** EEG trace

Figure 2.1: Images showing modern EEG cap and EEG trace

Figure 2.2: Figure 2.1a shows an illustration of an EEG cap. A series of electrodes feed into an elastic cap. The electrodes are electrically connected to the scalp through a conductive gel which is in contact with both the scalp and the electrode. An additional electrode on the left cheek serves to record the potential of the skin further away from the brain. This is used as a reference to remove potentials which are not created by the brain. The wires from the electrodes are connected to an EEG recording device which allows the signals to be stored for later analysis. Figure 2.1b shows an example EEG trace.

In 1946, Felix Bloch measured a voltage produced across normal matter (not a beam) when a radio frequency (RF) magnetic field was applied at right angles to the strong, constant field.[33] In the same year, and independently, Edward Purcell measured the absorption of the RF radiation in a similar experiment. This work decoupled the effect of the magnetic moment of the nucleus from the properties of the beam used by Rabi. In his paper, Bloch continues by describing the interactions of other forces upon the resonance created by the secondary field. In the base state of nuclear magnetic resonance (NMR), the nuclei of the material align their magnetic moments with the magnetic field. This produces a net magnetization of the material. When an RF field is applied, the moments of the nuclei are disturbed. They move away from the direction of the magnetic field and the net magnetization of the material is reduced. The precession of the moments is also synchronised between nuclei. In the absence of the energy from the RF field, the

moments gradually realign to the primary magnetic field. The time over which this occurs is characteristic of the nuclei being excited. Bloch calls this the longitudinal relaxation time (as it operates on the magnetization along the primary field), and labels it T_1. He also describes the transverse relaxation time, T_2, which is the time taken for the synchronisation of the precession of the moments to be lost. He goes on to state that in general $T_1 > T_2$. As these times are characteristic of the material being studied, they can be used to identify the material in a given sample.

In 1973, Paul Lauterbur described his invention of "zeugmatography", a term which is no longer in use. By producing a field gradient across a sample, Lauterbur was able to encode spatial information in the signal detected.[34] He followed this in 1974 with a paper describing the acquisition of images of more complex structures. The final image he presented was of the thoracic cavity of a live mouse. He noted that the lungs were visible, though not a lot else.[35] In his 1975 paper, Peter Mansfield developed the use of pulsed sequences to acquire images rapidly.[36] Lauterbur and Mansfield were awarded the Nobel Prize in 2003 for their work.

In 1977, Hinshaw, Bottomley and Holland employed this method to create an MRI image of a human wrist.[37] They applied to their sample two orthogonal field gradients which vary in time.

These fields create a static field along a line. By applying an additional static gradient along this line, an NMR signal is produced which can be interpreted using a Fourier transform. Both papers noted the potential for the use of spatial NMR in medical diagnosis, particularly in the investigation of tumours.

Throughout the 1970s and 1980s, full-body MRI scanners were developed and refined. With the increase in available computing power and magnet strength, the resolution of MRI images has increased dramatically. Rapid acquisition sequences also allow for time varying images to be obtained in a technique known as functional magnetic resonance imaging (fMRI). MRI and fMRI are used extensively in the clinic and for researching the structure and processes of the brain. A T_1-weighted MRI image of a human brain is shown in Fig. 2.3.

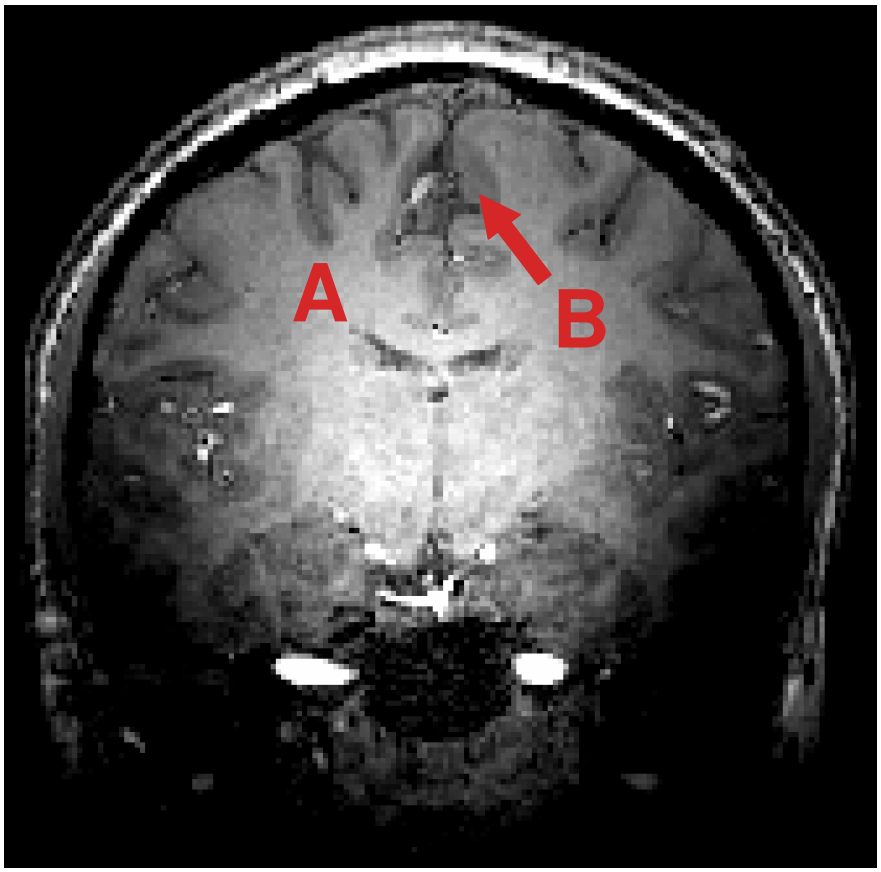


Figure 2.3: Structural MRI of a human brain. This image is T_1-weighted, meaning that it is sensitive to the longitudinal relaxation time. As such, fatty tissues appear bright, while water appears dark. The image is a 3D volume, but only a single slice is shown here. (A) marks the white matter. The myelin sheaths of the axons in the white matter are rich in fatty tissue, so they appear bright. (B) marks the grey matter. The cell bodies of the neurons in the grey matter are rich in water, so they appear dark.

2.3.3 Magnetoencephalography

MEG is the magnetic equivalent of EEG. As the brain's neurons create electrical currents (as demonstrated by Caton and Berger), Faraday's law of induction states that a magnetic field will also be generated. These magnetic fields have very small magnitudes, making them difficult to detect. In 1968, David Cohen made the first MEG measurements.[38] He conducted his experiments in a magnetically shielded room to reduce the influence of external magnetic fields. Even the magnetic field of the Earth is a substantial interferant with a strength of around 50 microtesla. The brain in contrast produces a field of only a few hundred femtotesla, around

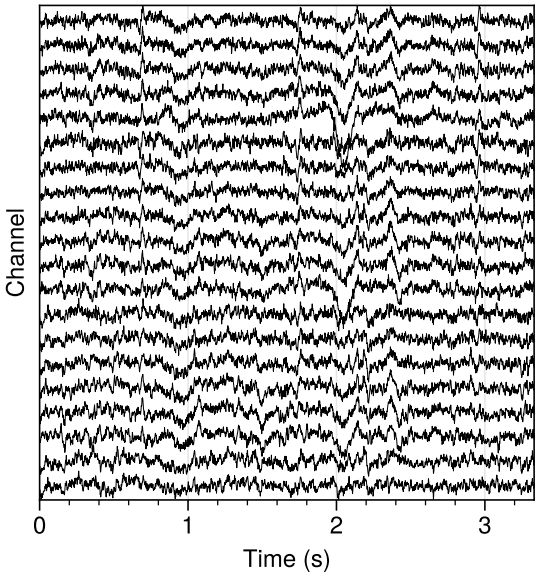
100 million times stronger.[39] In his initial experiment, Cohen used a one-million-turn copper coil as a magnetic detector. Magnetic fields in the brain induced a current in the coil, which was amplified and recorded. Even with the shielding and sensitive equipment, the noise from the environment was overwhelming for such a small signal.

A better solution to the problem of the measurement of tiny magnetic fields was a superconducting magnetometer developed by James Zimmerman in 1965.[40] Superconducting quantum interference devices (SQUIDs) are able to measure much smaller fields with reduced noise. Accompanied by improved shielding of the room, SQUIDs allowed for non-time-averaged recordings to be made. In these experiments the magnetic field was recorded by a single sensor meaning that only one area of the brain could be studied at a time. The desire to record measurements from multiple locations simultaneously led to the developments of multi-channel arrays containing many sensors. In 1993, Ahonen published a paper describing a MEG system which used 122 sensors.[41] By 1998, Neuromag had developed a system with 306 magnetometers.[42]

Unlike the apparatus of EEG, MEG requires large machines, big shielded rooms, and cryogenic cooling apparatus. The subject's head is placed within a helmet and their body position is adjusted so that their head remains stationary for the duration of the experiment. The helmet sits at the bottom of a dewar, a storage vessel containing liquid helium, which is used to keep the SQUIDs cool. The SQUIDs are at fixed positions in the helmet so keeping the subject still is important for consistent measurement. If the subject's head moves relative to the sensors, there is no longer a consistent mapping from sensor coordinates to brain coordinates. The subject's position can be adjusted to reduce movement, but a more effective method uses head casts.[43] Head casts largely eliminate the movement of the subject's head relative to the sensors by filling the intervening space with a magnetically inert material such as polyurethane. Casts are created uniquely for the combination of the subject and scanner and therefore represents an added cost in both time and



(a) MEG scanner



(b) MEG trace

Figure 2.4: Figure 2.4a shows an Elekta Neuromag MEG scanner. The subject's head is placed in the helmet at the bottom of the dewar. The dewar is filled with liquid helium to keep the SQUIDs cool. The subject's body position is adjusted so that their head remains stationary for the duration of the experiment. Image from MEG UKI (`https://meguk.ac.uk/meg-labs/university-of-oxford-meg-lab/`). Figure 2.4b shows an example MEG trace. Only 20 of 275 sensors are shown here.

money. Developments in the quality and reductions in price for 3D printing have the potential to reduce these barriers substantially.

2.3.4 Comparison of Neuroimaging Methods

Both EEG and MEG are capable of measuring the activity of the brain with high temporal resolution.[44] Changes in brain activity can occur on a timescale of milliseconds so measuring at high frequencies is important. FMRI is only capable of temporal resolutions on the order of seconds and therefore is not suitable for studying rapid changes such as the frequency content of neural oscillations.

MEG provides superior spatial resolution to EEG. The electric fields measured by EEG are smeared as they travel through the brain, skull, scalp and cerebrospinal fluid (CSF).[45] This is in part because the electric fields are attenuated by the dielectric properties of biological material and also because the fields are not conducted isotropically through the skull. In contrast, biological tissues are almost as perme-

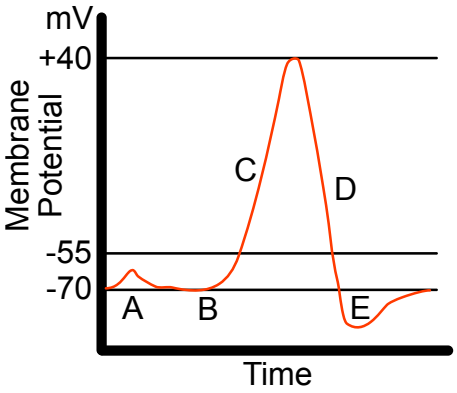
able to magnetic fields as vacuum. This means that the magnetic fields progress through the brain with minimal attenuation and the signals measured by MEG are not smeared. However, magnetic fields diminish more rapidly with distance than electric fields, so MEG is not as capable of measuring activity in deeper regions of the brain. FMRI is a volumetric technique producing high resolution even in deep regions of the brain.

This work will focus on MEG for time varying measurements and MRI for structural images.

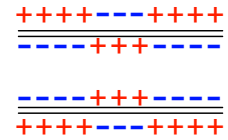
2.4 Physiology of MEG

Approximately, half the cells in the brain are neurons,[20] which receive, process and transmit electrical signals. Inbound signals are received at the dendrites, branched structures which extend from the cell body of neurons. Through ion exchange across the cell membrane, the neuron typically maintains a polarisation of around $-70\,\mathrm{mV}$, meaning the inside of the cell is less positively charged than the outside. Signals from the dendrites can reduce this polarisation. If it reduces to around $-55\,\mathrm{mV}$, a threshold potential is reached. This triggers channels to open through the membrane of the axon (a long protrusion from the cell body), allowing ions to travel down electrochemical gradients. Increases in the depolarisation of the membrane cause adjacent sections of membrane to depolarise. In this manner, the depolarisation travels down the axon as a wave. Ions only move across the membrane, not longitudinally through the axon. After the initial depolarisation, the membrane repolarises and returns to its resting potential ready to receive another input.[46]

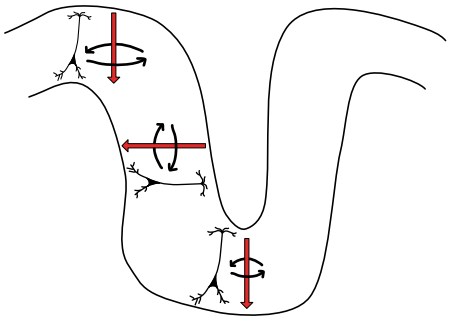
At the distal end of the axon, it joins to other neurons at a site called a synapse. Synapses are small gaps between neurons. They allow for chemical signals called neurotransmitters to mediate signal flow between cells. The neurotransmitters released from the presynaptic neuron diffuse across the synaptic cleft and bind to ligand-gated ion channels on the postsynaptic neuron's dendrites. Once bound, the ion channels open, allowing the flow of ions into the receiving cell. If sufficient depolarisation is achieved, the postsynaptic neuron will fire an action potential.



(a) Action potential of a neuron



(b) Membrane potential of a neuron



(c) Magnetic field of the cortex

Figure 2.5: In Figure 2.5a, (A) shows a failed action potential, where the threshold potential is not reached. At (B), the threshold potential of −55 mV is reached, and the action potential is triggered. (C) shows the depolarisation of the membrane to the peak potential of 40 mV. (D) shows the repolarisation of the membrane. (E) shows the hyperpolarisation of the membrane, where the potential drops below the resting potential. The membrane then returns to the resting potential. Figure 2.5b shows the membrane potential of a neuron during an action potential. The membrane potential is the difference in charge between the inside and outside of the cell. The intra-cellular side of the membrane becomes more positively charged during depolarisation, and more negatively charged during repolarisation and hyperpolarisation. Figure 2.5c shows the magnetic field of the cortex as generated by the pyramidal neurons. When the cortex is perpendicular to the scalp (the sulci), the electric current is parallel to the scalp, and the magnetic field is perpendicular to the scalp. This means the magnetic field is detectable by MEG.

Intuitively, we might expect that MEG would measure the long axonal current flows, but this is not the case.[47] The current produced by the action potential behaves as a propagating quadrupole with a magnetic field which declines in magnitude with the third power of distance. In contrast, postsynaptic currents flow along the dendrites and produce current dipoles. The complementary magnetic field falls off with the second power of distance and so is larger than that of the action potential when further away from its source.[48]

Crucially, an individual neuron does not produce a sufficient magnetic field to be measured using MEG.[48] Instead, a large number of dendrites aligned in parallel and firing synchronously are required to produce a detectable signal. Fortunately, this is found in the pyramidal neurons of the cortex.[39] The apical dendrites of pyramidal neurons are aligned perpendicularly to the cortex. This means that there are a significant number of spatially aligned postsynaptic currents. In comparison to the brief lifetimes of action potentials, postsynaptic currents can last for tens of milliseconds, which increases the tolerance for coherent action. Furthermore, while action potentials have both negative and positive relative polarisations which can cancel each other out across neuronal populations, postsynaptic currents are monophasic and therefore do not interfere deconstructively. Instead, we see the additive effect of all of the currents producing a measurable magnetic field.

There is an additional consideration to make regarding the detectability of the signal external to the scalp. That is the orientation of the fields. Only magnetic fields with a component perpendicular to the scalp are detectable.[49] Fields oriented parallel to the scalp are not measurable. The folds of the cortex mean that some parts (the walls of the sulci) are perpendicular to the scalp, while others (the crowns of the gyri) run parallel to it. As the dendrites of pyramidal neurons are perpendicular to the cortex, and the magnetic field is perpendicular to the dendrites, the magnetic field runs parallel to the surface of the cortex. This means that the walls of the sulci contribute the majority of the signal recorded by MEG.

2.5 Functional Connectivity

Functional segregation is the notion that different regions of the brain are responsible for different functionality.[50] Whilst functional segregation can be informative, a more holistic approach is required to gain true insight into the function of the brain. The study of the ways in which different brain regions cooperate to create function is called functional integration.[51] This is not to suggest that regions of the brain are not locally organised to perform specific functions, but rather that the situation is more complex than is allowed by this simplistic model. Any given perception or behaviour may be driven by a collection of diverse regions of the brain.

Functional connectivity (FC) concerns the transient interplay between spatially separated neural events.[52] FC is measured by fleeting deviations from the baseline independence of regions. These deviations suggest that spatially remote regions are acting together to produce function.[52] This neural synchrony exists at multiple spatial and temporal scales.[53]

2.6 Brain Spaces

Here, some mathematical spaces we use to talk about the brain (in particular in the fields of MRI and MEG) will be introduced and described.

2.6.1 Volumes

In general, when discussing functionality, we talk about volumetric regions which perform certain actions. The terminology of volumes is appropriate when referring to structural MRIs which comprise a grid of voxels. A voxel is a volume element - analogous to a pixel, which is a picture element.

We can characterise the regions of the brain as "parcels". Parcels are collections of adjacent voxels. We use parcels to group regions of the brain and represent them as a single object. Parcels can be determined arbitrarily, but the boundaries are normally set either by known biological structures (for example the superior occipital gyrus, or the precuneus) or data-driven approaches.

2.6.2 Surfaces

While volumes are the natural language of MRI, MEG requires discussion of surfaces. MEG sensors form a two-dimensional surface near the scalp. This means that MEG recordings are inherently two-dimensional rather than the three dimensions of MRI volumes. The magnetic field they measure is a combination of all of the fields created by the brain (as well as the heart, muscles and any unmitigated external fields). The data generated by the sensors is in "sensor-space". It is possible to do analyses in this space, but there are disadvantages. Variation in head shape means that the sensors will have different sensitivity to different parts of the brain for different subjects. This reduces the generalisability of any results. Additionally, the combined fields will be detected by multiple sensors. This means that the signals will be correlated, reducing the effective information content of any measurements.

2.6.3 Dipoles

The solution to the problems of sensor-space analysis is to move to a space which is three dimensional and mapped to the shape and size of an individual subject's brain. In MEG we are measuring the fields from effective current dipoles, and so we want to find a collection of dipoles which could be responsible for generating the signals we record. This is called "source-space". From this perspective the data is reflective of what parts of the brain are active at a given moment, rather than a jumbled combination of the fields which reach the sensors. However, this adds complexity. We only have the data from the sensors. The process of mapping from sensors to sources is called source reconstruction.

2.7 Source Reconstruction

Source reconstruction is a complex problem, but various tools have been created to attempt to solve it. The problem is that there are many combinations of sources which could produce the same magnetic field. We do not have enough information from MEG recordings to determine which one is correct.[54] This is called an

identifiability problem and is not unique to neuroimaging.[55] So, to find a field, assumptions must be made to narrow the space of potential solutions. The assumptions and the tool we will use in this thesis is the linearly constrained minimum variance (LCMV) beamformer.[56]

LCMV beamformers act as a spatial filter, attenuating activity from sources outside of a region of interest.[57] They do this by converging on a set of weights which maximise the power of the signal from the region of interest while minimising the power of the signal from outside the region of interest. The weights are calculated using the lead field and a covariance matrix of the data. A lead field represents how magnetic fields propagate through the volume of the brain and scalp. The field responds slightly differently to different tissues, but the distance of each of the sources we are trying to find to the sensors which detect them is the dominant factor.

2.7.1 Lead Fields

Calculating the lead field is also a problem with multiple solutions, depending on the level of accuracy required. As electric fields are much more susceptible than magnetic fields to the differences between tissues, this is of greater concern when analysing EEG data than MEG data.

For MEG, we take an approach called "local spheres".[58] This involves fitting a sphere to the brain for each sensor and optimising the position and radius of each sphere to best reflect the head geometry near that sensor. Head geometry is found by measuring a series of locations on the scalp using a Polhemus pen and matching these to the scalp found in the structural MRI of the subject. For each sensor-sphere pair, the lead field is calculated using the Sarvas equation,[59] which describes the magnetic field generated by a current dipole in a homogeneous sphere. With the lead field defined, the beamformer can find a valid solution which describes the activity of current dipoles producing the measured field.

2.7.2 Leakage Correction

When performing source reconstruction, a phenomenon known as source leakage can occur. This is where activity from a source outside of the region of interest is reconstructed as activity within the region of interest. Source leakage produces a smearing effect across virtual dipoles such that neighbouring regions show spuriously correlated activity, but it can be mitigated. Fortunately, when considering source leakage, the only cross-correlations introduced are of zero-lag, that is they occur at the same time-points rather than at some temporal delay. Removing these correlations is known as orthogonalisation. In this work orthogonalisation was performed using the method described by Colclough et al.[60]

2.8 Time-delay Embedding

To explicitly model the auto-correlation of a given time course, or the cross-correlation of multiple time courses, a technique called time-delay embedding can be applied to the data. This is a method of explicitly incorporating the temporal structure of data into an analysis. It is done by creating a new dataset where each sample is a vector of the original data at a given time and the n previous times. This is done by concatenating the original data with itself, shifted by n samples. We can describe the result as a matrix $\mathbf{X}$ where each column is a sample and each row is a shifted time series:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{T-n+1} & \mathbf{x}_{T-n+2} & \cdots & \mathbf{x}_T \end{bmatrix} \tag{2.1}$$

Of course $\mathbf{X}$ is only for a single time course, so the full dataset is a three dimensional tensor where the first dimension is the index of the time course, the second dimension is time, and the third dimension is the lag. By using a time-delay embedding, we can capture the temporal structure of the data and better discover the dynamics of the system.

The covariance matrix of an embedded dataset includes auto-correlations of each time course at each lag. This generates the auto-correlation function which captures spectral information about the data. It also contains cross-correlations with other time courses for both the zero-lag components and for the lagged components.

2.9 Analysis Techniques

In order to analyse the results of the various neuroimaging techniques, statistical analysis methods must be employed. There are numerous existing analysis approaches. They largely focus on the analysis of networks formed between different regions of the brain. Here, a network is a set of connections between spatial regions of the brain.

2.9.1 Static Network Analysis

The most simplistic model of functional networks is the static assumption. This assumes that the functional connectivity of the brain remains unchanged across an imaging session. The result of this style of analysis is to acquire the average connectivity during this period.[61] A region which is a member of two temporally distinct but spatially overlapping networks will cause the networks to appear connected, even if they are not functionally related.[62] As multiple networks with common nodes could appear to be one network in a static analysis, an approach which takes into account the dynamics of the brain is warranted.

2.9.2 Dynamic Network Analysis

By their nature, static analyses cannot model the dynamic changes which occur in the brain both spontaneously and as a result of tasks. A variety of dynamic techniques have been established to overcome this challenge.

2.9.2.1 Sliding Windows

One of the earliest approaches to investigating the dynamic changes in functional connectivity is the sliding window. The essence of the approach is to consider the total time series of subject data as a series of overlapping segments (windows) with equal length and step. By considering the correlation between activation in the segments, a representation of dynamic brain networks can be constructed.[63]

The choice of length and step are crucial for determining which dynamics can be observed.[64] Windows must have sufficient length to make reasonable statistical analyses, but also be short enough to prevent transient states being overwhelmed and buried. Low frequency signals are also diminished when using short windows.

Figure 2.6 shows the advantage of using sliding windows over a static analysis for a simple example. The data is sampled from a mixture of two, three-dimensional multivariate normals (MVNs) with means of zero and different covariance matrices. The mixing factor moves linearly from $[0, 1]$ to $[1, 0]$ over the course of the data. The static analysis shows a single network with a mixture of the two covariance matrices. The sliding window analysis shows that the network changes over time. Further analysis of the sliding window results can reveal the timings of the transitions between the two networks.

2.9.2.2 Independent Component Analysis

Independent component analysis (ICA) is a statistical method which separates a signal into a series of maximally independent components. Although previously used as a static analysis technique,[65] ICA has also become a tool for exploring dynamics. One way this method has been adapted is by performing ICA on a sliding window. Considering the variation of the patterns derived by ICA across these windows, insight can be gained regarding the temporal fluctuations of different functional networks.[66] This approach does not allow for internal fluctuations of the connection strengths between the nodes of the networks.[62]

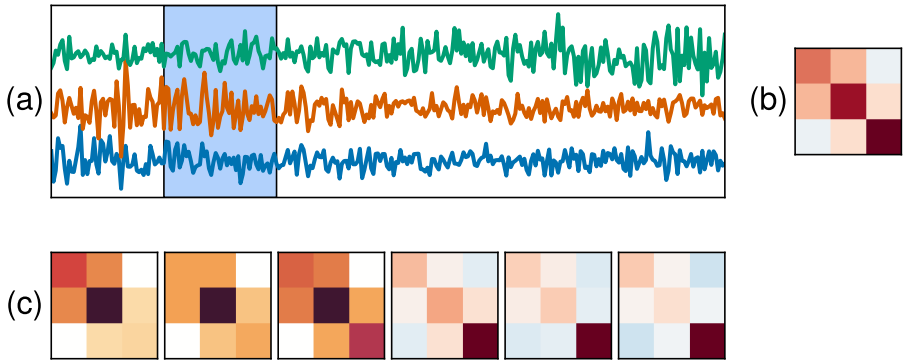


Figure 2.6: A comparison of static and sliding window analyses. The data (a) is sampled from a mixture of two, three-dimensional MVNs with means of zero and different covariance matrices. The mixing factor moves linearly from $[0, 1]$ to $[1, 0]$ over the course of the data. The static analysis shows a single network (b) with a mixture of the two covariance matrices. The sliding window analysis (c) shows that the network changes over time. The total length of the data is 300 samples, with a window length of 50. There is no overlap between windows. An example window is shown in light blue.

2.10 Machine Learning

Machine learning (ML) is a category which links many disparate algorithms. The common thread is the desire to create generalisable models, the parameters of which are found without the need for human intervention. While this may seem conceptually similar to statistical modelling, Bzdok suggests that the difference is whether the goal is inference or prediction.[67]

Statistical models are introspective of the data they collect, defining relationships between various quantities. Statisticians may craft and tweak models to best represent a hypothesised underlying structure in their data for a specific project. Meanwhile, ML models are extrospective, looking to maximise predictive power for unseen data while making minimal changes to the structure of the model. The same model structure, trained on different data, should be capable of predicting a variety of patterns. In MEG analysis, a dominant form of machine learning has been the HMM.

2.10.1 Hidden Markov Models

HMMs model data as having an underlying Markov process. A Markovian process is memoryless.[68] The only information required to determine its next state is the current state.[69] The central description of a Markovian process is the transition probability matrix. Given the current state of the system, this matrix provides the probability of the system transitioning to each of the other states.

Simulating a Markov process is merely sampling from the distribution provided by the transition probability matrix at each step. To transform a Markov process into an HMM, we have to hide the process and rely on inference from a set of observations generated by a superimposed emission process which depends on the chain.[70] This is best illustrated through an example.

Suppose you are trapped in the kitchen of a cafe with no view of the outside world. Orders come in for the three types of drinks the cafe provides - coffee, hot chocolate and lemonade. For each person who orders, the drink is determined only by the weather. Additionally, the weather (sunny, overcast or raining) changes every hour based on a set of probabilities determined by the current conditions. Let's set out the probabilities in a table called a transition probability matrix:

		Next Weather		
		Sunny	Overcast	Raining
Current Weather	Sunny	0.7	0.2	0.1
	Overcast	0.3	0.4	0.3
	Raining	0.1	0.4	0.5

This means that if it is currently raining, 40% of the time the next hour will be overcast. We also know the distribution of customer preferences for drinks, which form an emission matrix:

		Drink Preference		
		Coffee	Hot Chocolate	Lemonade
Current Weather	Sunny	0.4	0.1	0.5
	Overcast	0.5	0.4	0.1
	Raining	0.3	0.5	0.2

From the kitchen, you can't see the weather - it is our hidden Markov process. Instead, you see the drinks ordered which are randomly distributed, but with probabilities determined by the weather. The orders are our observable quantity. The question is, can you determine the most likely weather at any given time based only on the orders? The answer is yes, using the Baum-Welch algorithm.

Stepping away from our thought experiment, can we discover the underlying state of the brain (the weather) from looking at neuroimaging data (the drink orders). Taking the assumption that the brain occupies distinct, recurring states, we can add some principled assumptions to make useful inferences on data.

The first assumption is that the brain occupies a fixed number of quasi-stable states which switch according to a transition probability matrix. We then need an emission process to get neuroimaging data from the current state. A simple observation model is a set of MVN distributions. Each state is represented by a distinct MVN which is determined by a mean vector and a covariance matrix. In much the same way that you could infer the weather by considering the drinks ordered, we can infer the most likely underlying brain state by considering the data we record. The state time course (STC) produced by this inference provides us with a representation of the data in a much lower dimensional space, making it easier to interpret.

This combination of an observation model and a Markovian process produces an HMM. HMMs have shown recurrent, transient patterns of activity aligning to known resting state networks but on far shorter time scales than had been reported prior to the use of HMMs.[13] They have also revealed state hierarchies. This suggests that the measured states can be grouped into a series of metastates, within which transitions are more common than transitions which move between metastates.[5] It has been shown that HMMs are an effective tool for detecting known networks such as the dorsal and ventral attention networks.[71] HMMs can be used for analysing whole-brain, source-space, group-level data for both resting state and task based studies. They have been shown to successfully infer the timings of tasks with an unsupervised approach which is unaware a priori of the timings of the tasks.[8]

An advantage of HMMs is that the user does not need to specify a window length, unlike with sliding window techniques. The observation model and transition probability matrix are inferred directly from the data, eliminating any bias emerging from the parameter selection process. This is not to suggest that there are no parameters to select when training an HMM, only that the temporal resolution is not explicitly one of them. This leads us to a related advantage - HMMs can provide point estimates at each time step rather than an average over the window size. This increases the temporal resolution of the results and eliminates the blurring effect of a sliding window.

Although the Markovian property is a useful and powerful assumption, the constraint it imposes comes at a cost. The transition probability matrix does not encode information about the lifetimes of the state activations. This can be countered by introducing an hidden semi-Markov model (HSMM) which combines the Markovian condition for inter-state transitions with a probabilistic definition of the state lifetimes.[72]

Referring back to the cafe example, for a HSMM everything would stay the same except for the probability of the weather staying the same - the diagonal of the transition probability matrix. Instead, whenever a state is selected, a length for the visit duration would be drawn from a probability distribution. For example, each weather state could have an associated Poisson distribution to determine the number of hours for which the weather stays the same. This relaxation of the Markovian property allows for the model to learn another key aspect of the data provided to it.[73]

It is also worth noting that an HMM requires that only one state can be active at a time. This mutual exclusivity constraint is arguably not a good description of the brain, which performs multiple functions simultaneously[12] and includes various networks which overlap both spatially and temporally. Another consequence of the Markovian property is that long term dependencies cannot be modelled. As an HMM can only see one time step into the past, any processes and patterns of activity which occur over longer time scales are lost. It is known that the brain acts

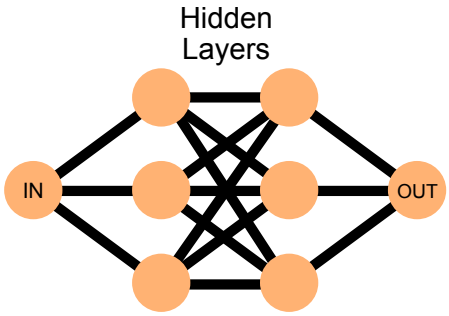


Figure 2.7: A multi-layer perceptron. Data flows from left to right; from the input layer, through the hidden layers, to the output layer. Each node in the network is connected to every node in the next layer. The weights of these connections are learned during training. The output layer is compared to the true value and the error backpropagates through the network.

on multiple time scales with considerable range.[74] To understand these patterns we need to be able to reveal these long term structures in activity.

2.10.2 Deep Learning

Deep learning refers to a set of machine learning techniques in which multiple layers of connected nodes process data.[75] It is inspired by the propagation of signals through the brain's neurons as discussed in Section 2.1. The "depth" of the network is determined by how many layers the data passes through between the input and output nodes. There are numerous deep learning techniques, for example multi-layer perceptrons, convolutional neural networks and recurrent neural networks.

2.10.2.1 Multi-Layer Perceptrons

The simplest form of neural network is the multilayer perceptron (MLP),[76] sometimes described as a 'vanilla' neural network. To form an MLP, a minimum of three layers of nodes are required: an input, an output, and at least one hidden layer. The layers are connected by non-linear activation functions such as softplus and hyperbolic tangent. This allows MLPs to find patterns in data which do not behave in simple, linear ways.

Each time data travels between nodes in the network, it is multiplied by a weight. This weight determines how much each input contributes to the output. The network outputs data which can be compared to a target value. A family of

algorithms exist for tuning the weights of the network to minimise the disparity between the expected and calculated value, called the loss. This process is called training and generally involves backpropagating gradients based on the change in loss caused by each weight.

2.10.2.2 Convolutional Neural Networks

While MLPs are powerful tools for analysing unstructured data, if spatial or temporal structure exists, convolutional neural networks (CNNs) may be more appropriate. Rather than a simple network of weights, CNNs apply a series of smaller matrices to their inputs. These smaller matrices are called kernels, and they are applied using the mathematical process of convolution.[77] During convolution, the kernel is moved across the input data and calculates a new set of values by multiplying the input values with the weights of the kernels.

Kernel convolution has been part of image processing for decades and is capable of various tasks including blurring and edge detection. The value of a CNN comes from its ability to learn the most appropriate kernels for a given task without human assistance. A common demonstration is handwritten digit recognition. The network seeks to predict which digit is in an image and compares it to the correct answer. It may develop kernels which detect edges, curves, corners, or other features of handwriting and determine combinations of these which are predictive of the digit being analysed. CNNs are widely used for image analysis, but are also sometimes used for time series analysis.[78] However, when considering time series, it may be better to use a type of network which has an understanding of the importance of sequence ordering.

2.10.2.3 Recurrent Neural Networks

RNNs aim to overcome a shortcoming of convolutional neural networks, namely their inability to learn properties related to the temporal proximity of data points in a data set. This means that when MLPs or CNNs process sequential data, much of the most important information is lost. As neuroimaging data such as that recorded from MEG is sequential and highly correlated over time, losing temporal

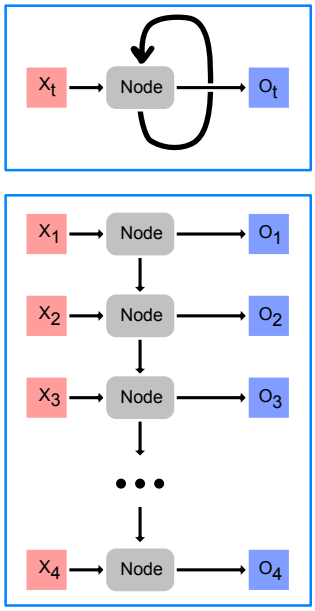


Figure 2.8: RNN topology. The upper diagram shows the cyclic connections of an RNN. The lower diagram shows the same network with the cyclic connections unrolled. The node is the same instance in all cases, but the weights are different as it moves through time, down the page.

relational structure is extremely problematic. RNNs are similar to MLPs, but the nodes are connected cyclically.[79] These cyclic connections allow data to modify the network in much the same way as in MLPs. If you were to "unravel" an RNN, you would find a neural network with multiple references to the same node. This is shown in Figure 2.8.

Modifications by one sample will affect the way the next sample is processed and so the effect of each sample propagates through the network as it learns.[79] In both MLPs and RNNs, the network is trained by considering the partial derivative of each node's weight with respect to some cost function. A cost function is a mathematical way of defining the goals of optimization for a model. A successful model will minimise its cost function as it learns. Back-propagation of gradients is a very effective way to train MLPs, but the cyclical nature of RNN nodes causes gradients to diverge rapidly. Small gradients vanish as they propagate which halts training. Large gradients explode as they propagate which makes training unstable.

Long short-term memory (LSTM) cells were introduced to combat the vanishing gradient problem.[80] LSTMs add a series of internal weights which alter the flow of

data through the network. These weights determine:

- How much the input data should impact the internal weights.
- Which parts of the internal weights should be 'forgotten'.
- How much the internal weights should impact the output data.

Like the node weights in a standard RNN, the internal weights of an LSTM cell are trainable and contribute to the model. LSTMs provide RNNs with a degree of memory without encountering the vanishing gradient problem.

2.10.2.4 Why Deep Learning?

Deep learning allows us to construct models which are highly adaptable. Frameworks developed for one aspect of data analysis easily translate to very different problems. RNNs release us from the constraints of the Markovian assumption and allow models to learn the temporal structure of the data over much larger sections of data. We also have the flexibility to define how states combine, something not offered by the HMM. Additionally, in the last two decades, graphics processing unit (GPU) performance and availability have increased considerably and software packages have been developed which allow for convenient construction and deployment of deep learning models. This means that large models and large datasets can be processed considerably faster than would have been possible until recently. Having considered the strengths of RNNs and the weaknesses of HMMs, we can formulate a set of aims for our new method to tackle.

3
DyNeMo

Neuroimaging data is inherently multivariate. Devising methods to extract meaningful information from this data is a key challenge in the field of neuroscience. Numerous techniques have been utilised to analyse neuroimaging data, including ICA, principal component analysis (PCA), sliding windows and HMMs. These methods have been applied to a wide range of neuroimaging modalities, including fMRI, MEG, EEG and PET. Each of these methods has its own strengths and weaknesses, and the choice of method is often dictated by the nature of the data and the question being asked.

DyNeMo seeks to improve upon the HMM's description of neuroimaging data. It relaxes the Markovian constraints of mutual exclusivity and memorylessness. By doing so, a new type of analysis arises which models the coactivation of multiple modes of activity within the brain.

As stated in Section 1.2, the development of DyNeMo was a collaborative effort. I was responsible for much of the initial work of developing and packaging DyNeMo and `osl-dynamics`. Working as part of a team of developers, I have continued to contribute to improving and expanding on `osl-dynamics`. This work has included improving the analysis and plotting tools provided, many of which are used in this chapter.

DyNeMo is a model for the analysis of dynamic brain networks. It is a generative model, meaning that it generates neuroimaging data modelled on the data on which it is trained. The aim of DyNeMo is to discover some underlying structure in the data which can be described using a set of modes. The modes learned by DyNeMo are linearly combined at each time point to create a complete momentary description of the data. This last point is critical, as the departure from states as seen in HMMs is what removes the constraint of mutual exclusivity.

Modes are static - unchanging over time. They represent a spatial description of brain activity. Each mode is described by a multivariate gaussian distribution, defined by a covariance matrix and a mean vector. The dynamism enters the equation through a set of mixing factors, known as alphas. These are the linear mixing factors which define the contribution of each mode at each time point.

Formally, we can describe the alphas as the latent representation and the means and covariances as the observation model. With only one time-varying mixing coefficient for each channel in the data, the alphas provide a low-dimensional representation of the data which is easy to interpret. The static latent representation, through the covariance matrix, describes both the power (via variance) in each region (diagonal elements) and the correlation between activity in different regions (off-diagonal elements). The combination of the observation model and the latent representation is sufficient to generate neuroimaging data when the model parameters are correctly inferred. This is the generative model.

3.1 Mathematical Formulation

We can formulate the generative model mathematically. We start with $\boldsymbol{\theta_t}$, a logit time-course containing free-parameters from $\mathbb{R}$. Here, logits are an unnormalised input which will be mapped to a probability distribution using the softmax function. They represent the "log odds" of each entry. $\boldsymbol{\theta_t}$ is drawn from a probability distribution defined by the values at previous time-steps:

$$p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{1:t-1}) = \mathcal{N}\left(\boldsymbol{\mu}_{\theta_t}(\boldsymbol{\theta}_{1:t-1}), \boldsymbol{\sigma}^2_{\theta_t}(\boldsymbol{\theta}_{1:t-1})\right). \tag{3.1}$$

Here, our parameters are:

- $\boldsymbol{\theta}_{1:t-1} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{t-1}\}$, the historic logits,
- $\boldsymbol{\mu}_{\theta_t}$, the mean vector, and
- $\boldsymbol{\sigma}^2_{\theta_t}$, the diagonal covariance matrix.

Future values of $\boldsymbol{\mu}_{\theta_t}$ and $\boldsymbol{\sigma}_{\theta_t}$ are predicted by a unidirectional RNN from the previous logits $\boldsymbol{\theta}_{1:t-1}$ as follows:

$$\begin{aligned} \boldsymbol{\mu}_{\theta_t}\left(\boldsymbol{\theta}_{1:t-1}\right) &= g_\mu\left(\mathrm{LSTM}\left(\boldsymbol{\theta}_{1:t-1}\right)\right), \\ \boldsymbol{\sigma}_{\theta_t}\left(\boldsymbol{\theta}_{1:t-1}\right) &= \xi\left(g_\sigma\left(\mathrm{LSTM}\left(\boldsymbol{\theta}_{1:t-1}\right)\right)\right), \end{aligned} \tag{3.2}$$

Here g_μ and g_σ are affine transformations to be learned, and ξ is the softplus function with sharpness parameter k,

$$f\left(x\right) = \frac{\ln\left(1 + e^{kx}\right)}{k}.$$

Using softplus enforces positivity on the standard deviation, $\boldsymbol{\sigma}_{\theta_t}$. We pass the predicted logits through the softmax function,

$$\zeta(\boldsymbol{z})_i = \frac{e^{\beta z_i}}{\sum_{j=1}^{K} e^{\beta z_j}} \text{ for } i = 1, \ldots, K \tag{3.3}$$

where:

- $\boldsymbol{z}$ is the vector of logits,
- β is the inverse temperature parameter, and
- K is the number of modes.

The softmax function normalises the logits to a probability distribution, meaning that all values are positive and sum to one. It can be thought of as a smooth approximation to the argmax function. The inverse temperature parameter β controls the sharpness of the softmax function. A high value of β will result in a sharp distribution, with one mode dominating the others. A low value of β will result in a flat distribution, with all modes having similar probabilities. In the limit as

$\beta \to 0$, the softmax function becomes the uniform distribution. The softmax function is used to generate the mixing coefficients, $\boldsymbol{\alpha}_t$, which are the linear mixing factors for each mode at each time point:

$$\boldsymbol{\alpha}_t = \zeta\left(\boldsymbol{\theta}_t\right). \tag{3.4}$$

The mixing coefficients are used to create the global mean vector, $\boldsymbol{\mu}_t$, and covariance matrix, $\boldsymbol{\Sigma}_t$, for each time point:

$$\begin{aligned} \boldsymbol{m}_t &= \sum_{j=1}^{J} \alpha_{jt} \boldsymbol{\mu}_j, \\ \boldsymbol{C}_t &= \sum_{j=1}^{J} \alpha_{jt} \boldsymbol{D}_j. \end{aligned} \tag{3.5}$$

with $\boldsymbol{\mu}_j$, $\boldsymbol{D}_j$ and α_{jt} being the mean vector, covariance matrix and time-dependent mixing factor for mode j, up to the number of modes J. The global mean vector and covariance matrix are used to generate the data at each time point from the multivariate normal distribution:

$$\boldsymbol{y}_t \sim \mathcal{N}\left(\boldsymbol{m}_t, \boldsymbol{C}_t\right). \tag{3.6}$$

The model is illustrated in Figure 3.1.

3.2 Inference

While the model described in Section 3.1 will produce output data, without being trained it will not be useful. Training the model involves finding the parameters which best describe the data. $\boldsymbol{\mu}_j$ and $\boldsymbol{D}_j$ are learned as point estimates, while the full posterior distribution of $\boldsymbol{\theta}_t$ is learned using variational Bayes (VB).

3.2.1 Bayesian Inference and Variational Bayes

Bayesian inference is a method for finding the posterior distribution of a set of parameters given some data and a prior distribution. More simply, in a Bayesian framework, the observer starts with a prior belief about the parameters, and then

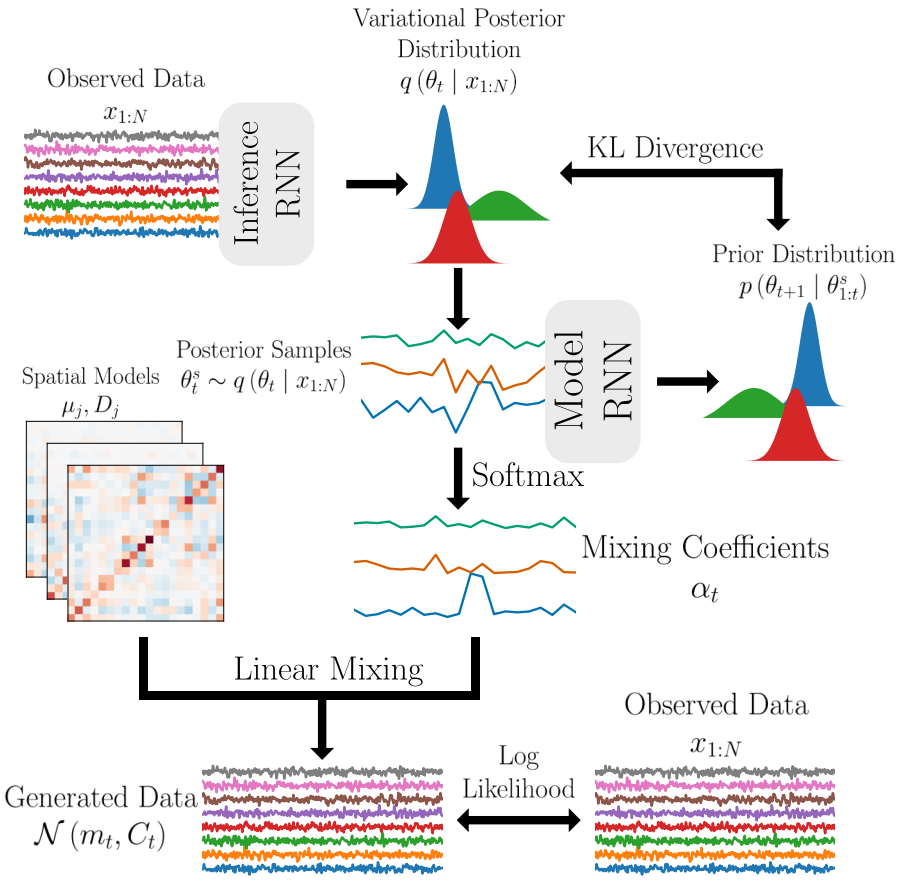


Figure 3.1: The generative model for DyNeMo. The model is a RNN with a softmax function at the output. The RNN predicts the logits, which are passed through the softmax function to generate the mixing coefficients. The mixing coefficients are used to combine the mode mean vectors and covariance matrices. This produces the generated data.

updates that belief based on the data. The central tenet of Bayesian inference is Bayes' theorem:

$$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)}, \tag{3.7}$$

where:

- Y is the set of parameters of the model,
- X is the observed data,
- $p(Y \mid X)$ is the posterior distribution of Y given X,
- $p(X \mid Y)$ is the likelihood of X given Y,

- $p(Y)$ is the prior distribution of Y, and
- $p(X)$ is the marginal likelihood of X.

VB is a method for approximating the posterior distribution by finding a distribution which is close to the true posterior distribution. The true posterior distribution is often intractable, so VB is used to find a tractable approximation. The approximation is found by minimising the Kullback-Leibler divergence (KLD) between the true posterior distribution and the approximation. The KLD is a measure of how different two distributions are. Minimising the KLD is equivalent to maximising the evidence lower bound, which is a lower bound on the log marginal likelihood of the data. The log marginal likelihood is the log probability of the data given the model and is a measure of how well the model fits the data.

Classical VB requires that a set of update equations be specified for each model. This means that the generative model must be simple enough that the update equations can be derived. It also means that the variational posterior must be from a family known as conjugate distributions. Conjugate distributions are distributions which, when multiplied by a sampling distribution, result in a posterior distribution of the same family as the prior distribution. With each model parameter being modelled by its own variational distribution, the memory requirements of VB can be prohibitive. The solution to these problems can be found from work on variational autoencoders (VAEs).

3.2.1.1 Amortised Inference

VAEs are a type of neural network which reduce the dimensions of the data by learning a latent representation of the data. The latent representation is a low-dimensional representation of the data which can be used to reconstruct the original data. A process called amortised inference is used to reduce the number of variational parameters. Amortised inference involves learning a function which maps the data to the parameters of the variational posterior. This reduces the scale of the problem from the number of data points to the number of weights

in the neural network. Amortised inference is used to learn the full posterior distribution of the logits $\boldsymbol{\theta}_t$.

3.2.1.2 The Reparameterisation Trick

DyNeMo also utilises the reparameterization trick. The reparameterization trick is a method for sampling from a distribution which allows the gradient to be back propagated through the sampling operation.[81] The parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{D}_j$ are learned as point estimates, while the full posterior distribution of $\boldsymbol{\theta}_t$ is learned. Learning these parameters is accomplished by minimising the variational free energy using stochastic gradient descent (SGD).

The inference of $\boldsymbol{\theta}_t$ relies on an approximation of the posterior distribution:

$$q\left(\boldsymbol{\theta}_t \mid \boldsymbol{x}_{1:N}\right) = \mathcal{N}\left(\boldsymbol{m}_{\theta_t}\left(\boldsymbol{x}_{1:N}\right), \boldsymbol{s}^2_{\theta_t}\left(\boldsymbol{x}_{1:N}\right)\right), \tag{3.8}$$

where $\boldsymbol{m}_{\theta_t}$ and $\boldsymbol{s}^2_{\theta_t}$ are the mean and covariance of the variational posterior distribution, and $\boldsymbol{x}_{1:N}$ is the observed data. The covariance matrix is diagonal. Much like in Eq. (3.2), the mean and covariance are predicted by an RNN from the previous logits $\boldsymbol{\theta}_{1:t-1}$. In this case, the RNN is bidirectional, meaning that it can use information from both the past and the future. This bidirectional RNN leads us to the equations for the mean and covariance of the variational posterior distribution:

$$\begin{aligned} \boldsymbol{m}_{\theta_t}(\boldsymbol{x}_{1:N}) &= f_m(\mathrm{BLSTM}(\boldsymbol{x}_{1:N})) \\ \boldsymbol{s}_{\theta_t}(\boldsymbol{x}_{1:N}) &= \xi(f_s(\mathrm{BLSTM}(\boldsymbol{x}_{1:N}))). \end{aligned} \tag{3.9}$$

Again, as in Eq. (3.2), f_m and f_s are affine transformations, and ξ is the softplus function.

3.2.2 Free Energy

As stated in Section 3.2, Bayesian inference is a strategy for updating a prior belief about the parameters of a model based on some data. We must determine what we consider to be the measure of success of the model. As in the previous discussion of deep learning, we require a loss function. For this, DyNeMo relies on the principles of free energy (FE) as described by Friston et al. in 2006.[82]

We can see from this reworking of the variational free energy that it is composed of two terms: the KLD between the approximate posterior and the true posterior, and the log-likelihood (LL). The KLD is a measure of the difference between two probability distributions. Its role in the loss function is to discourage the variational posterior from diverging too far from the prior distribution provided by the model RNN. The use of this term reduces the risk of overfitting the model to the data, which could result in the model being more sensitive to noise.

The LL is a measure of how well the model explains the data. It gives the probability of the generated data given the observed data. The LL is maximised when the model is able to explain the data well, so we take the negative of the LL to minimise it along with the KLD.

DyNeMo uses a modified version of the loss function which introduces a weighting factor λ to control the trade-off between the two terms:

$$\mathcal{L} = -\text{LL} + \lambda \cdot \text{KL}. \tag{3.10}$$

This allows DyNeMo to be more sensitive to the LL term when the model is first learning, and then to focus more on the KLD term as the model converges. At the start of training, the model can converge quickly to a local optimum by focusing on the LL term. As the model converges, the KLD term becomes more important to prevent overfitting. The changing of the weighting factor λ through training is referred to as simulated annealing.

The global mean and covariance, $\boldsymbol{\mu}_j$ and $\boldsymbol{D}_j$, are learned as point estimates. They only appear in the LL term of the loss function. There is sufficient data to learn these parameters as point estimates, and they are not used in the KLD term. While μ_j is a free parameter, $\boldsymbol{D}_j$ has a constraint that it must be positive definite. This constraint is enforced by learning the Cholesky decomposition of $\boldsymbol{D}_j$, $\boldsymbol{L}_j$, as a free parameter. The covariance matrix is then reconstructed from the Cholesky decomposition as $\boldsymbol{D}_j = \boldsymbol{L}_j \boldsymbol{L}_j^T$. L_j is a lower triangular matrix called the Cholesky factor. To increase the numerical stability of the Cholesky decomposition, the softplus function is applied to the diagonal elements of $\boldsymbol{L}_j$ and a small constant

is added. Constructing $\boldsymbol{D}_j$ in this manner ensures that it is positive definite and therefore a valid covariance matrix.

3.3 Prediction

Once trained, DyNeMo can be used to infer the underlying alpha time course of a given data set. While the outputs of all of the layers can be extracted, the primary interest is in the mode time courses and covariances. Methods are provided for both. The inference of the alphas is referred to as prediction.

3.3.1 Mitigating Edge Effects

One effect of predicting the alpha time course in batches of sequences is that edge effects can emerge. The outputs of the RNNs are different if the data fed in is the first point in a sequence compared to if it is in the middle. When concatenating sequences of alphas, this becomes apparent due to regular, chaotic changes in the alphas every sequence length. To combat this, it is possible to predict the alphas for overlapping data sequences and interpolate across the overlap. This was found to be particularly important in small-data regimes.

3.4 Software

Reproducibility is critical to good science. A functional, robust, understandable reference implementation of software is essential to enable reproducibility. Open source software contributes to this as it can be vetted by anyone who wishes to use it. By developing cleanly, rigorously and visibly, users can "trust, but verify" what is happening to their data. Clear, structured documentation allows easier onboarding of new users and more effective long term use of the methods presented.

The package developed for this work is fully open source, a departure from the previous HMM-MAR package[1] which is built in MATLAB (a proprietary, closed source software) and uses proprietary libraries. DyNeMo has been meticulously

[1] `https://github.com/OHBA-analysis/HMM-MAR/`

crafted from its inception, with a deliberate focus on adhering to established principles and best practices. A core development team maintains the codebase and determines which features to include. Issues can be raised and contributions made by anyone with a GitHub account (although additions are monitored and moderated by a team) and all bugs found are tracked as issues for full transparency. In combination with OHBA Software Library (OSL), data can be processed from collection to publication reproducibly and robustly.

3.4.1 Implementation

DyNeMo is implemented in Python in the `osl-dynamics` package. `osl-dynamics` is a general-purpose package, containing numerous models for modelling dynamical systems. It is fully open source and available on GitHub[2]. The package is available on the Python Package Index (PyPI) and can be installed with pip. Detailed documentation is available on readthedocs. In this section, we consider some of the implementation details of DyNeMo in `osl-dynamics`.

The package is designed to be extensible, so that it is straightforward to add new models to it. This means that the implementation of DyNeMo in `osl-dynamics` is sometimes more general and therefore more complex than is required for DyNeMo itself. `osl-dynamics` is implemented in Python 3.9+,[83] it uses TensorFlow 2.9+[84] and numpy 1.20+[85] for its numerical computations. `osl-dynamics` takes advantage of TensorFlow's automatic differentiation and GPU acceleration.[86] It features substantial plotting capabilities, which are implemented using matplotlib. An application programming interface (API) is defined for the package called the config-api. This allows users to define their models in YAML files, a human-friendly data serialisation language for all programming languages, which are then parsed by `osl-dynamics`. This means that model configurations are in a human-readable format, which can be easily shared with others. The config-api is also accessible from a command-line interface, which allows users to run models with minimal effort. This contributes to the reproducibility of the results.

[2] github.com/OHBA-analysis/osl-dynamics

3.4.2 Data Structures

One of the primary components of `osl-dynamics` is the Data class. This class is used to store and manipulate time series data. Its primary duty is to prepare the data for use in the model. This includes:

- Loading the data from a file or files,
- validating the data,
- filtering the frequency content of the data,
- downsampling the data,
- applying PCA to the data,
- applying time delay embedding (TDE) to the data,
- calculating the amplitude envelope of the data,
- calculating a moving average of the data,
- standardising the data and
- converting the data to a TensorFlow Dataset object.

Relevant directories, files and parameters can be specified either in code or in a YAML configuration file. In order to maintain accessibility to users with low memory systems, the data is stored on disk and operated on using numpy memory maps.

3.4.3 Models

While the focus of this thesis is on DyNeMo, `osl-dynamics` also implements a number of other models. These include an implementation of an HMM, multi-dynamic adversarial generator encoder (MAGE) and some variants of DyNeMo. Each model provides a Model class which wraps an underlying TensorFlow model. The Model class provides a consistent API for the models, which makes it easy to use them interchangeably.

4
Simulation Results

Before using DyNeMo to gain insight into the dynamics of a real MEG dataset, it was important to first validate the model. Validating the model on simulated data allowed verification that the model was able to recover the ground truth parameters of simulations. We use different simulations that correspond to the different types of scenarios we expect to see in real data, allowing us to gain insight into the potential limitations of the model.

All the simulations in this section were generated using `osl-dynamics` and are easily modified to explore other scenarios. Each one was designed to test a different aspect of the model. As the HMM is the current gold standard for analysing MEG data in this fashion, we compare the performance of DyNeMo to that of the HMM throughout.

4.1 HMM Simulation

The first simulation was designed to illustrate the ability of the model to recover the ground truth parameters of a basic HMM simulation. This was an important baseline to establish when seeking to remove some of the potential limitations of HMM inference, such as the assumption of mutually exclusive states and Markovian temporal dependencies. The simulation was defined by a transition probability

matrix (which determined the hidden action of the Markov chain) and a multivariate normal distribution for each state (an emission process). To construct the simulated data, the Markov chain was initialised with a random state and new states were drawn from the transition matrix. When the state time course was known, the observations were drawn from the corresponding multivariate normal distribution. The observations were then concatenated to form the simulated time course. The model was trained on this simulated data, and the parameters were compared to the ground truth.

HMMs trained on these simulations were unsurprisingly able to recover the ground truth parameters of this simulation almost perfectly. The model parameter space was identical to that of the simulation, and so it was expected that they would perform well. DyNeMo's generative model and the corresponding assumptions are not tuned to this simulation, and so had the potential to perform more poorly. Despite this, DyNeMo matched the parameter recovery of the HMM, as shown in Figs. 4.1 and 4.2. Most notably, even though DyNeMo does not assume mutual exclusivity, it has the flexibility to infer approximately mutually exclusive dynamics if that is how the data can be best represented.

When compared with the simulated time course, the HMM attained a dice coefficient of 0.9998, while DyNeMo reached 0.9984. We now move on to more complex simulations.

4.2 HSMM Simulation

This simulation was constructed similarly to the HMM simulation, but with the addition of a state duration distribution. The state duration distribution was drawn from a gamma distribution, which allowed for the simulation of states with a range of durations. A set of durations was drawn for each state, and the state time course was constructed by concatenating the states in the order they were drawn. The observations were then drawn from the corresponding multivariate normal distribution for each state. The model was trained on this simulated data, and the parameters were compared to the ground truth.

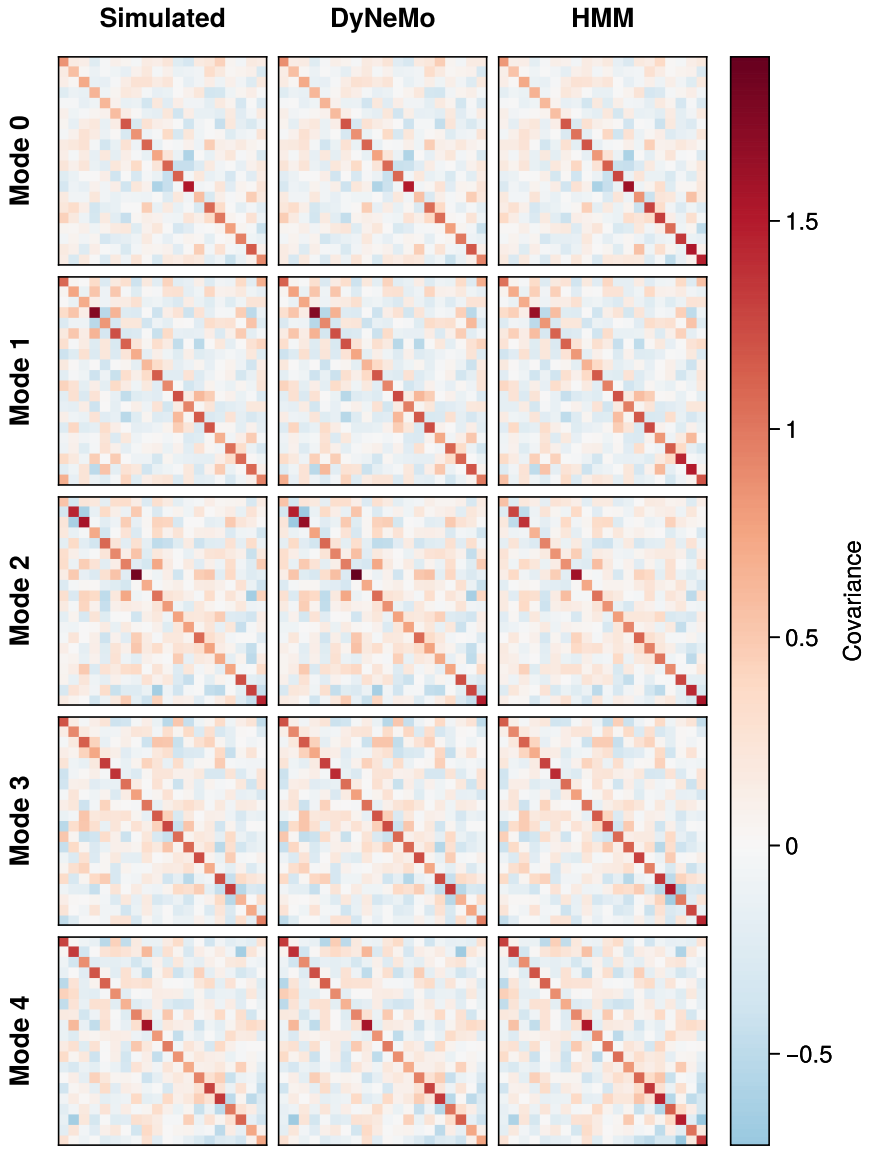


Figure 4.1: The covariance matrices found by DyNeMo and the HMM very accurately reconstruct the covariance matrices in the observation model of the simulated HMM. It is expected that the HMM and DyNeMo should perform similarly on a simple dataset such as this. The HMM is able to fully parameterize the simulated time course. DyNeMo's ability to learn these covariances correctly is a demonstration that it can recover parameters which it is not specifically engineered to learn.

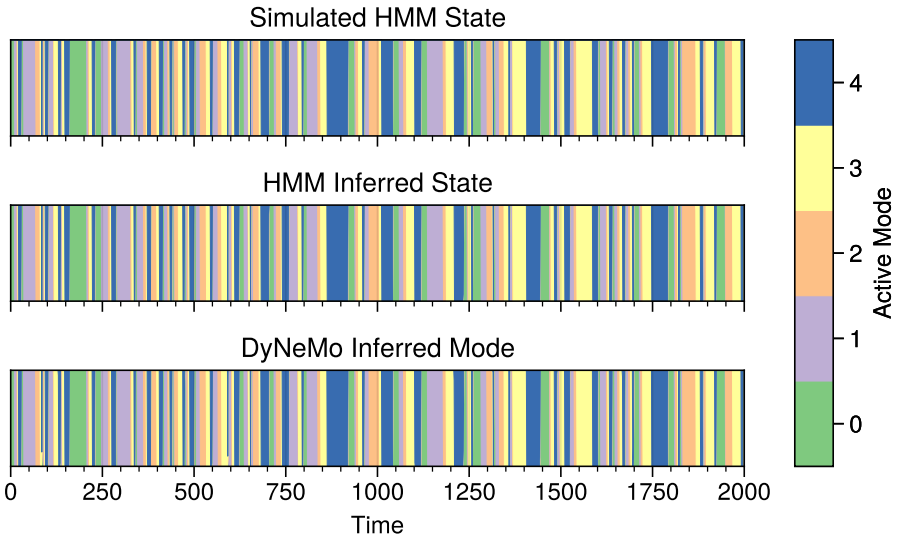


Figure 4.2: DyNeMo and the HMM correctly infer the state time course of the simulated HMM. As with the covariances, this is expected, but shows that DyNeMo is capable of learning in regimes for which the HMM is specifically designed. Time is given in samples and is arbitrary as there is no frequency content in the data. The dice coefficient of the inferred state time courses relative to the simulation is 0.9998 for the HMM and 0.9984 for DyNeMo.

As shown in figure Fig. 4.3, both the HMM and DyNeMo are able to correctly infer the ground-truth state time courses. This result might seem counter-intuitive, as the HMM is correctly inferring state time courses that have longer range temporal structure, despite only being able to model short range temporal structure. However, the state time courses inferred by the approaches are not the same as the state time courses one gets if you generate data from the inferred models.

If we instead examine the state time courses generated from the inferred models, the HMM is only capable of generating exponentially distributed state visits. Comparing this with DyNeMo, the distribution of generated state visit lengths matched the ground truth distribution well. This is shown in Fig. 4.4.

However, despite this deficiency, the HMM is nonetheless able to infer state time courses that reasonably well match the ground truth lifetimes (see Fig. 4.4). This is because in this simulation there is enough information available directly from the data to allow the state time courses to be inferred well, without the need for regularisation from the HMM's temporal modelling. It should be possible to

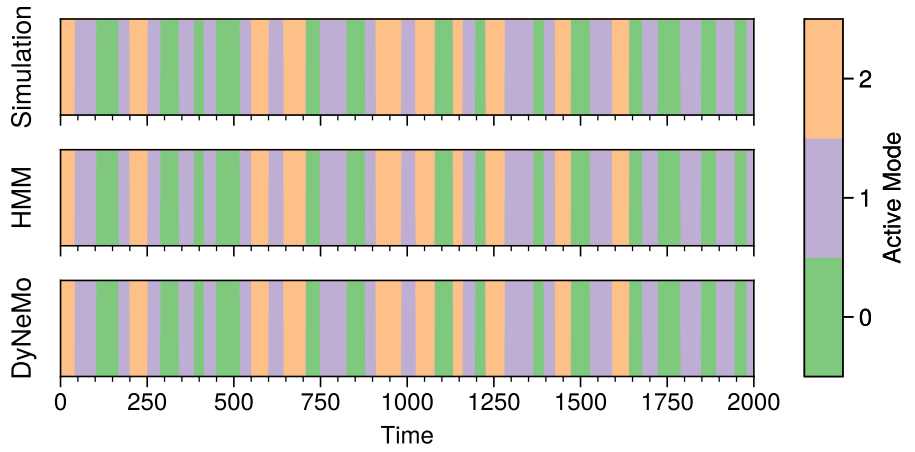


Figure 4.3: Both DyNeMo and the HMM can correctly recover the state time course of the simulated HSMM. As with the HMM simulation, this result is expected. The HMM can to infer state time courses which follow non-Markovian activations. As DyNeMo is unrestricted by the Markovian assumption, there is no difference between the HMM and HSMM simulations.

show that the HMM also infers state time courses less well than Dynemo in other scenarios that require the information coming from the data to be supplemented by regularisation from the temporal modelling. For example, when there is lower signal-to-noise ratio (SNR) or less data available.

Although this simulation was not an accurate model of MEG data, it demonstrates the ability of DyNeMo to model long-range dependencies and gives confidence that it has the potential to model the dynamics of MEG data and generate data which exhibits similar structure.

4.3 Co-activating Modes Simulation

One of the key desired features of DyNeMo is the ability to infer modes that can activate at the same time as each other; hence why prefer to refer to them as "modes" rather than "states". In this simulation we illustrate the ability of the model to recover the ground truth parameters of a linear mixture of several modes. This required us to use simulations from something other than vanilla Markov models which assume mutually exclusive states.

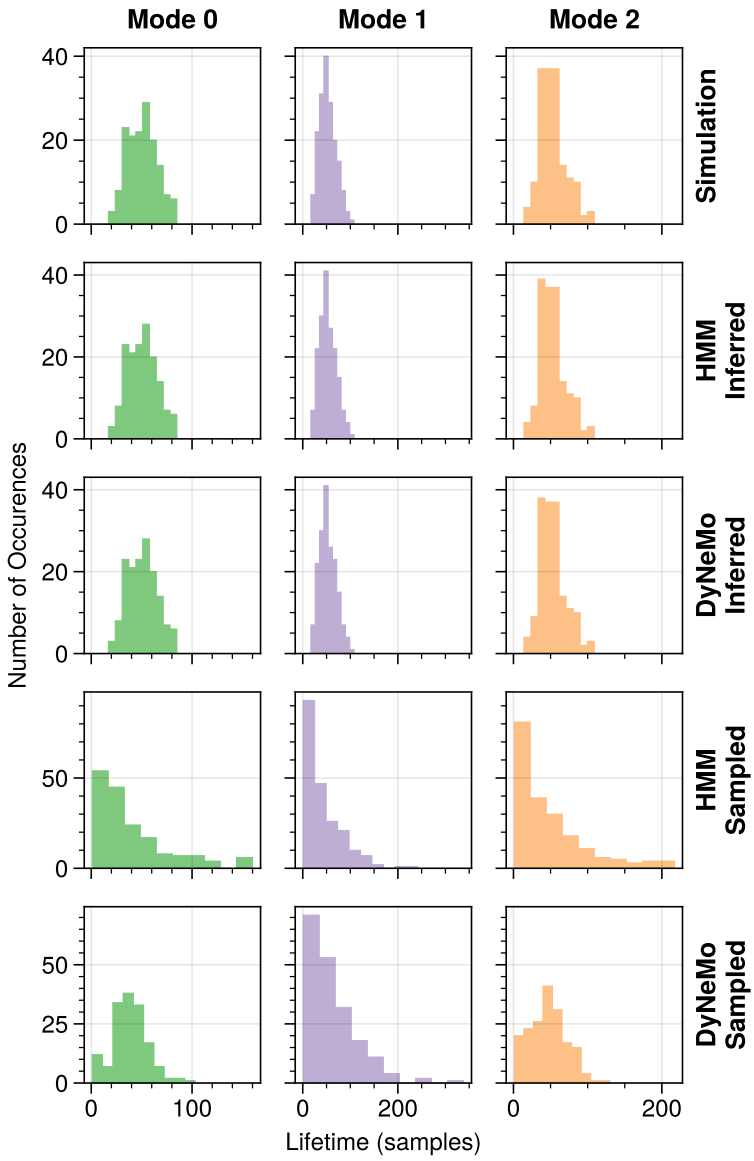
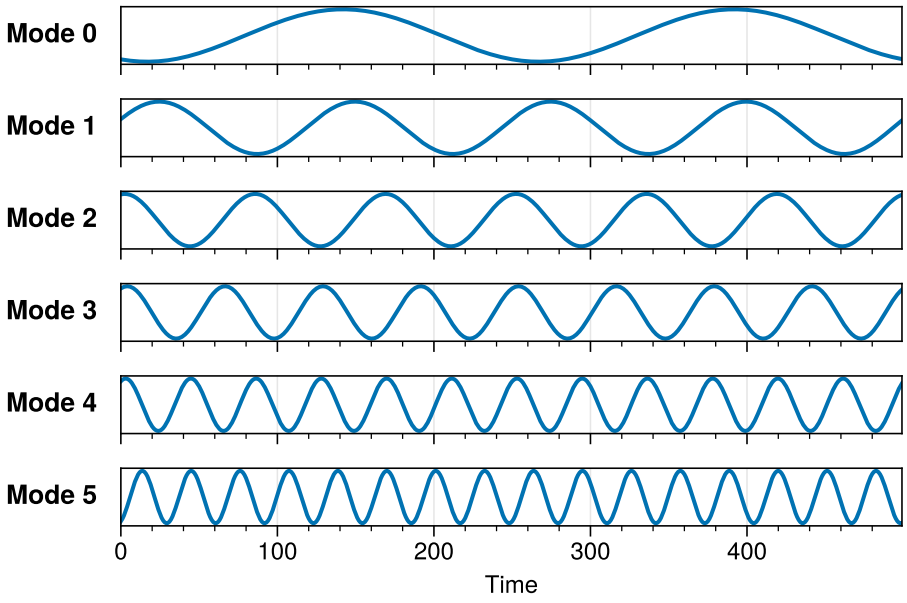


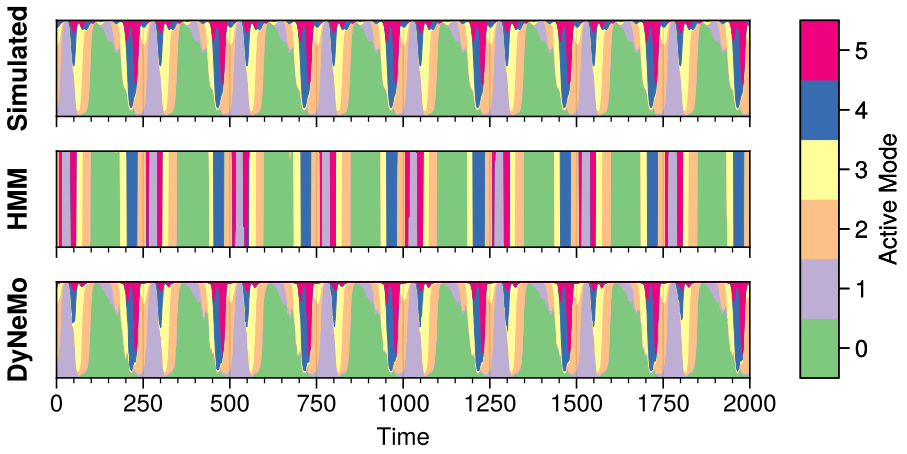
Figure 4.4: State lifetimes are correctly inferred by both DyNeMo and the HMM. However, examining the lifetimes of the timecourses generated by the trained models, it can be seen that the HMM is only able to produce states with lifetimes which exponentially decrease in occurrence as they increase in length. DyNeMo can generate a new mode time course which matches the simulated lifetime distribution well. This demonstrates that the model can learn long-range dependencies from the data.

The mode mixing coefficients (i.e. the mode time courses) were determined by applying a softmax function to a series of linearly combined sine waves (see Fig. 4.5a). The data were then generated by sampling from a MVN with the parameters of the distribution determined by the mode mixing coefficients. An HMM requires that its states are mutually exclusive, and so it should be unable to model the co-activation of multiple modes. Recovering the ground truth parameters of this simulation would demonstrate the expected ability of DyNeMo to infer the co-activation of multiple modes where an HMM would fail.

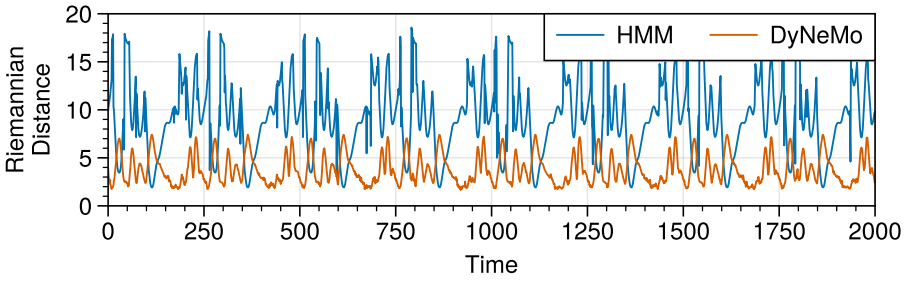
As shown in figure Fig. 4.5b, the HMM inferred a mutually exclusive set of states which were unable to capture the combined activation of the generated modes. In contrast, DyNeMo was able to infer a set of activations which closely matched the ground truth. The Riemannian distance between the inferred and ground truth modes was calculated to quantify the performance of each model. This gave conclusive evidence that DyNeMo outperformed the HMM in this scenario. It can be seen in Fig. 4.5c that when a single state is dominant in the ground truth, there is less difference between the models. However, when multiple states are active, DyNeMo was much closer to the ground truth than the HMM. Over the totality of this simulation, DyNeMo had a mean Riemannian distance of 3.8, while the HMM had a mean Riemannian distance of 9.1.


Mode 0
Mode 1
Mode 2
Mode 3
Mode 4
Mode 5
0
100
200
300
400
Time

(a)


Simulated
HMM
DyNeMo
0
250
500
750
1000
1250
1500
1750
2000
Time
5
4
3
2
1
0
Active Mode

(b)



(c)

Figure 4.5: A series of sinusoids with increasing frequency are created (Fig. 4.5a). These are the mode activation logits of the simulation. Using the softplus function to combine them such that they have a total magnitude of 1 at all time points creates the simulated mode time course in the first row of Fig. 4.5b. The HMM fails to infer the overlapping modes of the simulated dataset (4.5c). This is because it seeks to assign one active state for each time point. In contrast, DyNeMo accurately reproduces the mode time course of the simulation. DyNeMo successfully models coactivating modes.

5 Multichannel DyNeMo

5.1 Introduction

DyNeMo was applied to the Nottingham MEG UK dataset of 51 subjects to explore the dynamics of the brain's response to a visuo-motor task. The subjects provide 332 minutes of data in total, averaging 6.5 minutes per subject. The details of the dataset are described in Section 5.2.

In this chapter, we will consider the spatiotemporal structure of the networks inferred by DyNeMo. By comparing to networks generated by the HMM, we will identify similarities and differences in the networks inferred by each model. The primary shifts from the analysis of the HMM are caused by the introduction of linear mixing factors in place of binary state time courses. This removes the simple notions of state lifetimes and intervals, as well as the ability to select time points from the data which contribute wholly to just one state.

To overcome the loss of lifetimes, we use Gaussian mixture models (GMMs) to determine a probable "on-off" value for each mode. These can be used to binarize the mode time courses in a data-driven manner. In HMM analyses, it is common to calculate quantities such as the power spectral density (PSD) for the selection of the data which is represented by each mode in the state time course. As multiple DyNeMo modes are active at the same time, it is necessary to carry out a linear

regression instead. The alphas for each mode are the regressors against the wavelet transform of the data on which the model was trained. This analysis is equally valid for the HMM, though it adds a layer of unnecessary complexity. We aim to show that the results of DyNeMo demonstrate a different, valid description of the data when compared to the HMM.

By using task data, we will demonstrate that DyNeMo is learning a meaningful description of the data. The correlation of the inferred alphas to the task structure will show that the representation which DyNeMo finds is indicative of underlying structure in the data.

The work in this chapter forms part of the analysis published in the 2022 NeuroImage paper, for which I shared joint first authorship with Chetan Gohil and Ryan Timms.[87]

5.2 The Data

5.2.1 Acquisition

The data used in this thesis were collected at the University of Nottingham as part of the UK MEG Partnership.[88] They were collected using a 275-channel CTF MEG system at a sampling rate of 1.2 kHz.. Third-order synthetic gradiometers were used to measure the magnetic field while reducing the effects of environmental noise. Structural MRI scans were collected using a 7T Philips Achieva scanner.

59 participants were recruited for the study, but only 51 are included in this analysis[1]. Those 51 were distributed as follows:

- 24 male, 27 female
- 46 right-handed, 5 left-handed
- Minimum age: 19yr
- Maximum age: 64yr

[1] Of those excluded, the supplementary materials in Hunt et al. (2018) describes one as follows: One participant in the Nottingham cohort blinked with such vigour as to cause highly disruptive artefacts that could not be removed with post-processing.

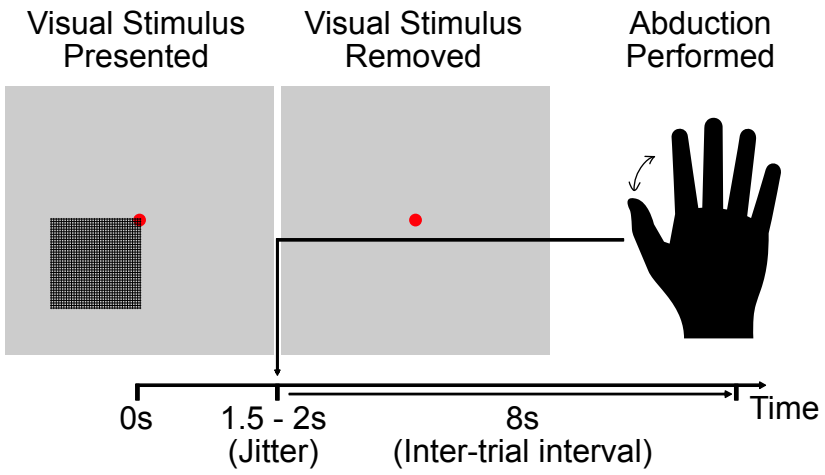


Figure 5.1: Task Structure. Participants are shown a screen with a fixation dot. A grid is presented with its top right corner at the fixation dot. After between 1.5 s to 2 s, the grid is removed. The participant then performs an abduction of the thumb and index finger on the right hand. The inter-trial interval between the removal of the grid and the presentation of the next grid is 8s.

- Mean age: 38.3yr

5.2.1.1 Task

The participants in this study were asked to watch a screen. On the screen, a focus point was constantly present. A high-contrast grating was presented with its upper right corner at the focus point. The grating was presented for between 1.5 s to 2 s with the jitter being randomly selected for each trial. When the grating was removed, the participants were asked to perform an abduction with the index finger and thumb of their right hand. The abduction was measured using an electromyography (EMG) sensor. There was then an 8 s rest period before the next trial began. The structure of the task is shown in Fig. 5.1. The dataset comprises 1837 trials.

5.2.2 Preprocessing

For this thesis, the data were preprocessed using OSL by downsampling to 250 Hz, removing the power line noise at harmonics of 50 Hz, and filtering the data between 0.1 Hz and 45 Hz. We are willing to discard the high-frequency (gamma) content

of the signal because it makes negligible contributions to network dynamics. This is because high-frequency oscillations are used to communicate between local collections of neurons, for example within the visual cortex. OSL was also used to remove segments of data which were excessively noisy.

To transform the data from sensor-space to source-space, a source reconstruction was performed. To do this, a model was created which describes the geometry of each subject's head. Points on the outside of the scalp were digitised using a Polhemus pen. These points were then matched to the subject's scalp from their MRI scan which was extracted using brain extraction tool (BET).[89,90] With the shape of the scalp known, the local spheres method[58] was used to create a forward model which describes the relationship between the magnetic field measured at the sensors and the magnetic field generated by the brain. With the forward model created, a source reconstruction was performed using an LCMV beamformer.

The beamformer was applied to the sensor-level data to produce a time series of activity for each virtual dipole in the brain. The dipoles were created in an isotropic grid with a spacing of 8mm. Orthogonalisation was performed to remove spurious correlations from the dipole time courses.

A parcellation of the brain was used to group the voxels into regions of interest. The parcellation chosen had 42 regions and can be found in the `osl-dynamics` repository[2]. The regions are indexed as follows:

- Medial prefrontal cortex (1)
- Frontal lobes (right: 2-6; left: 38-42)
- Temporal lobes (right: 7-9; left: 35-37)
- Visual cortices (right: 10-13; left: 31-34)
- Sensorimotor cortices (right: 14-16; left: 28-30)
- Parietal lobes (right: 17-20; left: 24-27)

[2] `github.com/OHBA-analysis/osl-dynamics`

- Posterior precuneus (21)
- Posterior cingulate cortex (22)
- Anterior precuneus (23)

The time series for each region was calculated by applying PCA to the time series for all voxels in the region.

At this stage, the data were in a form which could be used for analysis. However, extra steps were taken to improve some additional metrics. Firstly, a time-delay embedding was applied to the data. Using 15 time delays on this dataset produced a time-delay embedding with a dimensionality of 630.

Unfortunately, the embedding step dramatically increases the dimensionality of the data. As the model was run on GPUs, this was a problem. GPU memory (VRAM) is generally more limited than CPU memory (RAM), so the dimensionality of the data had to be reduced. This was done using PCA. Reducing the number of channels to 80 reliably captured 51% of the variance in the data and sufficiently reduced the memory requirements.

Finally, applying a z-transform along the time dimension of each PCA component ensured that the data were all on the same scale. This reduced the influence of outliers and improved the performance of the model. The data were then ready for analysis using DyNeMo.

5.3 Inferred Networks

The covariance matrices which DyNeMo infers are interpretable in two main ways. Firstly, the diagonal elements correspond to the activation of each parcel in the mode. Using these values, we can create power maps which help visualise the spatial distribution of each mode. Secondly, the off-diagonal elements correspond to the functional connectivity between each parcel in the mode. We can also derive these values by regressing the wavelet transform of the data onto the alpha time course for each mode. It is this second approach which we will use here. The

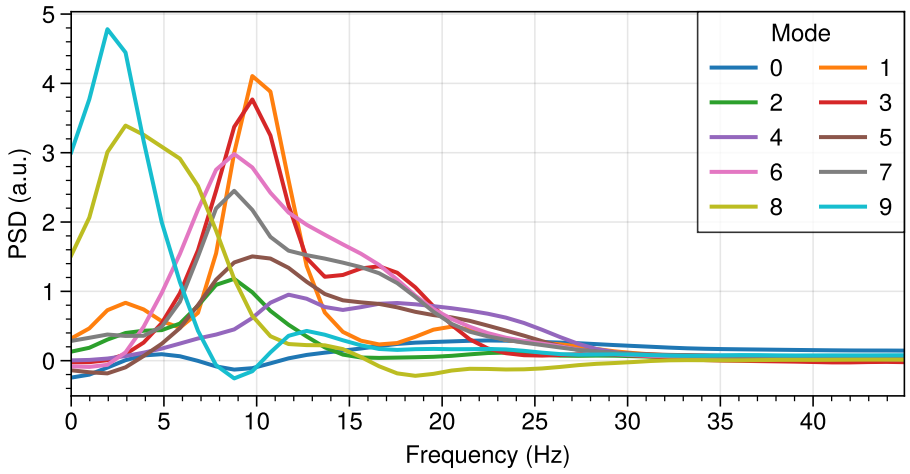


Figure 5.2: PSD of the modes learned by DyNeMo. PSDs show the contribution of each mode to the frequency distribution of the data. Different modes have peaks in the theta, alpha and beta ranges, and each mode's PSD has an individual balance of power across the spectrum. They are calculated by regressing the wavelet transform of the data onto the alpha time course for each mode.

alpha mode time courses also allow us to look for patterns in the activation of modes, but we will come to that later.

We calculate the PSD for each mode by first computing a time-frequency power decomposition of the data, and then regressing this onto alpha time courses separately for frequency. The PSDs correspond to the resulting regression coefficients (Figure 5.12), and show us the contribution of each mode to the frequency distribution of the data. We take the mean over parcels for each mode to get a single value for each frequency. Plotting the PSD for each mode allows us to visualise the differential frequency distribution of each mode (Figure 5.2). We see that modes 2 through 7 show activity in the alpha band (8 Hz to 12 Hz). Modes 4 through 7 also show increased activity in the beta band (13 Hz to 30 Hz). Modes 8 and 9 have large peaks in the lower frequency delta and theta bands (1 Hz to 7 Hz).

If we take the mean of the PSDs over frequencies instead of parcels, we can see the contribution of each mode to the power of each parcel. Projecting these values onto the surface of the brain gives us a power map for each mode (Figure 5.3).

Examining the spatial maps of each mode in turn, we can attempt to align these with known brain networks from the literature.

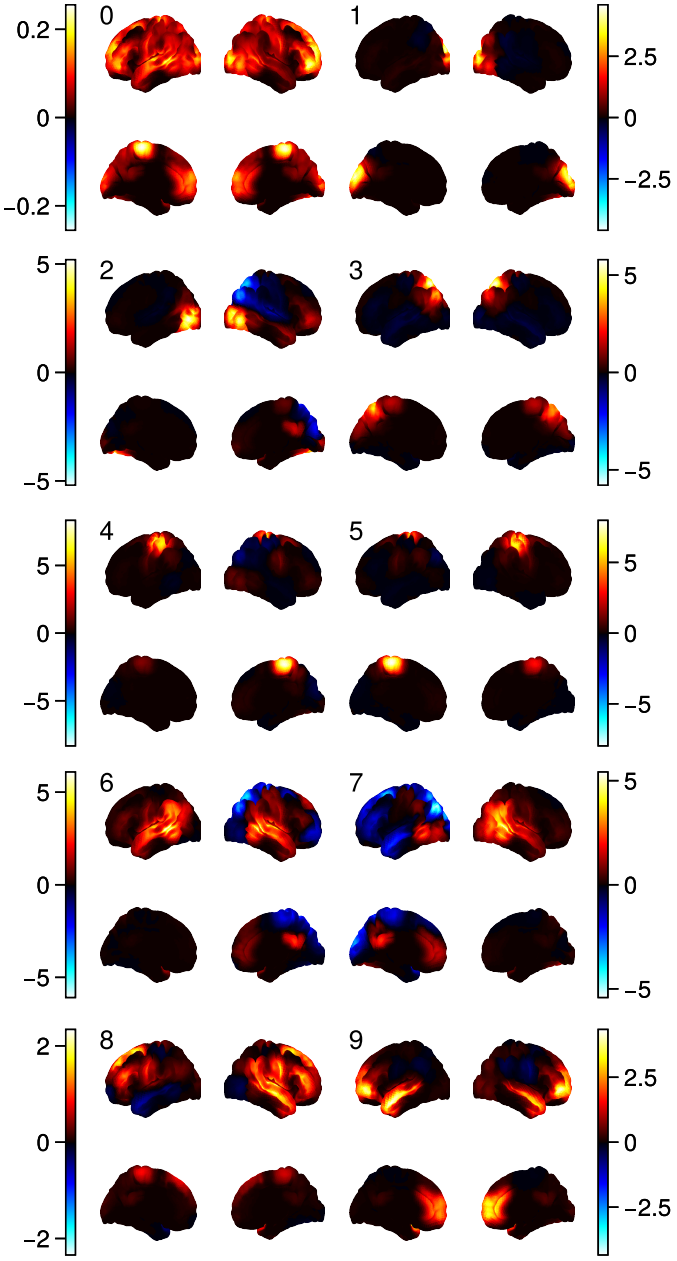


Figure 5.3: Power maps show the projection of power from each parcel onto the surface of the brain for frequencies in the range 1 to 45 Hz. The spatial modes found by DyNeMo are representative of known brain patterns relating to motor activity, visual activity and baseline activity. Mode 0 is a background mode which encapsulates the background activity of the brain.

- Mode 0 is a special case, which will be discussed later.
- Modes 1 and 2 are roughly bilaterally symmetric network, with strong activation in the visual cortex.
- Mode 3 shows symmetric activation in the posterior parietal cortex. This is a notable region in the posterior portion of the default mode network.
- Mode 4 and Mode 5 show activation in the motor cortex, with more connectivity in the left and right hemispheres respectively.
- Mode 6 and Mode 7 have high power in regions associated with language and auditory processing.
- Mode 8 displays power in the frontal lobe which covers a wider area in the right hemisphere. The connectivity shows significant correlation between parcels in the left hemisphere of the parietal lobe. This matches with expectations of the fronto-parietal network, which is involved in decision making and working memory in goal-oriented tasks.
- Mode 9 shows activation in the frontal lobe, with a strong correlation between the left and right hemispheres. This is consistent with the anterior default mode network.

We can also plot the coherence of each mode, which shows the functional connectivity between each parcel in the mode (Figure 5.4). The colour of each line indicates the strength of the correlation between the two parcels. We see that the connectivity of each mode is consistent with the known brain networks which we identified from the power maps. For example, mode 1 shows strong connectivity between the left and right visual cortices. Take note of the connectivity of mode 0, which we will discuss later.

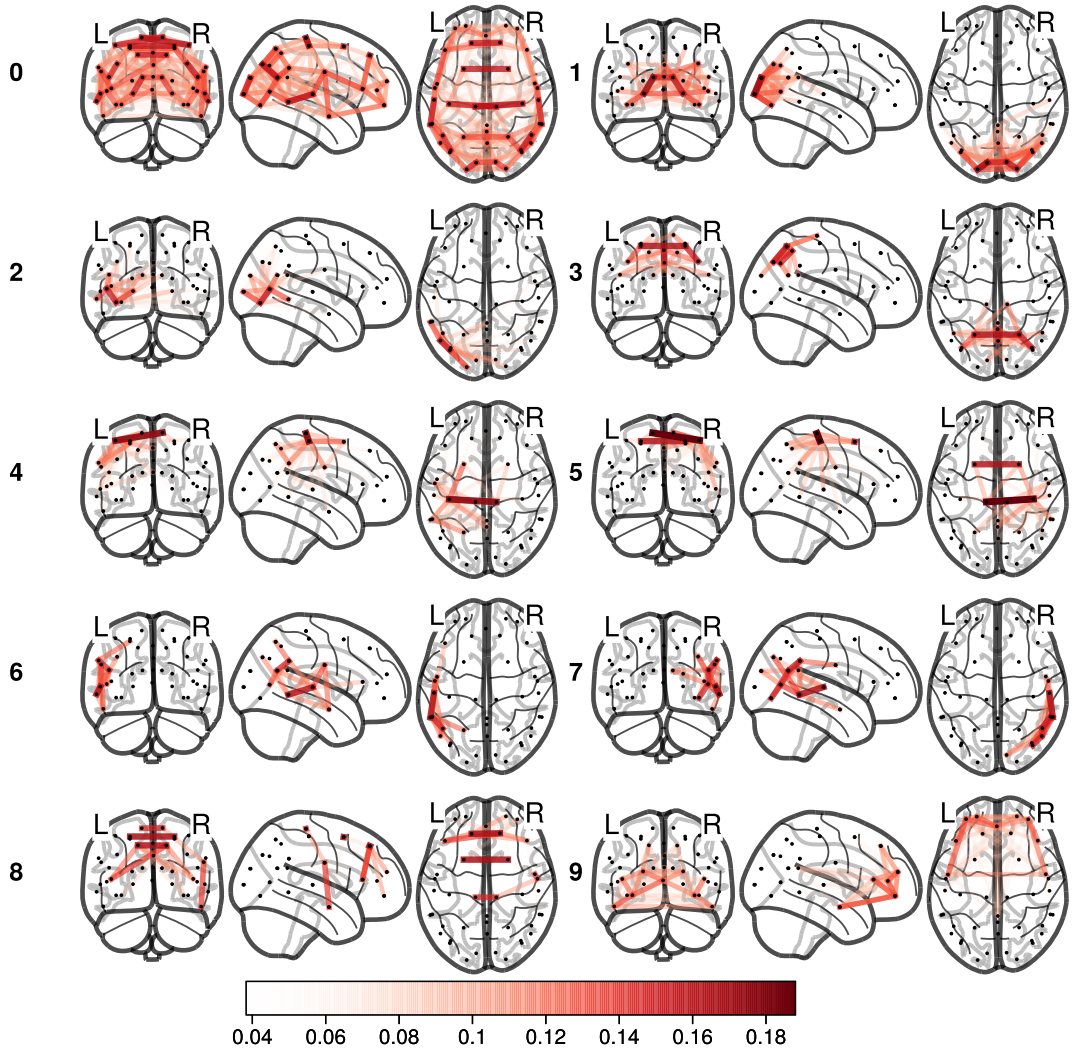


Figure 5.4: Connectivity plots for each mode, showing the functional connectivity between each parcel in the mode. The colour of each line indicates the strength of the correlation between the two parcels. The connectivity of each mode is consistent with the known brain networks which we identified from the power maps. For example, mode 1 shows strong connectivity between the left and right visual cortices. Mode 0 shows connectivity between all parcels, which provides evidence that it is a background mode.

5.4 Background Activity

Mode 0 highlights one key difference in behaviour of DyNeMo compared to the HMM, which has also been seen in resting MEG data analyses using DyNeMo. It shows a power distribution (Figure 5.3) and connectivity profile (Figure 5.4 which is active across the entire brain. It also has a PSD which shows little variation across frequency (Figure 5.2). The next clue to its function is to notice that it has a consistently high mixing coefficient throughout the dataset. Additionally, its activation is anti-correlated with the other modes, as shown in Figure 5.5 All these properties can lead us to interpret the mode as a kind of "background" that is constantly active to a reasonable extent.

In the HMM, such a background state would need either to be active only when no other states were, or to be incorporated into the observation model of every other state. DyNeMo allows for this mode to continue in the background while other modes activate and deactivate. Not forcing this mode to split between the more interpretable modes potentially allows for a set of modes to emerge which are less contaminated by the ongoing background activity.

5.5 HMM Comparison

With our knowledge of the background state, we will now compare the results of DyNeMo with those of the HMM. The HMM was trained on the same data as DyNeMo and the same number of states was used as the number of modes in the DyNeMo model.

The HMM PSD (Figure 5.6) shows a broader distribution of power across frequencies than the DyNeMo PSD (Figure 5.2). This is because HMM states capture the frequency distribution for the entire brain when they are active, whereas DyNeMo modes only capture the frequency distribution for the partial brain network which they represent. We still see peaks in the delta, alpha and beta bands, but these are more smoothly mixed with a background frequency distribution. In

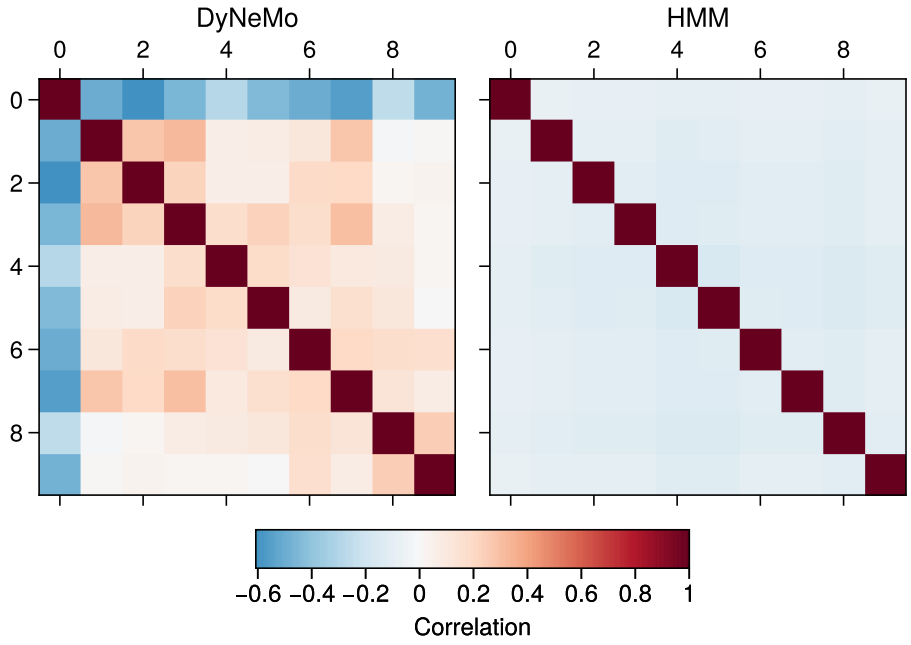


Figure 5.5: All states inferred by the HMM are anti-correlated. This is a natural consequence of the mutual exclusivity of the states. It is impossible for them to be correlated as they are not coactive. In contrast, modes inferred by DyNeMo can be correlated as they are allowed to be coactive due to the assumption that the alpha time courses are a linear mixture. Mode 0 is demonstrated to be a background state through its anticorrelation with the other modes. This shows that it reduces its contribution to the data when other modes activate.

the DyNeMo model, the background activity is captured by a single mode, which allows the other modes to be more specific.

Turning to the power maps (Figure 5.7), we see that the HMM states are more diffuse than the DyNeMo modes. They are also more similar to eachother, which is again caused by their mutual exclusivity. The HMM states are forced to capture the entire brain, and so they cannot be as specific as the DyNeMo modes. We can partially overcome this by subtracting the mean power over all states from each state's power map. This mean is a coarse proxy for the background activity. The resulting power maps (Figure 5.8) are more similar to the DyNeMo modes, but they still show more diffuse activation. Notably, the HMM states do not separate the motor cortex into left and right hemispheres, as the DyNeMo modes do. The power maps inferred by DyNeMos are more mode-specific than the power maps inferred by the HMM. This is because the HMM states are forced to capture the activity

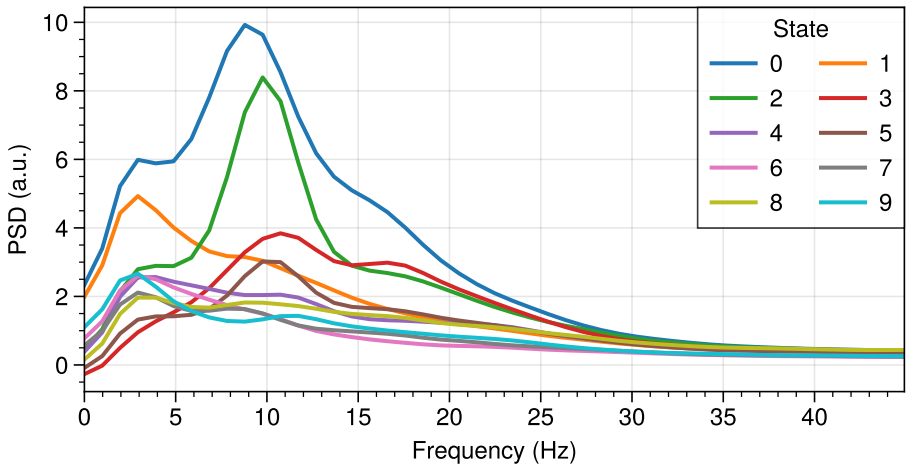


Figure 5.6: PSDs of the states learned by the HMM when trained on multichannel, task data. PSDs show the contribution of each state to the frequency distribution of the data. Different states have peaks in the theta, alpha and beta ranges, and each state's PSD has an individual balance of power across the spectrum. These PSDs are more diffuse than the DyNeMo modes, because the HMM states are forced to capture the activity of the entire brain. They are calculated by regressing the wavelet transform of the data onto the alpha time course for each state.

of the entire brain, whereas the DyNeMo modes are only required to capture the activity of a subset of the brain. There is less cross-contamination between the modes in the DyNeMo model, which allows for more specific modes. This means that they form a more orthogonal basis set from which to construct the data.

The connectivity of the HMM states (Figure 5.9) shows strong connectivity in the visual cortex for states 2 and 5, which is consistent with the power maps. State 3 shows connectivity in the motor cortex which is also consistent. The other connectivity maps are largely sparsely populated and do not show any clear patterns.

5.6 Recovery of Resting State Networks

In addition to the data recorded during the task, the Nottingham dataset includes resting state data. In these recordings, participants were asked to do nothing but keep their eyes open. These data were analysed (analysis of rest data was carried out by Chetan Gohil) in an equivalent manner to the task data, and the relative power maps are shown in Figure 5.10.

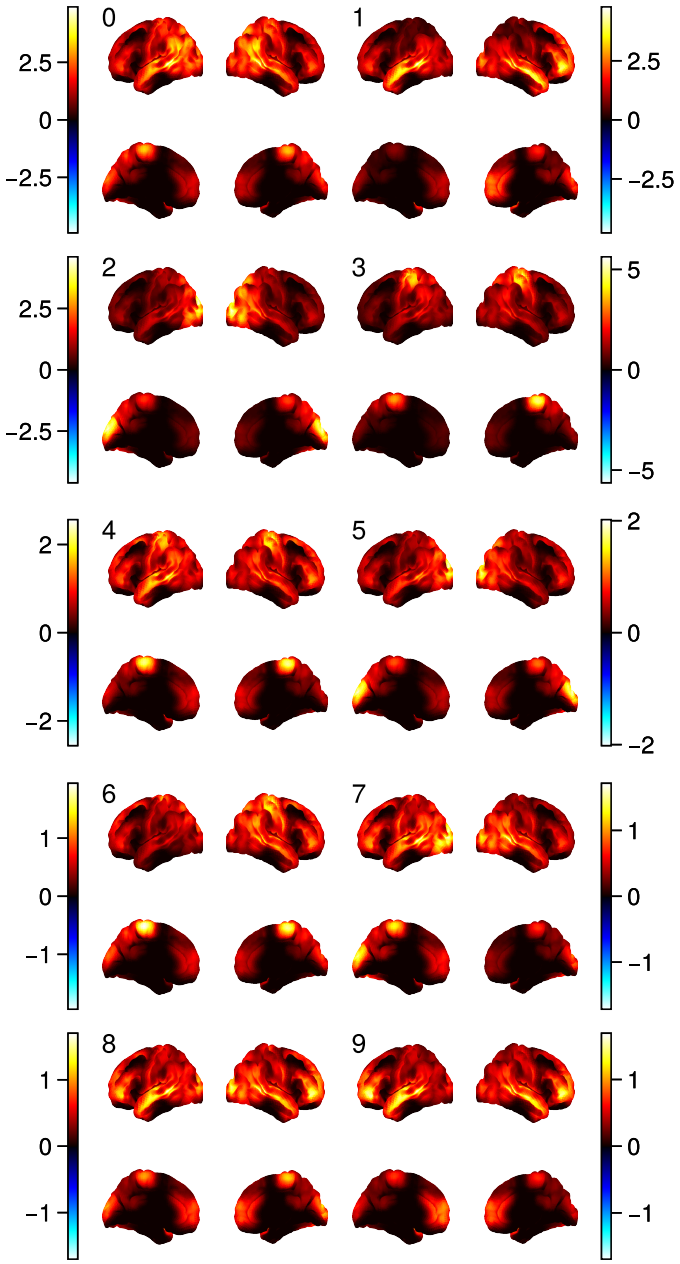


Figure 5.7: Power maps show the projection of power from each parcel onto the surface of the brain for frequencies in the range 1 to 45 Hz. The spatial states found by the HMM are more diffuse than the DyNeMo modes. For example, we see motor parcels active in all states, whereas the in DyNeMo it is only seen in specific modes.

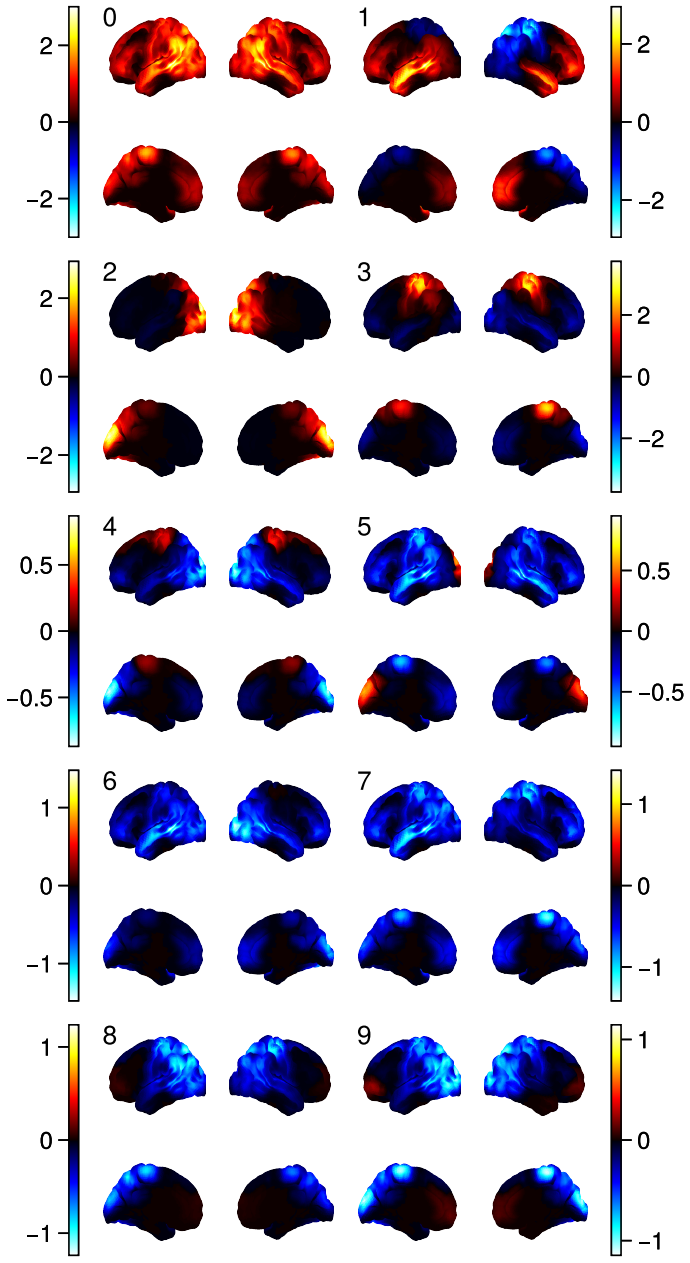


Figure 5.8: Power maps show the projection of power from each parcel onto the surface of the brain relative to the average power over all states for frequencies in the range 1 to 45 Hz. The spatial states found by the HMM are more diffuse than the DyNeMo modes. Subtracting the mean power over all states from each state's power map gives a better comparison with the DyNeMo modes. The mean is a coarse proxy for the background activity which is distributed across all states.

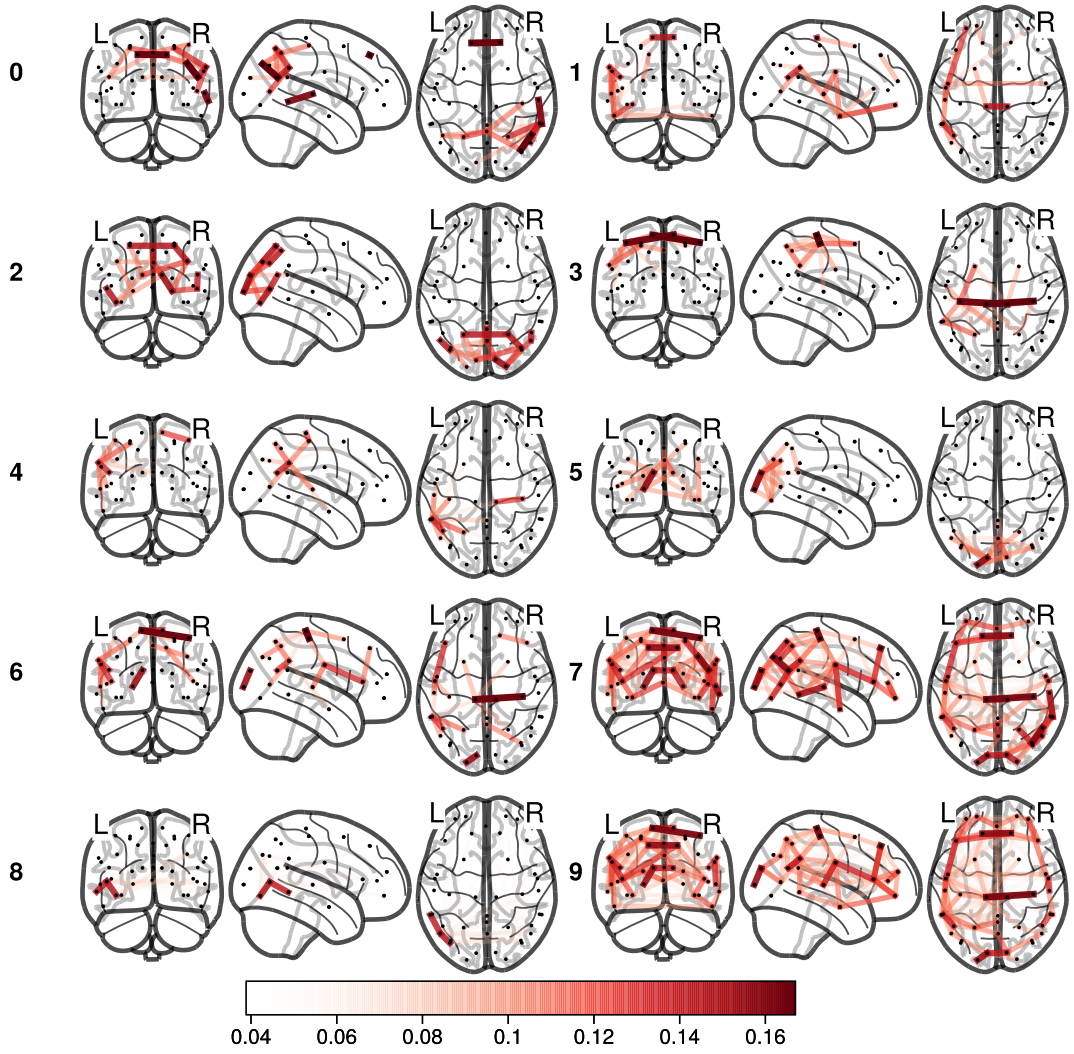


Figure 5.9: Connectivity plots for each state, showing the functional connectivity between each parcel in the state. The colour of each line indicates the strength of the correlation between the two parcels. The connectivity of each state is not as clear as the connectivity of the DyNeMo modes, but there are some notable patterns. For example, state 2 shows strong connectivity between the left and right visual cortices.

Comparing the results of the two analyses, it can be seen that there is considerable overlap in the modes found during rest and task. Although resting state modes are recruited in the task analysis, there are some differences in the modes found. Importantly, when trained on task data, DyNeMo splits the sensorimotor network by hemisphere. This implies that the increase in activation of this region caused by the task activity leads to the split mode being a better explanation of the data. This is facilitated by DyNeMo's ability to model spatiotemporally overlapping modes.

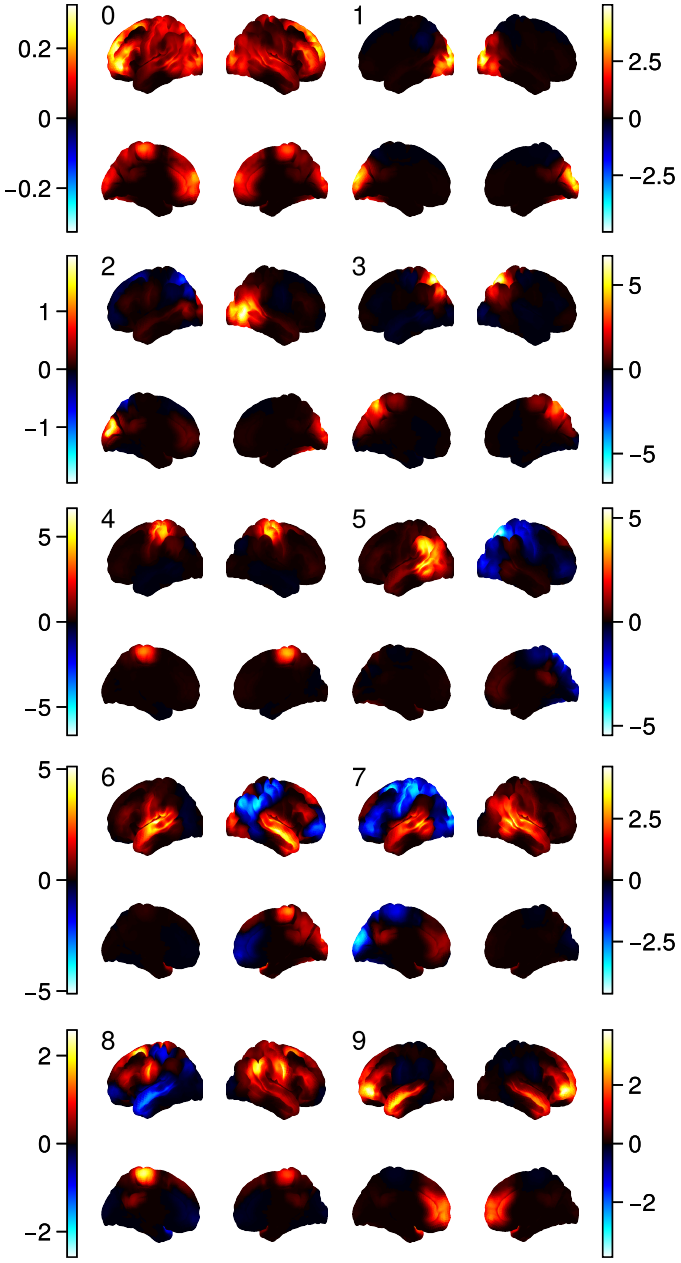


Figure 5.10: The task networks recovered by DyNeMo are similar to those discovered in resting state data. The primary difference is the splitting of the visuo-motor state into two lateralized components in the task. This may indicate that the model was able to justify an additional mode to represent these regions due to the increase in relevant data. The background mode is present in both rest and task.

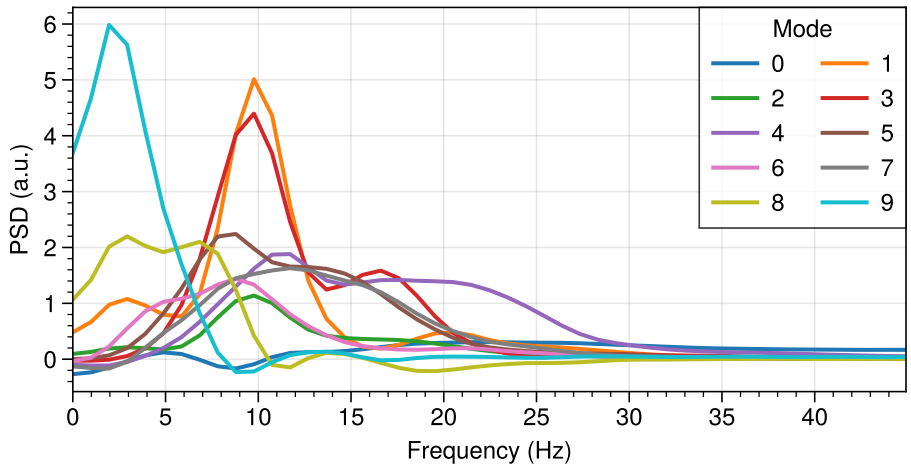


Figure 5.11: The PSD of the resting state modes shows peaks in the delta, alpha and beta bands. The alpha band is less prominent than in the task data, which is may be because as the resting state data was recorded with the eyes open.

We also see less activity in the alpha band in the resting state data. This data is recorded with the eyes open, and so the alpha band is not as prominent as it might be in a closed-eye resting state recording. The PSD of the resting state modes is shown in Figure 5.11.

5.7 Mixing Coefficients

While the alpha time courses inferred by DyNeMo express how much each mode contributes to the linear mixture of modes, i.e. the mode mixing coefficients, they do not express how much each mode contributes to the total power of the data. The power is determined by a combination of the mode mixing coefficients and the mode variances, i.e. the diagonal elements of the mode covariances. To find a measure of the contribution of each mode to the total power, we can normalise the mode mixing coefficients by the trace of the mode covariance.

By once again setting the sum at each time point to 1, we can find the relative contribution of each mode to the total power. This is shown in Fig. 5.12. It now becomes evident that all the modes contribute roughly equally to the power on average, though at any given time point, the contribution of each mode can vary significantly. This is critical for cogent analysis of the alphas as they are more

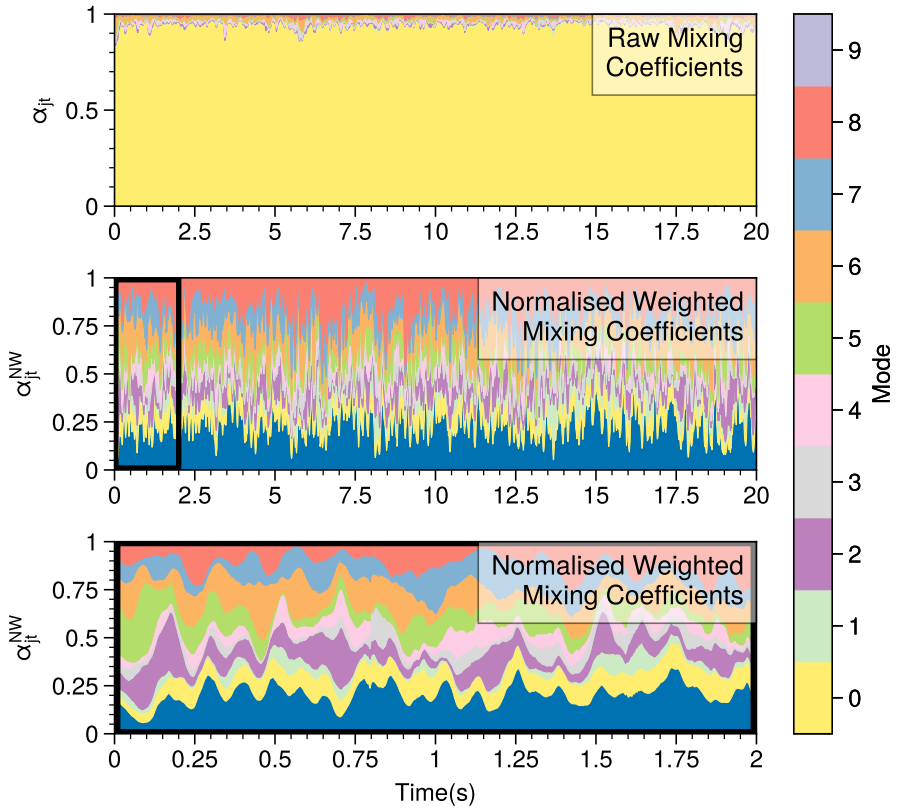


Figure 5.12: The alphas inferred by DyNeMo are typically dominated by a single mode - the background mode. While this shows the contribution of each mode, it does not reflect the power output. The background mode, which has consistently high mixing coefficients, has low variance and so contributes little to the total power. The variance of each mode is correlated to its power. Using the trace of the corresponding covariance matrix, each mode can be weighted by its power. This creates a more balanced description of the alphas with roughly equal contributions. These weighted alphas must be renormalised to ensure that the total alpha value at each time point is still equal to 1.

naturally comparable. In this regime, the background mode no longer dominates the mixture as it has a low covariance trace. The balance between the other modes becomes more readily apparent.

5.8 Evoked Response

As we are working with data from a task MEG experiment, we can see how the alphas inferred by DyNeMo relate to the task, we can plot the alphas against the task events. Although, we note that DyNeMo infers the alphas in a completely unsupervised manner, with no knowledge of the task timings, and so any task

structure in the alphas has been automatically detected by DyNeMo.

To examine the task structure in the alphas, we can perform similar analyses on the alpha time courses that are performed on, e.g. sensor data time courses. Specifically, we first chop up each alpha's time course into windows around each task event and label this an epoch. For example, when considering the presentation of the visual stimulus to the participant, we can take a window of one second before and four seconds after the stimulus. This gives a five second epoch which allows us to see the changes which occur around the event of interest.

We can visualise the trial-wise alphas for a given mode by plotting a kind of heatmap, where each row is a trial and each column is a time point. This gives the impression of a complex barcode. Examining the barcode in Figure 5.13 shows us general trends in the alphas, but also allows us to see the variability between trials. For a more quantitative analysis, we can average the alphas across trials for each time point, to compute an "evoked" response as shown in Figure 5.14. Doing this shows us the pattern of mode activation for each mode around task events.

We are primarily interested in two task events: the presentation of the visual stimulus and the performance of the abduction. As there is a jitter in the timing of the removal of the visual stimulus which triggers the abduction, we need to consider these separately. We also choose a baseline period to do baseline correction, so that we simplify the comparison of activity. In this instance, we use one second prior to the presentation of the visual stimulus, as this is as close to a resting state as we can expect to get.

First, we consider the visual stimulus. Upon presentation, we see a strong increase in the activation of mode 2, which we identified as the visual cortex. Mode 3 is also activated, but to a lesser extent. As time progresses, we see modes 2 and 3 drop below the baseline before returning to their background level. Further activation is seen in modes 4 and 8, and then in mode 9. This represents the movement of activation from the visual cortex, through a broad activation in the posterior brain and then into more frontal areas.

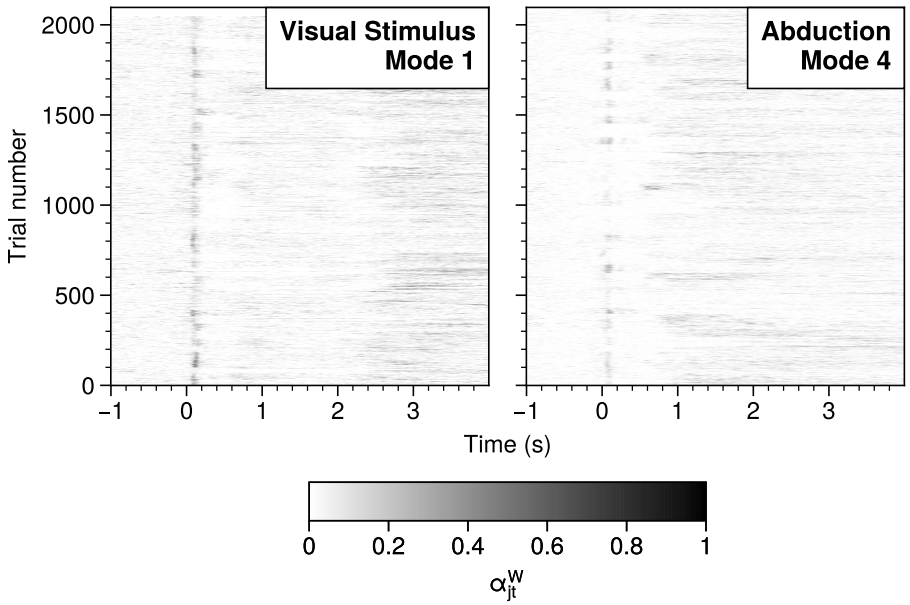


Figure 5.13: DyNeMo's mode activation time courses (the alphas) can be visualised on a trial-wise basis, and not just as a trial-averaged phenomenon. This barcode chart shows the activations of relevant modes for each trial for modes relevant to visual stimulus presentation and abduction. Moving away from the binarised state model of the HMM allows DyNeMo to model multiple active modes at once. This reduces the disruption to the inference of any given mode by the presence of another. In practice this means that while a mode may reduce in contribution due to the activation of another state, the activation of any mode will be more reliable across trials.

Next, we consider the window around the abduction. A sharp spike is seen in mode 5, one of the modes we noted early that has strong motor cortex activity. Also note the large dip in mode 4, which we identified as a part of the default mode network.

As mentioned above, when DyNeMo is trained, it is not given any information about the task events. Any correlation between the inferred alphas and the task events is purely a result of the model's own ability to discover any structured dynamics in the data. Using a sign-flip permutation test, we can test the significance of the correlation between the alphas and the task events for each time point. Defining significance to be regions with a p-value of less than 0.05, we mark the significant regions with a red highlight.

Looking once again at the barcode plots in Figure 5.13, we can see consistent trial-wise activation of modes 2 and 5 in their respective windows. This shows

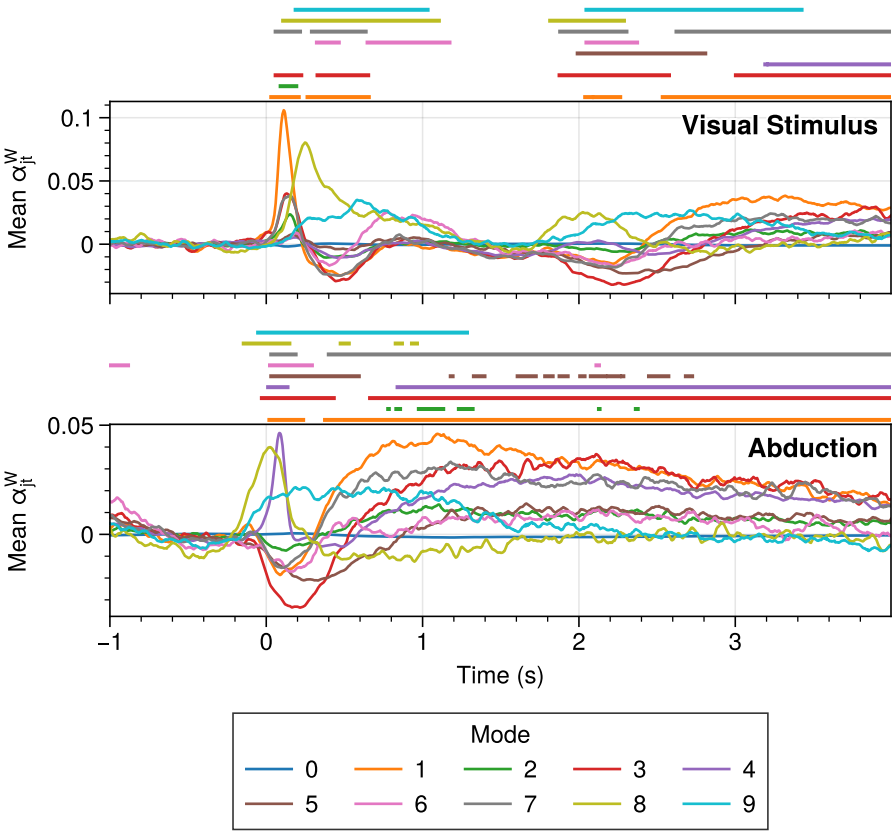


Figure 5.14: Epoching the inferred alphas from DyNeMo and taking a mean across each trial in the dataset shows a considerable task dependence. The alpha time courses were weighted by variance and renormalized prior to epoching. Additionally, we do baseline correction, where the baseline value of each mode (taking the average of the alpha from -1 to 0 secs prior to stimulus presentation) is subtracted from each mode. In the top plot, sharp rises can be seen in modes 1, 3 and 7 with a gentler rise in modes 8 and 9 all correlated with the presentation of the visual stimulus to the participant. The bottom plot shows large spikes in modes 4 and 8 when the participant performs an abduction in response to the removal of the stimulus. There is also a large drop in activation of mode 3. The coloured lines above each plot indicate the significance of the correlation between the alphas and the task events. When the lines are present, the relevant mode has a p-value of less than 0.05 according to a sign-flip permutation test.

that the states are active on many trials individually. As the HMM only captures binary activation, this trial-wise activation tends to be less consistent. The activation of the expected mode can be easily obscured by activity from some other mode. DyNeMo provides a more robust measure of activation, and so we can be more confident in the results.

5.8.1 Comparison with HMM

Let's compare once again with the HMM. We can plot the HMM states in a similar manner to the DyNeMo modes. Starting with the mean activation of each state across trials (Figure 5.15), we see that the HMM states also show strong task dependence. The HMM states show a similar pattern of activation to the DyNeMo modes, but there is more moment to moment variability. This is because the HMM states are binary, and so the activation of a state is either on or off. The mean activation of a state is equivalent to the probability of the state being on over all trials. The DyNeMo modes are a more literal representation of the partial activation of the brain network.

Considering states 2 and 5, we can look at the barcode plots in Figure 5.16. We see that the HMM states are not as consistently active as the DyNeMo modes. While there is a greater probability of activation of the state around the task event, there is still a lot of variability. This is a key conceptual distinction between the HMM and DyNeMo.

5.9 Time-Frequency Reconstruction

Capturing both the dynamic activations of the modes and their static power spectral densities allows us to reconstruct a dynamic spectral profile of the data. By comparing this to the time-frequency information of the original data, we can calculate the reconstruction error and get a sense of the comparative spectra captured by different models.

We construct a time-frequency representation of the data by computing the Welch spectrogram of the data. This is a time-frequency representation of the data which is computed by taking the Fourier transform of overlapping windows of the data. The window size and overlap (via step size) are chosen to balance the temporal and spectral resolution of the spectrogram. In this analysis we use a window size of four seconds and a step size of 20 samples. The alphas are downsampled using a Hann window to match the resolution of the spectrogram. A

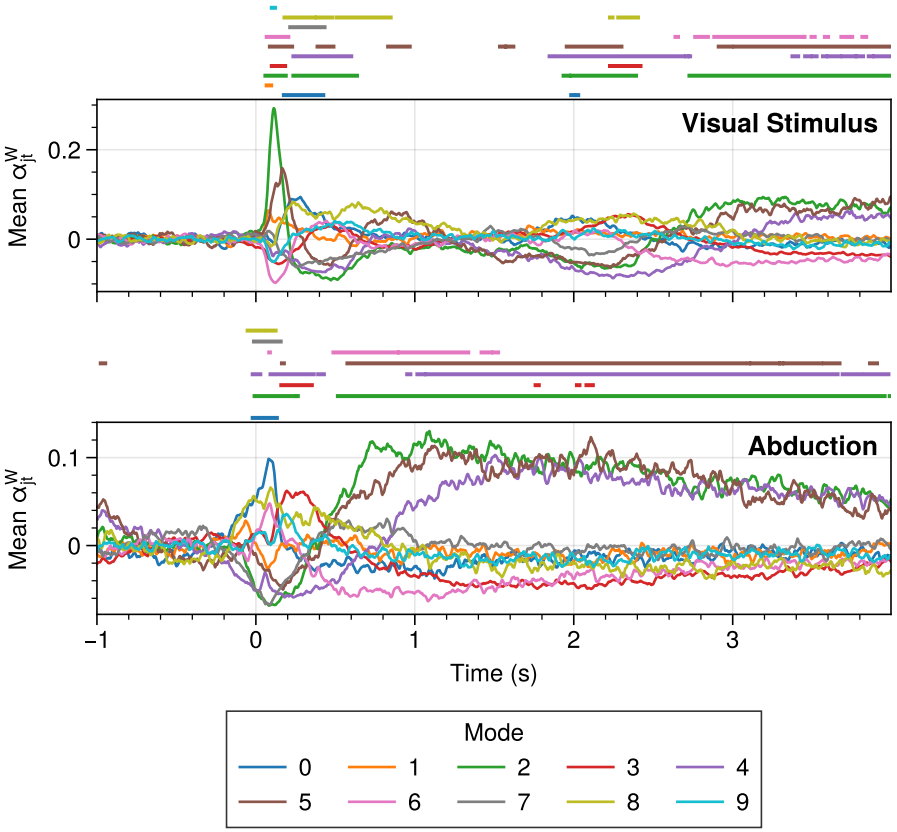


Figure 5.15: Epoching the inferred states from the HMM and taking a mean across each trial in the dataset shows a considerable task dependence. The state time courses were weighted by variance prior to epoching. Additionally, we do baseline correction, where the baseline value of each state (taking the average of the state from -1 to -0.1 secs prior to stimulus presentation) is subtracted from each state. In the top plot, sharp rises can be seen in states 2 and 5 with a gentler rise in states 0 and 8 all correlated with the presentation of the visual stimulus to the participant. The bottom plot shows large spikes in states 0 and 3 when the participant performs an abduction in response to the removal of the stimulus. There is also a large drop in activation of state 4. The coloured lines above each plot indicate the significance of the correlation between the states and the task events. When the lines are present, the relevant state has a p-value of less than 0.05 according to a sign-flip permutation test.

regression is performed between the spectrogram and the alphas to find the best frequency representation of the alphas.

Considering spectrograms epoched by the task structure, we can see the broad strokes of expected spectral patterns. Reductions in recorded beta power are seen during movement. These are recreated by both DyNeMo and the HMM as shown in Figure 5.17. DyNeMo's time-averaged PSD has a correlation of 0.79 with the PSD of the original data for the visual task window and 0.90 for the abduction window

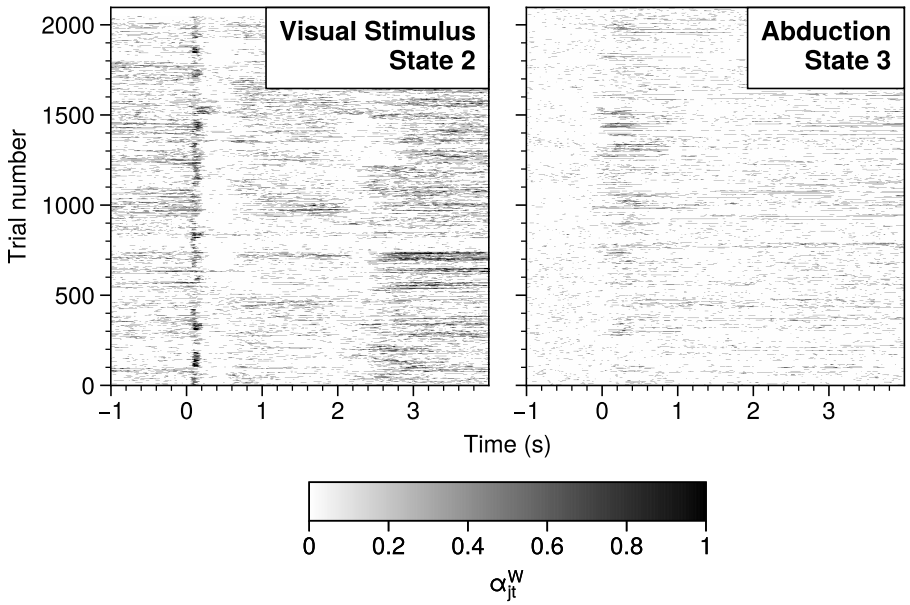


Figure 5.16: HMM states can be visualised on a trial-wise basis, and not just as a trial-averaged phenomenon. This barcode chart shows the activations of states relevant to visual stimulus presentation and abduction. The HMM states are not as consistently active as the DyNeMo modes. While there is a greater probability of activation of the state around the task event, there is still a lot of variability. This is a key conceptual distinction between the HMM and DyNeMo.

compared to 0.73 and 0.88 for the HMM. This is only a subtle improvement, but the trial averaging hides potential higher inaccuracies the HMM may have at the trial-wise level, and at least shows that DyNeMo is capable of modelling the spectral content of data at least as well as the HMM.

5.10 Summary

In this chapter, we have seen that DyNeMo can be applied to multichannel data to learn a set of modes which capture the spatiotemporal structure of the data. The modes are learned in a data-driven manner, and so they are not constrained to be spatially or spectrally independent. This allows them to capture the complex interactions between brain regions and frequencies which are present in the data. The modes are also not constrained to be mutually exclusive, which allows them to capture the overlapping nature of brain activity. This is particularly important for the analysis of multichannel data, as the brain is a complex system

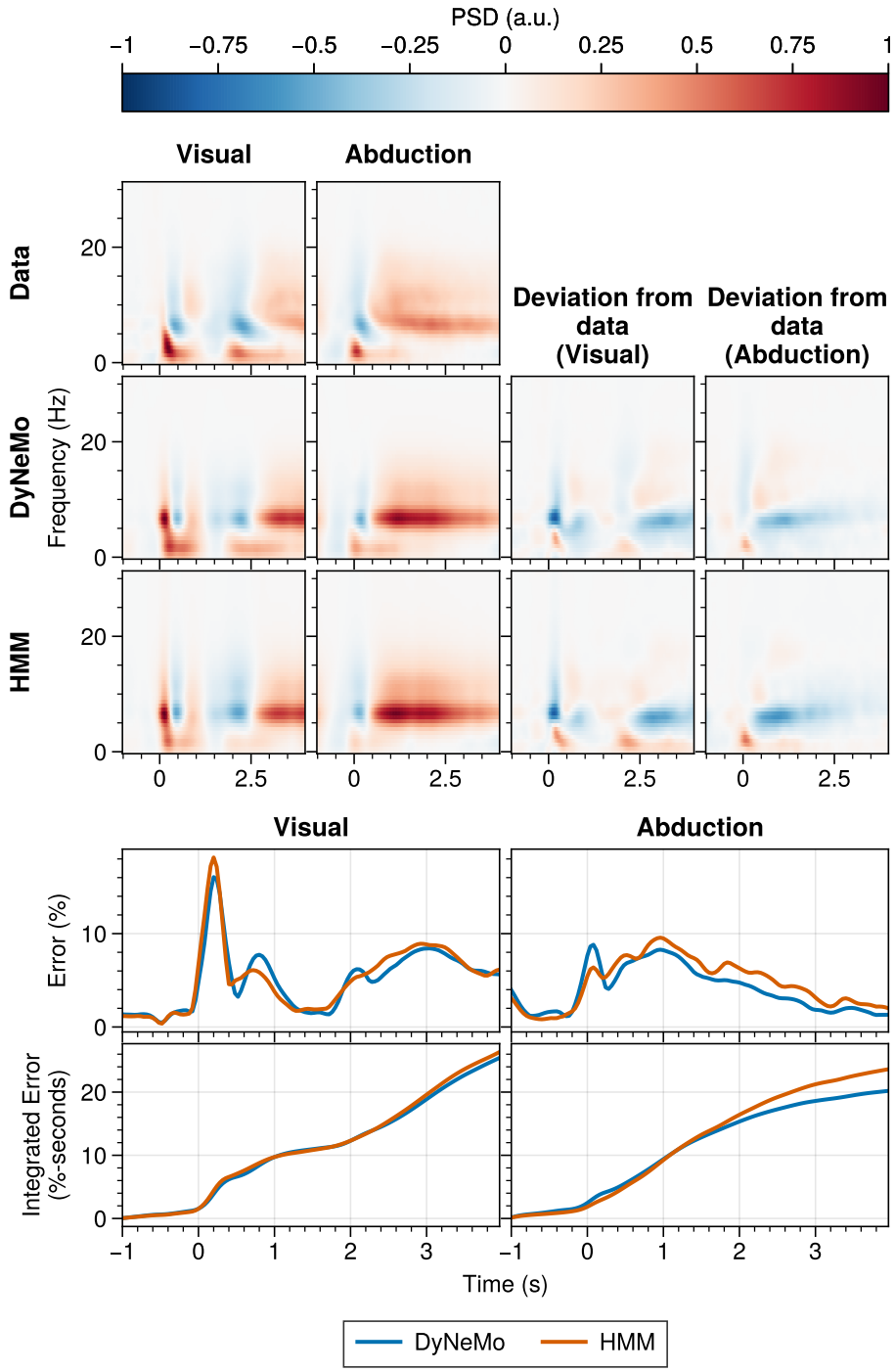


Figure 5.17: Time-frequency reconstructions demonstrate that both the HMM and DyNeMo can capture the frequency content of inferred modes. Although the models find different representations of the data, both reconstruct the time-frequency information to a similar standard. The first row shows the original data. The second row shows the reconstruction from DyNeMo including a comparison of the PSDs of the reconstruction and the original data. The third row shows the reconstruction from the HMM including the same comparison. The fourth row shows the reconstruction errors over time. The fifth row shows the reconstruction errors integrated over time.

with many interacting regions.

The modes are learned in a way which is consistent with the task structure of the data, and so they can be interpreted in terms of the underlying brain activity. This is demonstrated by the correlation of the mode alphas with the task structure of the data. The modes found by training DyNeMo task data are consistent with the modes found by training resting state data. This suggests that the modes are capturing some underlying structure in the data, rather than simply learning the task structure.

The modes learned by DyNeMo are compared to the states learned by the HMM. The HMM states are forced to be mutually exclusive, and so they are more spatially diffuse than the DyNeMo modes. The HMM states are also forced to capture the entire brain, and so they are less specific than the DyNeMo modes. Unlike HMM states, DyNeMo modes do not need to be modified the best visualise the data. We need to subtract the mean power over all states from the HMM states to make them comparable to the DyNeMo modes. This mean is a coarse proxy for the background activity which is distributed across all states. Crucially, the states inferred by DyNeMo are more mode-specific than the states inferred by the HMM. The reduction in cross-contamination between the modes in the DyNeMo model allows for "cleaner" modes which are more specific to the data. This means that they form a more orthogonal basis set from which to construct the data.

When analysing the frequency structure of the data, we see that although the HMM states and DyNeMo modes have differing PSDs, they both show peaks in the delta, alpha and beta bands. HMM states have a broader distribution of power across frequencies, because they are forced to capture the dynamics of the entire brain.

Overall, DyNeMo has been shown to be a powerful tool for the analysis of multi-channel data. It captures a different, valid description of the data when compared to the HMM. Removing the constraints of mutual exclusivity and independence allows DyNeMo to capture the complex interactions between brain regions and frequencies which are present in the data.

6 Single-channel DyNeMo

6.1 Introduction

In the multi-channel analysis carried out in Chapter 5, the entire dataset from all brain parcels was fed into the HMM or DyNeMo as a multi-channel dataset. However, it is also possible to investigate brain dynamics by running multiple, separate single channel time series (e.g. HMM or DyNeMo) analyses, and then running a post-hoc analysis to combine information over the results from the different channels.

In their 2020 paper,[91] Seedat et al. took this approach using the same dataset we described in Section 5.2, and trained a series of HMMs on each channel individually. Using this method, they aimed to identify a set of states which were representative of beta bursts. The proposition is that functional connectivity is driven by bursting, and so looking for coactivation of beta burst states will reveal the functional connectivity of the brain. They found that the HMM was able to identify a set of states, some of which captured beta burst dynamics.

Through comparison with the amplitude envelope of the beta band of the training data, they were able to show that there was high correlation between the beta burst states and the beta band amplitude. By examining the coincidence of bursts, they showed that they could reconstruct more than 90% of the correlation in the

amplitude envelope correlation matrix using around 10% of the data. This is an interesting result, and demonstrates that a multiple, single channel approach can be appropriate for gaining insights into multi-channel MEG data.

However, it is important to realise that DyNeMo, with its continuous, mixed mode assumption, does not naturally lend itself to burst detection in the same manner as the HMM. Instead, the focus of this chapter will be on taking advantage of other benefits of multiple, single channel analyses using DyNeMo.

6.1.1 Potential Benefits

6.1.1.1 Model Memory Requirement

For the parcellation and number of embeddings we are using, training on a single channel at a time causes a reduction in the number of channels by a factor of 78, from

$$n_{\text{channels}} \times n_{\text{embeddings}} = 78 \times 23 = 1794$$

to

$$n_{\text{embeddings}} = 23.$$

As the size of the covariance matrix scales with the square of the number of time series, reducing by a factor of 78 reduces the number of parameters by 6084, a not inconsiderable number, particularly when the multiplicative factor of batch size is considered.

In practice, the high number of channels (1794) in a standard, multi-channel analysis is reduced for computational reasons using PCA dimensionality reduction (typically to ∼100).[87] However, this comes at the considerable cost of reducing the explained variance of the data.

The number of parameters in the RNN also scales, but as this is a linear relationship, the effect is not as pronounced. Training on a single channel allows us to forgo the PCA dimensionality reduction, and so retain more of the explained variance of the data. Additionally, the memory usage of single-channel models does not scale with the number of channels and so a much larger number of channels could be used.

6.1.1.2 Post-hoc Mode Aggregation

When DyNeMo is trained separately on multiple, single-channel data, the degrees of freedom of the combined models are larger than when trained in the multi-channel regime. This is because the single channel alphas are not constrained to match between parcels. In the multi-channel regime, all the parcels share a set of alphas, whereas in the single channel regime, the alphas are fully decoupled, though correlation is potentially still expected. This decoupling leaves the user free to select a set of conglomerate modes which aggregate the parcel modes into whole brain modes. The conglomerate modes may be selected using whichever criterion is desired, and so present a flexibility in the analysis that is not available after training a multi-channel model. In this chapter, we propose some candidate approaches for doing this post-hoc analysis to find what we refer to as "meta-modes".

6.1.1.3 Observation Model Constraints

As mentioned above, multi-channel DyNeMo is necessarily used in combination with PCA dimensionality reduction. The PCA dimensionality reduction makes it difficult to impose appropriate constraints on the mode-specific covariance matrices. In this chapter, we propose an approach that imposes constraints on the mode-specific covariance matrices in the single-channel regime such that DyNeMo finds modes that have distinct autocorrelation functions (i.e. distinct power spectral shapes/profiles).

6.1.2 Potential Drawbacks

As is generally the case, there is no "one size fits all" method for examining brain network dynamics in MEG and EEG data. Compared with the multi-channel regime, one disadvantage of the multiple, single-channel regime is that the latter is not based on an end-to-end generative model. Generative models have the benefit of encapsulating assumptions in a mathematically precise manner, as compared to more heuristic approaches that are defined as a chain of analysis steps. There are also potential benefits from the ability of the multichannel model to impose

sensible constraints on the inferred single channel information (i.e. regularisation) when inferring all-in-one, particularly when handling smaller datasets or low SNR.

6.1.3 Chapter Overview

In this chapter, we will develop a pipeline for benefitting from the potential of a multiple, single region approach to analysing MEG data using DyNeMo. This consists of two main contributions: 1) An approach for finding modes in single channel data that have distinct autocorrelation functions (i.e. distinct power spectral shapes/profiles), and 2) Approaches for doing post-hoc analysis that can group the multiple, single-channel DyNeMo fits to find aggregated whole-brain modes, which we refer to as **meta-modes**.

6.2 Methods

6.2.1 Data Preparation

For the single channel analysis, the data were prepared in a similar way to the multi-channel dataset (Section 5.2). To match the work of Seedat et al., the data were downsampled to 100 Hz rather than 250 Hz. Additionally, rather than the 42-region parcellation, a 78-region parcellation based on the automated anatomical labelling (AAL) atlas[92] was used.

In the default configuration, the data are stored in a directory with a numpy file for each subject. Each file contains a two-dimensional array where the first dimension is time and the second dimension is the channel. For the single channel analysis, each channel had its own directory. Within each channel directory, there was a numpy file for each subject. Each file contained a one-dimensional array of the data for that channel. This was done to allow the data to be loaded into memory more efficiently.

With the data loaded, they were standardised using a z-transform along the time dimension. A time-embedding was then applied to the data, this time using 23 lags rather than 15. Below is a table comparing the two datasets.

	Multi-channel	Single Channel
Sampling Rate	250 Hz	100 Hz
Parcellation	42 regions	78 regions
Time Embedding	15 lags	23 lags
Final Channels	80 (PCA)	23 (embeddings)
Beamformer	LCMV	LCMV
z-transform	✓	✓
Symmetric Orthogonalisation	✓	✓

Table 6.1: Comparison of the preprocessing performed for the multi-channel and single channel datasets.

6.2.2 Parameter Choices

While the majority of the data preparation steps for doing single channel (i.e. single brain region) network dynamic analyses from Seedat et al. were replicated, some changes were made to the model parameters. Beyond the most obvious change of using DyNeMo instead of an HMM, more modes were used than in Seedat's paper. With the HMM, a higher number of states could lead to unrepresentative state switching which could have obscured the true dynamics.

With DyNeMo, this was less likely to be a problem as the non-mutually exclusive nature of the modes means that spurious switching is not often a dominating factor. Choosing more modes also allowed us to capture more of the dynamics of the data, allowing for those dynamics to be more precise. Six modes were chosen, double the number of states used by Seedat.

6.2.3 Modes with Distinct Autocorrelation/Power Spectral Profiles

Here, we propose a modification of the DyNeMo observation model that can potentially find modes with distinct autocorrelation (i.e. distinct power spectral profiles). Finding mode-specific autocorrelation is an attractive approach, as it corresponds to finding mode-specific power spectra that cannot be driven by potential changes in the overall variance of the mode (which is the case for standard multi-channel

DyNeMo, where mode-specific covariances are used). In short, it is purely the profile or shape of the spectra, i.e. the relative amount of power in different frequency bands, that define each mode.

The observation model parameters of DyNeMo are the covariance matrices which define the Gaussian distributions of each mode. However, recall that we feed time-delay embedded versions of the parcel data into multi-channel DyNeMo.[87] The reason for this is that the mode specific covariance matrices are then able to capture: 1) the within-parcel autocovariances which in the Fourier domain corresponds to the power spectra for each parcel, and 2) the between-parcel cross-covariances which in the Fourier domain corresponds to the cross-spectra or coherence between parcels.

With single-channel DyNeMo, we can think about taking a similar approach, but because we have single channel data, we are limited to finding modes that capture the within-parcel autocovariances that correspond to the power spectra for each parcel. Any cross-parcel connectivity information will need to be assessed in the post-hoc analysis. As mentioned above, we propose to modify this, such that we can focus on finding modes that are defined as having distinct autocorrelation functions, which in turn correspond to distinct power spectral shapes/profiles.

We can find modes with distinct autocorrelation functions by imposing the constraint that the elements on the main diagonal of the single-channel mode-specific covariances are set to one. This is because the main diagonals are the autocorrelations at zero lag, which must have a value of one to be a valid autocorrelation function.

We can also consider applying constraints to the elements on the off-diagonal mode-specific covariance matrices that are motivated by the fact that we are modelling covariances/correlations of time-delay embedded data. Note that because of time-delay embedding the off-diagonal elements correspond to the auto-correlations of each mode at non-zero lags.

Consider the matrix element C_{ij}. This is the correlation between lag i and lag j. Now, consider the matrix element $C_{i+1,j+1}$. Both time series have been shifted by one lag, leaving the correlation unchanged. From this it can be concluded that

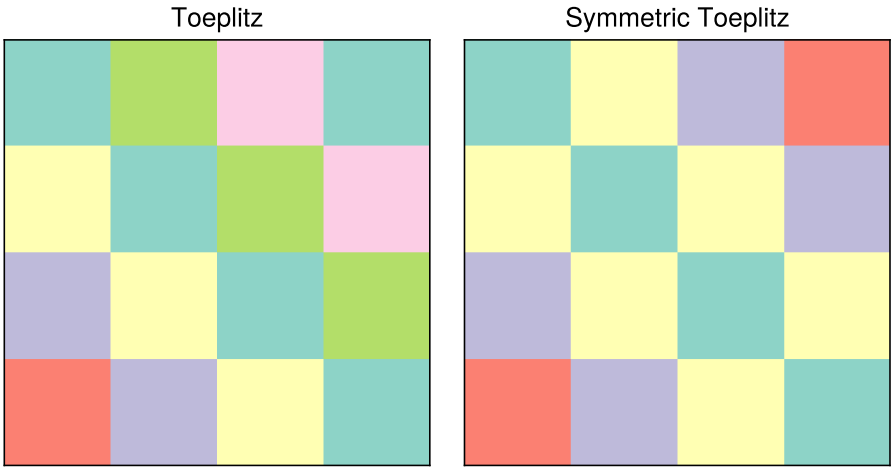


Figure 6.1: Toeplitz matrices have constant values along their diagonals. In these figures, each value is represented by a colour. A symmetric Toeplitz matrix has the additional constraint that the matrix is equal to its transpose. This means that each diagonal is also equal to the corresponding diagonal in the transpose of the matrix.

only the difference between the indices of the lags matters. As such the correlation matrix can be written as a function of the difference between the lag indices:

$$C_{ij} = c_{i-j} \tag{6.1}$$

This matrix has the same value for all elements on the same diagonal, and is known as a Toeplitz matrix, an example is shown in Figure 6.1. As correlation is a symmetric operation, the correlation between two lags is the same as the correlation between the negative of those lags. As such, the elements can be written as a function of the absolute difference between the lags:

$$C_{ij} = c_{|i-j|}. \tag{6.2}$$

A symmetric Toeplitz matrix, as shown in Figure 6.1. It can now be seen that the correlation matrix is fully defined by the values of c_k for $k \in \{0, 1, \ldots, n\}$ where n is the number of lags.

In summary, we intend to constrain the time-delay embedded, single channel, mode-specific covariance matrices to be Toeplitz matrices with diagonals set to one. This will constrain the covariance matrices to have the structure expected

of time-delay embedded data and give modes with distinct autocorrelations (and thereby distinct power spectral shapes/profiles).

Applying such constraints will help regularise the covariance matrices to have sensible autocorrelation functions even when SNR is low and/or there is not much data.

6.2.4 Implementation of Covariance Matrix Constraints

During development, it was found that producing a symmetric Toeplitz matrix which fulfilled the additional constraint of being positive semi-definite was difficult. A symmetric Toeplitz constraint can be implemented by reducing the number of free parameters in the covariance matrix. This can be done by only allowing the first row to be free. The first column can then be calculated from the first row and the symmetry constraint. The remaining elements can then be calculated from these values. There are methods for correcting matrices which are not positive semi-definite, for example by setting all negative eigenvalues to zero. However, these methods are not guaranteed to produce a Toeplitz matrix which defeats the purpose of the constraint.

To side-step this issue, it is relied upon that DyNeMo sometimes finds a near-Toeplitz solution and brute-force is applied to get the rest of the way to an appropriate solution. It is not guaranteed that the model with the best free-energy will be the one with the most Toeplitz-like covariance matrices, but there is a correlation to a degree. Selecting the model with the most Toeplitz-like correlation matrices is not at odds with choosing low free-energy models. While this is not mathematically rigorous and satisfying, it is a practical solution to the problem.

Measuring the Toeplitz-ness of a matrix can be accomplished by calculating the standard deviation across each symmetric diagonal as demonstrated in Figure 6.2. The mean of these standard deviations is then a measure of how close the matrix is to being Toeplitz. The model with the lowest mean standard deviation can then

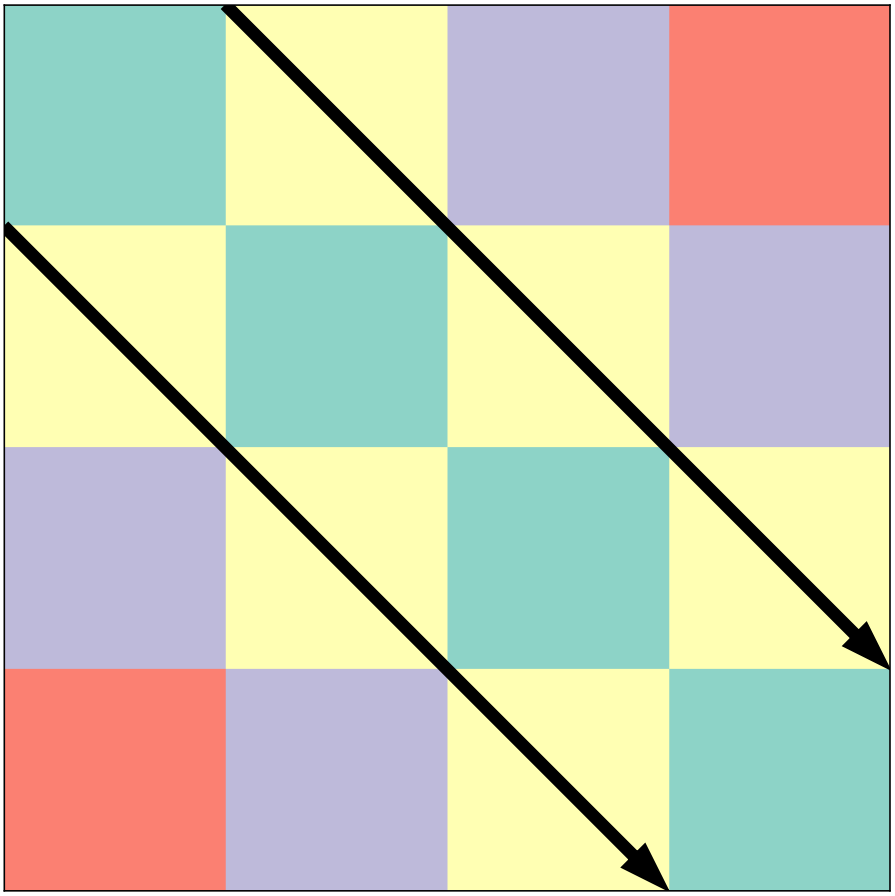


Figure 6.2: For any matrix, the symmetric Toeplitz deviation can be calculated. This figure illustrates the calculation for the diagonal $k = 1$. The standard deviation is calculated across the symmetric diagonals and then normalised by the number of elements in the combined rows. Adding the deviation from each pair of diagonals together provides the total symmetric Toeplitz deviation for the matrix.

be selected as the most Toeplitz-like. The Toeplitz deviation can be written as:

$$\sigma_T = \frac{1}{n}\sum_{k=0}^{n}\sqrt{\frac{1}{n-k}\sum_{i=0}^{n-k}\left(C_{i,i+k} - C_{i+1,i+k+1}\right)^2}. \tag{6.3}$$

The $\frac{1}{n-k}$ term is a normalisation factor which ensures that the standard deviation is not biased by the number of elements in the diagonal.

6.2.5 Multi-channel DyNeMo

For the sake of comparison, a multi-channel DyNeMo model was trained on the task data described in Section 5.2 as prepared in Section 6.2.1. 51 subjects were

used, with each subject's data containing 78 parcels with 23 embeddings. The 1794 channels were reduced to 320 using PCA. The model was trained with 6 modes.

6.2.6 Single DyNeMo

For all 78 parcels, 20 single-channel DyNeMo models were trained ($78 \times 20 = 1560$ in total). Each model was trained with 6 modes. Inferred alphas, correlation matrices, final free energies and PSDs were stored for further analysis.

6.2.6.1 DyNeMo Run Selection

DyNeMo is a stochastic algorithm (as is the HMM, ICA etc) with inherent run-to-run variability. As such, it is typical to fit multiple models and select the "best" one based on some criterion. Three different approaches were considered for selecting which of the DyNeMo runs to use for each parcel's DyNeMo results.

Firstly, the simplest metric is to use free energy. This is a natural quantity to use, as it is the value which is minimised as DyNeMo trains and is therefore a definitional marker of the quality of the fit. When this selection method is used, the model with the lowest free energy for a given parcel is kept while the others are discarded.

Secondly, we can use the metric we call symmetric Toeplitz deviation (sTD), as detailed above. This quantity defines how well the covariances of a model describe a valid auto-correlation matrix. When this selection method is used, the model with the lowest sum (over modes) sTD is kept while the others are discarded.

Finally, perhaps the most principled approach to take is to combine the first two. We can set a limit for tolerable Toeplitz-ness across all parcels and then select the best trained models by free energy under that limit. For the limit, the minimum sTD is found for each parcel. The maximum of these is determined to be the upper bound of acceptable sTD. For each parcel, models above this threshold are discarded and the models with the lowest free energies are selected.

6.2.6.2 Post-hoc Analysis: Meta-modes

We now consider how to combine the output from multiple, single-channel Dynemo fits using post-hoc analysis. We focus here on methods that look to group the decoupled parcel modes into aggregated whole-brain modes, which we refer to as meta-modes.

To perform this process rigorously, an objective function should be defined which determines the desirability of the result of the aggregation process. The objective function can be optimised to find the most desirable configuration according to the given metric. For different objective functions, this can lead to vastly different outcomes. Five objective functions were tested here and their relative strengths examined. A graphical representation of the process is shown in Figure 6.3.

6.2.6.3 Permutation Framework

To analyse aggregated meta-modes, we need to have a framework which can convert between the single parcel model outputs and derived meta-mode outputs. One way to achieve this is through permutations of the mode order for each parcel's modes. The simplest example of this is to take two parcels, each with three modes. The order of the modes for each parcel can be changed. We want to find a way to group the modes together so that the first, second and third modes from each parcel "match" in some way. To solidify this, we can consider the modes to be labelled with letters. The modes for each parcel are labelled A, B and C. DyNeMo will output its inferred modes in an arbitrary order, for example B, C, A for the first parcel and C, A, B for the second. We can then permute the order of the modes for each parcel to match the desired grouping. The objective is to find a permutation of the modes for each parcel which will optimize some function of the resulting meta-modes.

6.2.6.4 Simulated Annealing

In Chapter 3, simulated annealing was mentioned. Here it is used for a different purpose - as a tool for global optimisation. The process of finding a global optimum for an objective function using the permutation framework described above is an

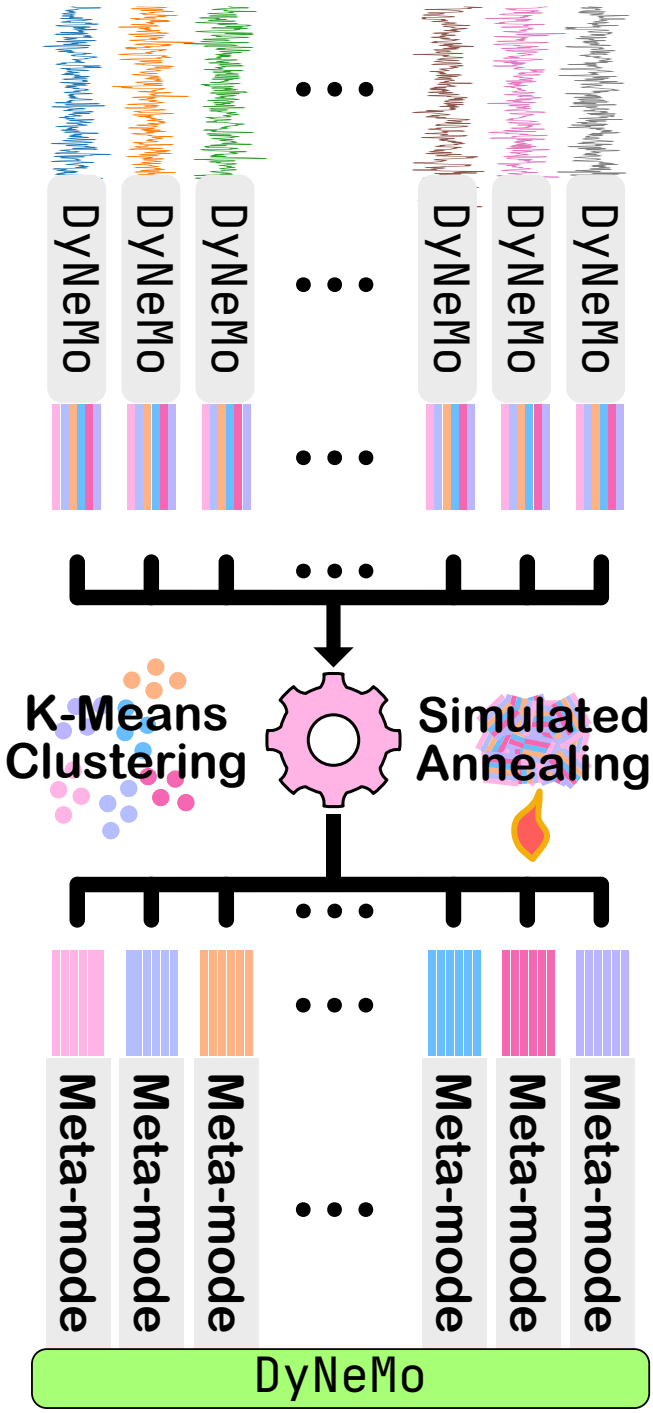


Figure 6.3: A graphical representation of the process of finding meta-modes from single-channel DyNeMo results. From the top down: Data from each parcel is used to train a single-channel DyNeMo model. Each model produces a set of parcel-modes. The parcel-modes are then grouped into meta-modes using an objective function. The meta-modes are then used to generate a new set of alpha time courses. The analysis can then continue using the meta-mode alphas as if they were a multi-channel DyNeMo model. All other parameters can be calculated by regression against the data.

example of a multidimensional assignment problem. Often, finding the name of a problem is half the work of finding an optimum solution, but in this case it reveals that the problem is NP-hard (where NP means nondeterministic polynomial time). This is a mathematical categorisation of problems which means that no algorithm has been discovered which can solve it deterministically in polynomial time. Here polynomial time means that an upper bound of the time taken for the algorithm to execute is defined by a polynomial of the size of the inputs, for example matrix multiplication scales with at most the third power of the size of the input. Not only is there no known algorithm for solving NP-hard problems in polynomial time, this also applies to verifying solutions. This means that even if a seemingly good solution is found, it is difficult to be certain that it is the best solution.

The term simulated annealing was coined in 1983[93] to describe an approach to solving problems of combinatorial optimization by analogy to statistical mechanics and metallurgy. In metallurgy, annealing is the process of heating and then gradually cooling a metal. The gradual reduction in temperature allows large grains to form, improving the workability of the metal. This works because the atoms in the crystal lattice are given time to migrate to locations which minimise the energy of the system while at high temperatures, before becoming less mobile and eventually fixed at lower temperatures.

In our context, the solution space is the arrangement of the modes within each channel. A neighbourhood is defined which describes the solutions which are proximate enough to be accessible from the current solution by a small perturbation. Each solution has a score which can be evaluated. A neighbour is chosen at random and evaluated. In a simple hill-climbing algorithm, a better solution would always be selected and a worse one rejected. However, in simulated annealing, the probability of acceptance is defined by a distribution dictated by the current "temperature" of the system. When the temperature is high, lower scoring solutions are more likely to be accepted allowing a broader range of the solution space to be explored. As the temperature decreases, the likelihood of accepting a worse solution decreases, and the algorithm focuses on optimising in the best neighbourhood it

has located. In the original paper, the travelling salesman problem (another combinatorial optimisation problem) is used to demonstrate the efficacy of the annealing method. This is also a many-dimensional discrete search space.

In their 1995 paper,[94] Tian et al. show that simulated annealing is a credible method for solving combinatorial problems in integer programming. Although it is not guaranteed that a global minimum will be found, simulated annealing offers good coverage of an enormous solution space.

In this experiment, the neighbourhood N of a solution S was defined to be the set of solutions which are equal to S with a permutation applied to the modes of a single channel. Several functions were used to illustrate the flexibility of the method. These have previously been referred to as cost functions, energies and scores, but from here on will be called objective functions. The task of the algorithm is to optimise the solution to the problem with respect to the objective function.

6.2.6.5 Alpha Correlation

The first objective function is intended to find meta-modes that group single-parcel-modes together that activate "coherently". This is achieved using inter-mode alpha correlation. To this end, the alpha time courses are reshaped from [modes $\times$ parcels $\times$ time] to [modes $\cdot$ parcels $\times$ time]. A correlation matrix with dimensions [modes $\cdot$ parcels $\times$ modes $\cdot$ parcels] is then computed between the mode alphas. Permuting the mode order for each parcel in this matrix allows for the matrix to be split into sub-matrices reflecting the correlation between parcels for the same meta-mode.

Two related objective functions will be tested at this stage. The first is a sum over each alpha correlation matrix followed by a mean over the sums. This objective function evaluates the alpha correlation between parcels within each meta-mode. The second is the mean of the sum of the squared elements of the correlation matrices. In contrast to the first objective function, this method is able to find meta-modes comprising both correlated and anti-correlated parcel-modes. The function

has no preference between these and simply evaluates the magnitude of the correlation. This is a departure from multi-channel DyNeMo which defines all parcels to have equal alphas for a given mode, representing them with a single time course.

6.2.6.6 Dummy Variables

It is possible to insert "dummy" entries into the alpha correlation matrix which provide the search algorithm with greater flexibility when matching modes. Dummy entries should not contribute to the objective function, but rather should allow a meta-mode to exclude a parcel from its entries. The correct choice for a dummy entry in the alpha correlation matrix is zero so that the algorithm has no incentive to include it. Adding one dummy entry to each parcel allows an extra meta-mode to be formed. Additional dummy entries allow for further meta-modes to be added. The condition of permutability could also be lifted. Instead, the algorithm could be allowed to select from any of the available modes, ignoring the exclusivity criterion of the permutation framework. This would allow not just for meta-modes to not include a given parcel, but for a parcel-mode to not contribute to any meta-modes (or to contribute to more than one). This relaxation of the problem space increases the flexibility of the search, but also has implications for the validity of the alpha time courses as, when a parcel-mode is unassigned, its explanatory power is lost from that parcel, and consequently the meta-model.

6.2.6.7 Frequency Clustering

Finding modes based on autocorrelation allows us to isolate the frequency component of the data from the variance. We can take advantage of this by creating meta-modes which reflect data-driven frequency bands. The aim is to have a series of meta-modes which have their own distinct frequency range and which can be used to demonstrate the spectral characteristics of different regions of the brain. For example, in the task we expect to see a reduction in beta power in the motor cortex caused by the event-related desynchronisation of beta band activity when the participant performs an abduction. We also expect to see alpha activity in the

visual cortex. Creating the meta-modes in this way could help to remove extraneous influences which cloud the results from this angle in multi-channel DyNeMo.

The objective function we use is a squared Euclidean distance metric over the PSDs of each parcel-mode. The PSDs can be calculated either as a regression over alphas and the time-frequency content of the single parcel's input data, or as a Fourier transform of the autocorrelation function defined by the correlation matrices inferred by DyNeMo. Either way, a set of N representative PSDs will be determined across the whole parcel-mode space. To achieve this, the PSDs will be treated as vectors with n_{freq} dimensions and a K-means clustering algorithm will be applied using the squared Euclidean distance metric. Maximising the correlation of the spectral profiles would also be a valid method for finding these clusters, but in a reasonably high dimensional space, the squared Euclidean distance is less affected by the magnitude of individual time points and more by the presence of some activation in a given dimension. The KMeans algorithm is faster to converge than the simulated annealing approach and is well defined using the squared Euclidean distance, thus it is chosen here.

With the representative PSD centroids found, each of the parcel-modes will be labelled by its closest centroid. No requirement for mutual exclusivity is imposed and so multiple modes from the same parcel could be assigned to the same centroid and therefore intuitively a given parcel could have no modes representing a given centroid. For each centroid, if a parcel is represented by more than one mode, the alpha time courses of those modes will be summed to find a combined activation. If no modes represent the centroid, its alpha time course for that parcel will be set to zero. This is directly equivalent to the allowance of dummy states in the previous section except that it is not required to add additional metamodes to contain the new content.

6.2.6.8 Post-hoc Trial Matching

The methods above seek to avoid polluting the analysis of the data with knowledge of the task. If this notion is discarded, there are more options available. The

analysis of multi-channel data shows strong correlation between the task structure and the alphas. It is therefore reasonable to use the alphas from a window around the task to match the single-channel alphas to each other.

We expected that the alpha time-courses of some modes from some parcels will align to show some underlying activity from the data. Each parcel-mode will be defined to be a vector in a space with $n_{\text{timepoints}}$ dimensions. There are many algorithms to find clusters of similar activity. For simplicity and speed, the K-Means algorithm was chosen. The choice of the number of centroids provides the freedom to select a number of modes which can be larger (or smaller) than the number of modes per parcel. This means that the solution space moves from

$$[n_{\text{parcels}} \times n_{\text{modes}} \times n_{\text{timepoints}}]$$

to

$$[n_{\text{parcels}} \times n_{\text{centroids}} \times n_{\text{timepoints}}].$$

As with the previous methods, once an ordering criterion had been established, a new set of alpha time courses can be generated with which to continue the analysis.

6.3 Results

6.3.1 Multi-Channel DyNeMo

A multi-channel DyNeMo model was trained on the same dataset as the multiple, single-channel models. The model was trained with the same parameters as the multiple, single-channel models, except that the number of modes was reduced to six to match the number of modes in the single-channel models. The power maps, alphas and PSDs of the multi-channel DyNeMo modes are shown in Figures 6.4 to 6.6. Typical DyNeMo modes (including motor and visual modes) are seen in the power maps, with the background mode in mode 5.

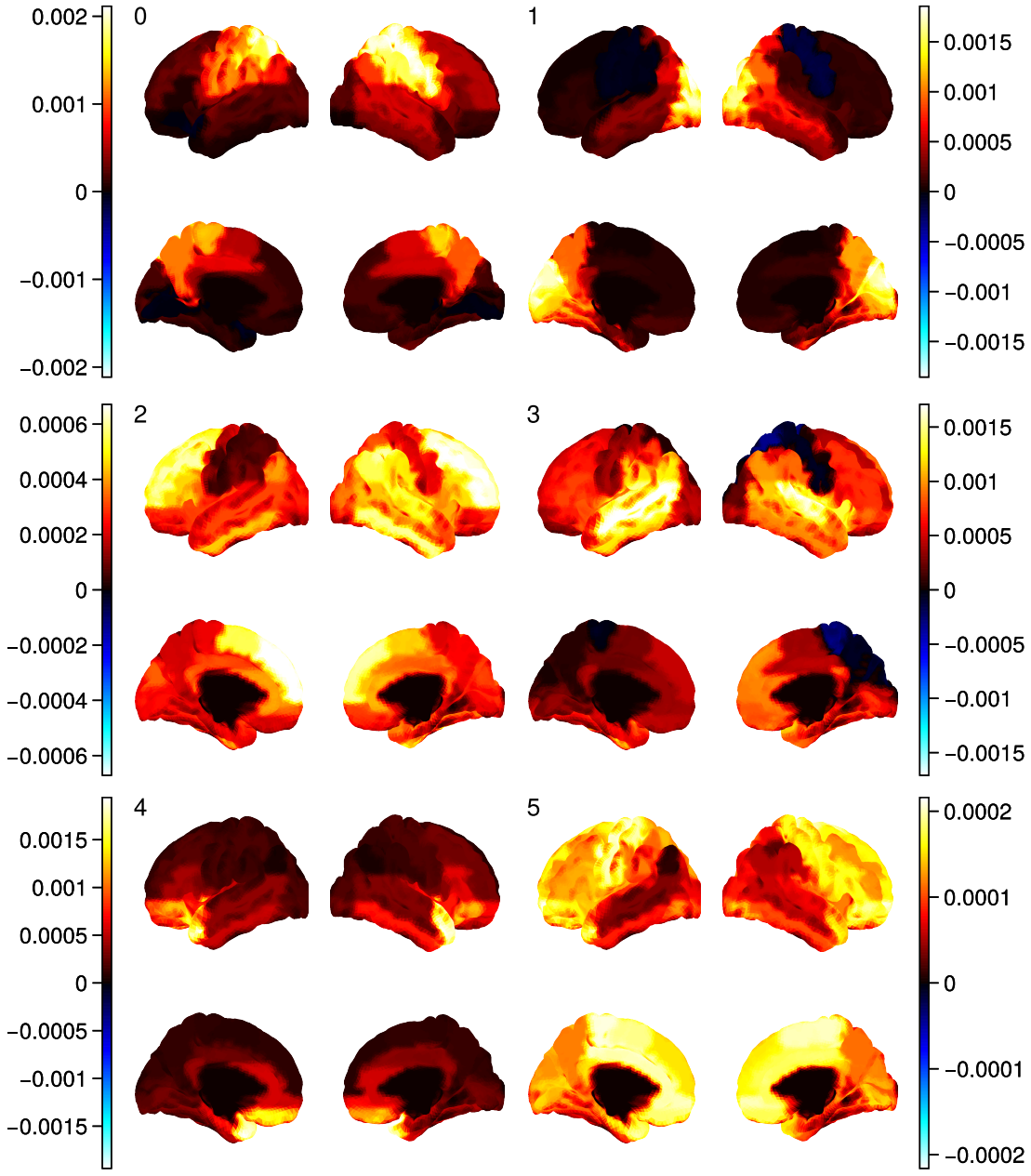


Figure 6.4: Multi-channel DyNeMo modes from a model trained on the MEG task data used in the multi-channel analysis chapter, but with the preprocessing specified for the single-channel chapter. Surface power maps show the projection of power from each parcel onto the surface of the brain for frequencies in the range 1 to 45 Hz. They reveal a typical DyNeMo background mode in mode 5. Only six modes were trained to match the number of modes in the single-channel models.

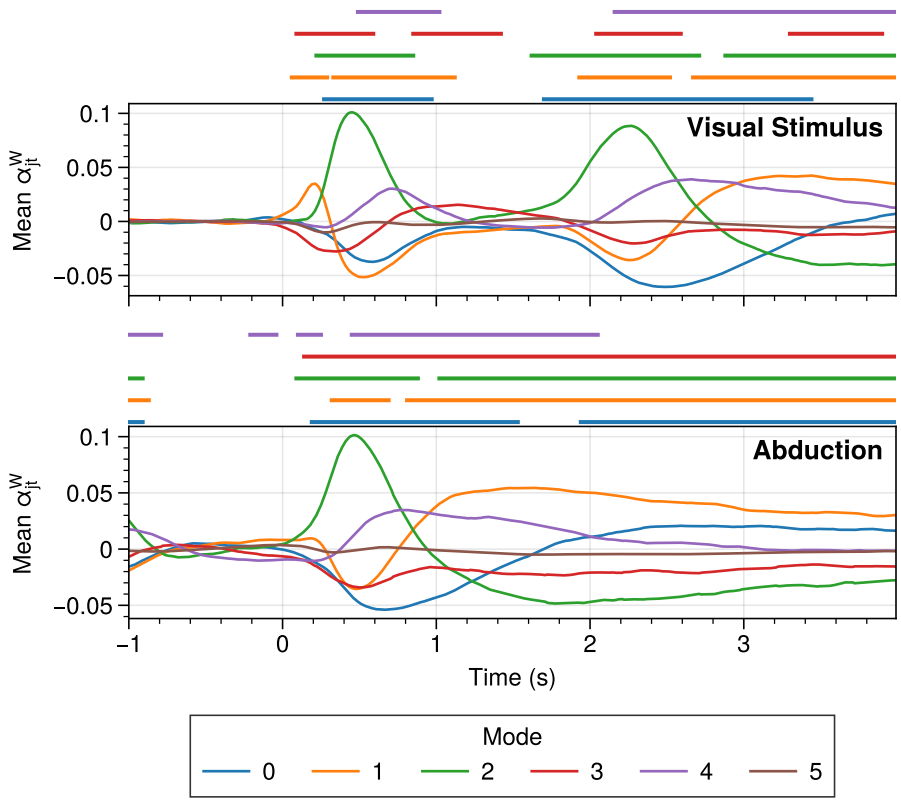


Figure 6.5: The activation of the multi-channel DyNeMo modes trained on the multiple, single DyNeMo data during the visual stimulus and abduction windows. Strong task dependence is shown. The coloured lines above each plot indicate the statistical significance of the task dependence. When the lines are present, the relevant mode has a p-value of less than 0.05 according to a sign-flip permutation test.

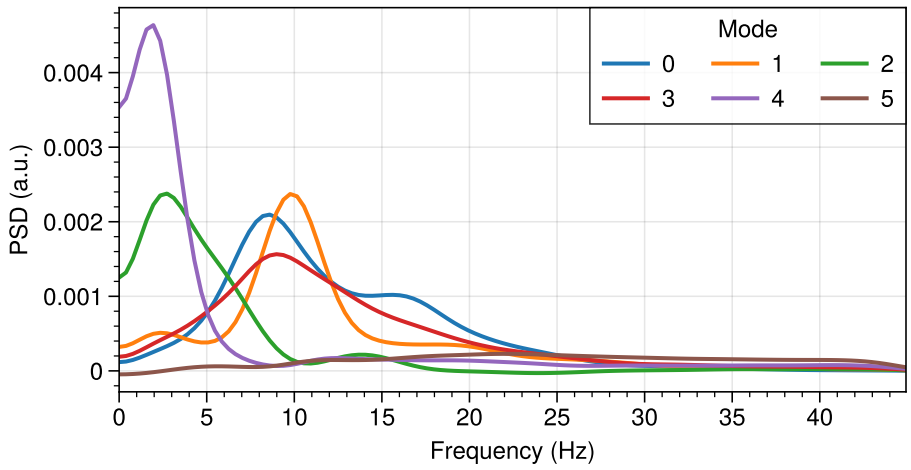


Figure 6.6: The PSD of each multi-channel DyNeMo mode, trained on the same dataset as the multiple, single DyNeMo models is plotted. As with the model from Chapter 5, peaks can be seen in the delta, alpha and beta bands.

6.3.2 Single DyNeMo

6.3.2.1 Run Selection

For each model, the free energy and summed (over modes) sTD was calculated. Subtracting the channelwise mean free energy and sTD from each model reveals a positive correlation between them. There is an outlying cluster with low free energy but non-Toeplitz structures. For the MinMax selection approach, the selection is performed on the sTD without demeaning as the aim is to establish an absolute bound on the sTD rather than a relative, per-channel one. The results of each selection method are shown in Figure 6.7. To illustrate the variability in the sTD metric over all inferred modes, each correlation matrix for all models is plotted in Figure 6.8, where they are sorted by their sTD. From here on, the MinMax approach is used in all analyses.

6.3.2.2 Mode-specific Autocovariance Functions

Single-channel DyNeMo can learn mode-specific autocovariance functions. We will walk through one run of one channel to demonstrate this.

Firstly, we will look at the correlation matrices learned by the model. In Figure 6.9, we see that have near-Toeplitz structure. As the first row of each matrix is the autocovariance function, we can apply a Fourier transform to the first row of each matrix to obtain the PSD of each mode. In Figure 6.10, we see that each mode has a distinct autocorrelation function, and therefore a distinct PSD. Forcing DyNeMo to learn autocorrelation-like matrices has allowed it to learn mode-specific PSDs.

6.3.2.3 Permutation

The correlation between modes was calculated for every mode in every channel for each run selected. This created a global correlation matrix shown in Figure 6.11a. Using the permutation framework, these correlations were split into six smaller matrices representing the correlation between the alphas of each channel in the assembled metamodes. A random initialisation is shown in Figure 6.11b. The

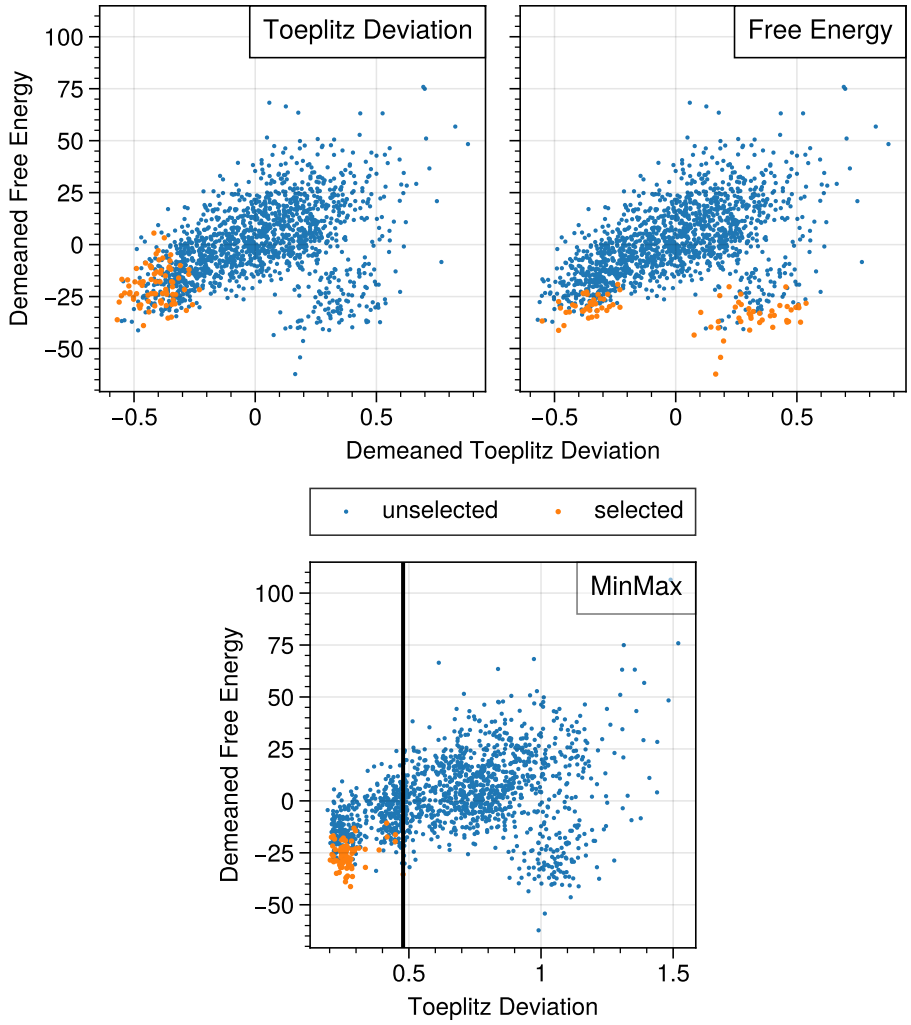


Figure 6.7: Three methods for selection of the most appropriate single-channel DyNeMo run for each channel are presented. In the top left figure, the model with the lowest symmetric Toeplitz deviation for each channel is selected. This disregards the value of free energy in determining a good model. In the top right figure, the mode with the lowest free energy for each channel is selected. This disregards the value of the Toeplitz deviation in selecting valid autocorrelation matrices. In the bottom, the minimum deviation is found for each channel. The largest of these is established as a barrier under which all deviations are acceptable. From the remaining runs, those with the lowest free energy are chosen. The black line indicates the limit of acceptable deviation. In the first two plots, the channelwise mean free energy and symmetric Toeplitz deviation are subtracted from each run. This removes the effect of channel data variability to show the relation between free energy and Toeplitz deviation without this confound. The MinMax method is shown with the symmetric Toeplitz deviation without demeaning, as the absolute value is the important quantity. Although there are many outliers, a positive correlation can be seen between the free energy and Toeplitz deviation.

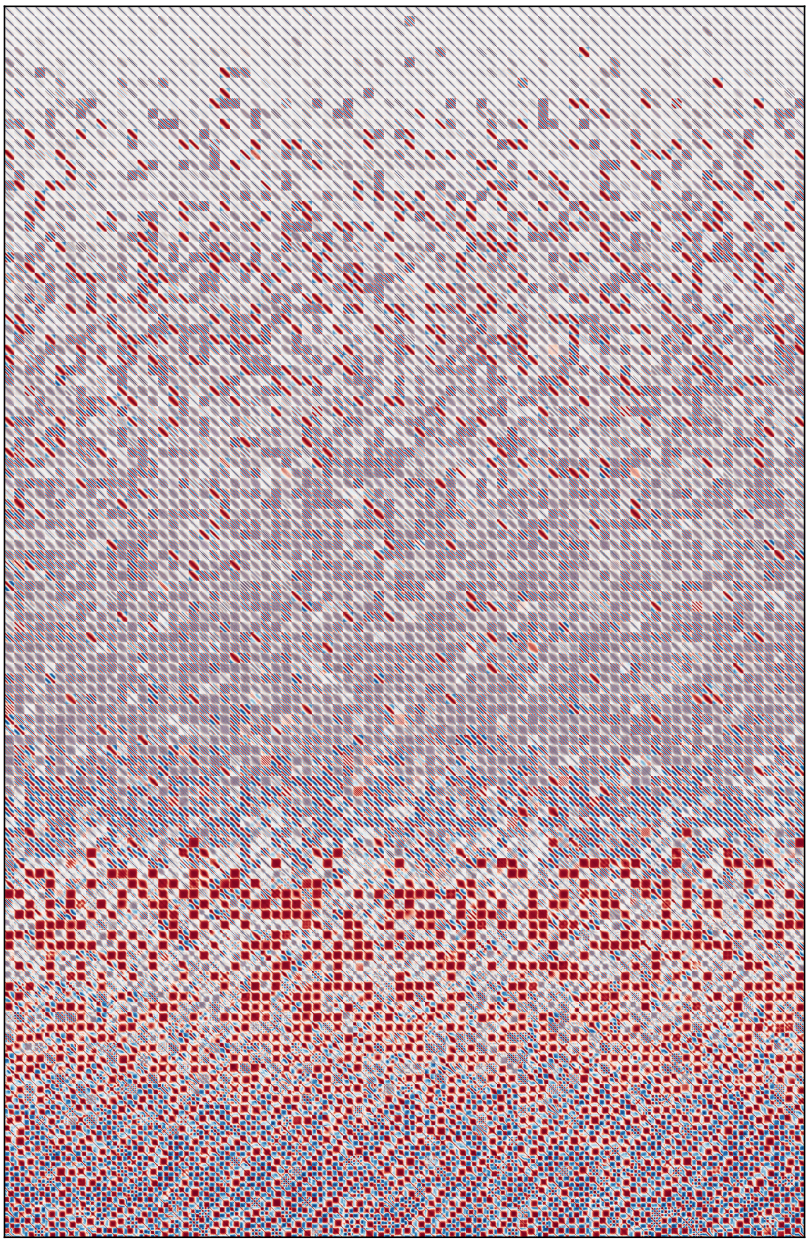


Figure 6.8: Selecting the "best" DyNeMo run using similarity to a symmetric Toeplitz matrix. Results correspond to 78 channels, 6 modes and 20 runs; producing 9360 correlation matrices. These are shown sorted by how much they deviate from the structure of a symmetric Toeplitz matrix.. The "most-Toeplitz-like" matrices are at the top, with progressively more deviant matrices further down. Matrices that are most-toeplitz-like will correspond more closely to be modes with valid autocorrelation functions and power spectral shapes/profiles.

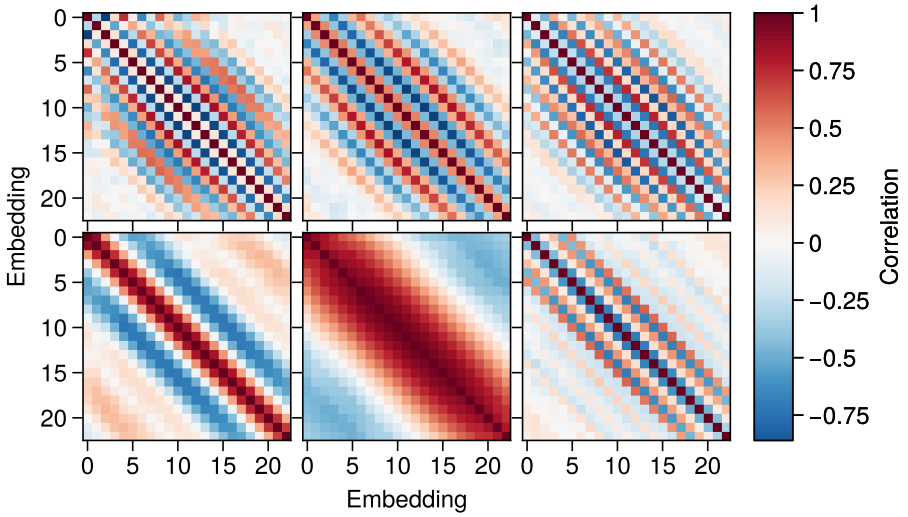


Figure 6.9: Correlation matrices learned by the single-channel DyNeMo model trained on the MEG task data used in the single-channel analysis chapter. These matrices are the correlation matrices for one run of one channel. The first row of each matrix is the autocovariance function of the corresponding mode. The matrices have near-Toeplitz structure.

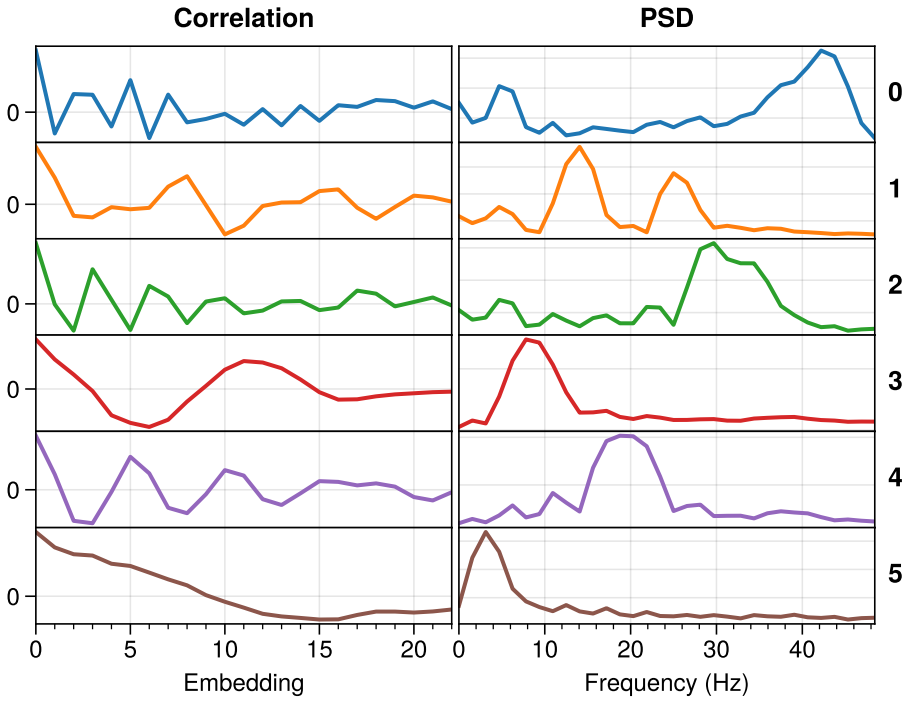


Figure 6.10: The left column shows the autocorrelation functions for each mode. The right column shows the PSD of each mode. The PSD is calculated by taking the Fourier transform of the autocorrelation function. Each mode has a distinct autocorrelation function, and therefore a distinct PSD.

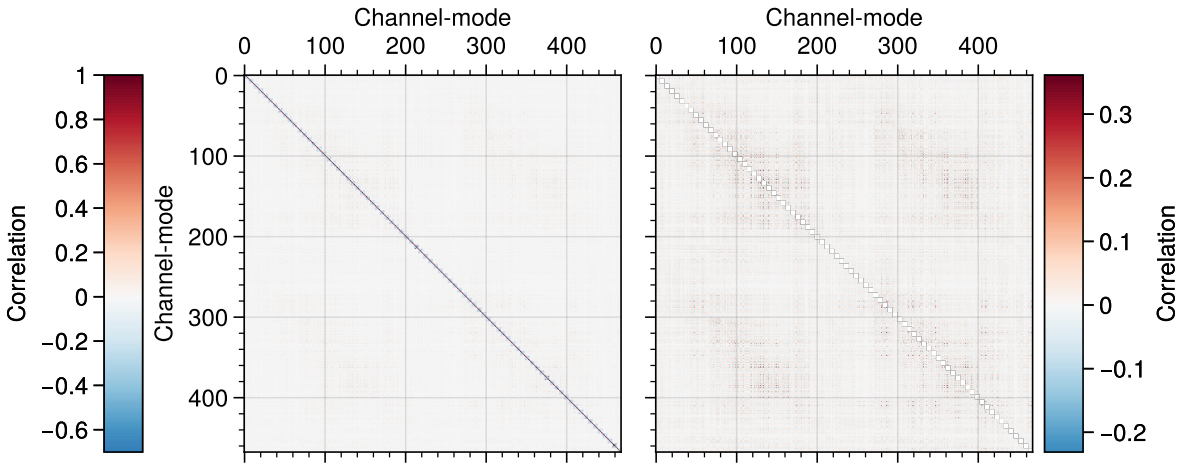
simulated annealing method was applied to the global covariance to find a set of metamodes which maximised an objective function.

6.3.2.4 Alpha Correlation

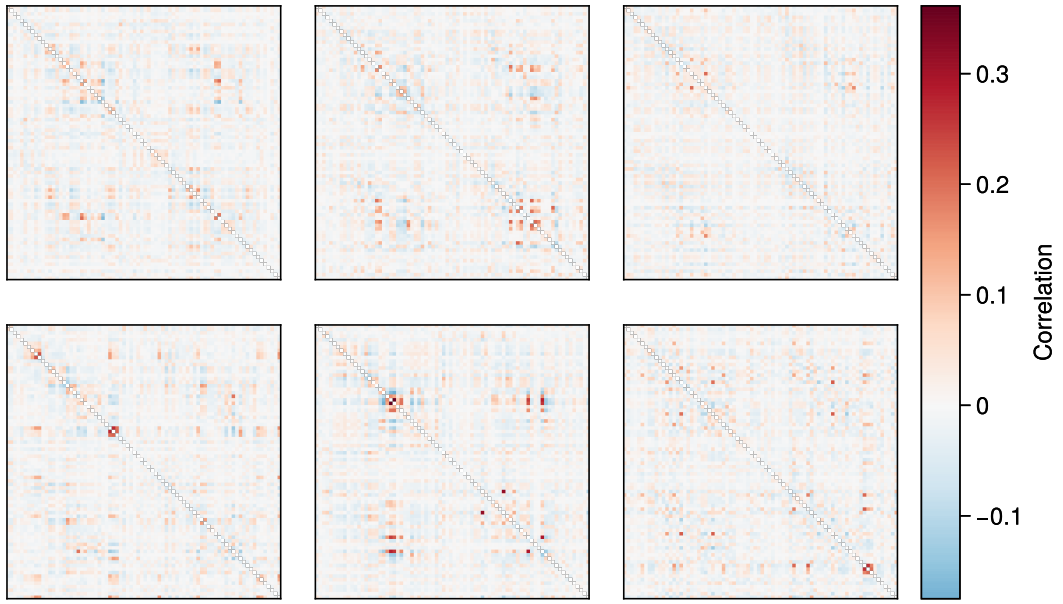
Plausible meta-modes can be identified based on correlation of parcel-mode alphas. In this analysis, the linear and square variations of the alpha correlation objective function found identical solutions. This is because it is possible to assemble matrices with larger absolute sums using exclusively the positive correlations of the modes. The alpha correlation matrices found are shown in Figure 6.12. Compared to the matrices in Figure 6.11b, the alpha correlation matrices have a much higher proportion of positive elements. This is an indication that the correlation is being maximised successfully.

Next, we consider the parcel-mode alpha groupings shown in Figure 6.13. We see that the trial-averaged alphas of each parcel-mode are sensibly grouped by the metamodes and show high consistency. Taking an additional mean over parcel-modes (shown in Figure 6.14) we can see that the metamodes show high task correlation. The coloured lines above each plot represent timepoints for each metamode where the p-value of the task correlation is less than 0.05 according to a sign-flipping permutation test.

The power maps of the metamodes are shown in Figure 6.15. The metamodes show clusters of nearby parcels activating together, even though DyNeMo is only able to train on each parcel in isolation. This shows that the metamodes are capturing consistent patterns of activation across parcels. Mode 2 shows the clearest motor cortex activation. It also shows a reduction in activation during the motor task. The PSD of the mode is shown in Figure 6.16 and shows a peak in the beta band. This displays the expected behaviour of the motor cortex during a motor task (event-related desynchronisation). The metamode PSDs show good frequency separation, with peaks at 1.95, 7.81, 10.94, 16.80, 25.78 and 41.02 Hz.



(a)



(b)

Figure 6.11: In Figure 6.11a (left), the correlation between every mode in every channel is shown. The task of the analysis is to adequately separate this matrix into a set of correlation matrices which are representative of the data. As each model has a background mode, the block diagonal elements show anticorrelation, except for the main diagonal which is equal to 1 as each mode correlates perfectly with itself. In Figure 6.11a (right), the block diagonals are removed to allow the structure in the rest of the matrix to show. Figure 6.11b shows a random selection which leads to a balance between positive and negative correlations. The goal of the methods in this section is to optimise some parameter by rearranging the elements in these matrices.

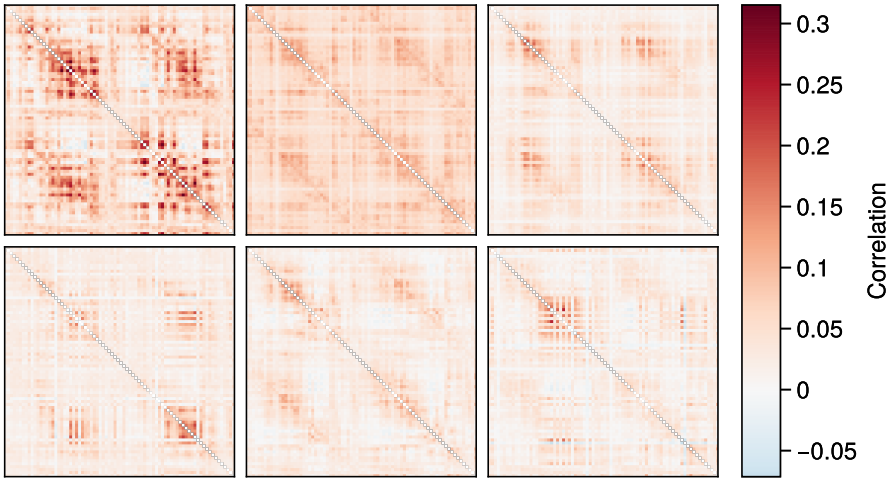


Figure 6.12: Simulated annealing was used to find a set of metamodes which maximised the total correlation between mode alphas within each metamode. Here we show the correlation of the alphas within the metamodes. Six metamodes were found with no dummy modes. The removal of negative elements is an indication that the correlation is being maximised successfully.

6.3.2.5 Dummy Variables

Adding dummy variables allows non-correlated parcel-modes to be isolated from the main meta-modes. Two dummy variables were added to each parcel. This allowed the creation of eight meta-modes rather than six. Simulated annealing was applied in the same manner as for the six meta-mode variant using parcel-mode alpha correlation. When the model was unable to find a parcel-mode alpha which positively correlated with the existing modes, a dummy mode was incorporated instead to avoid the addition of a negative correlation. This parcel-mode was then assigned to one of the dummy meta-modes. This is most noticeable in the sixth metamode of Figure 6.17 in which a large number of parcel-modes have been removed to maximise the correlation.

Considering the parcel-mode alpha groupings shown in Figure 6.18, we see that the trial-averaged alphas of each parcel-mode are sensibly grouped by the metamodes and show high consistency for the first six metamodes. The metamodes 7 and 8 show less consistency, but this is to be expected as they are primarily used to collect the modes which are most uncorrelated with the other metamodes.

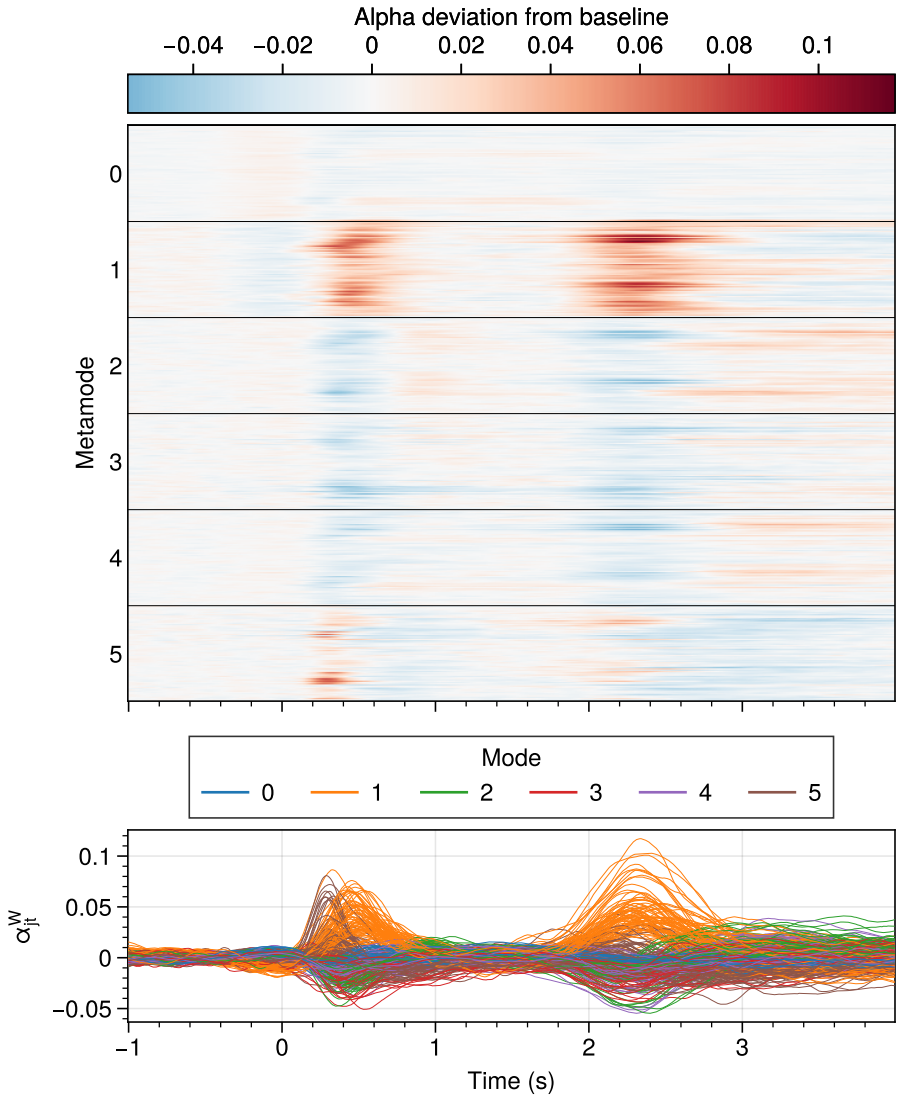


Figure 6.13: Trial averaged alphas for parcel modes. The groupings of parcel-modes into metamodes is performed by maximising the total correlation between mode alphas within each metamode. The similarity of the alphas within each metamode demonstrates that the metamodes are capturing consistent patterns of activation across parcels.

Not that although there are now eight metamodes, the number of parcel-modes in each remains the same as in the six meta-mode analysis. This is because the dummy variables are also included.

The metamode mean alphas are shown in Figure 6.19. The metamodes show high task correlation which we can see from the coloured lines indicating a sign-flipping permutation test p-value of less than 0.05. Mode 6 is not task correlated at all, and mode 7 has only around 0.2 s of task correlation around the presen-

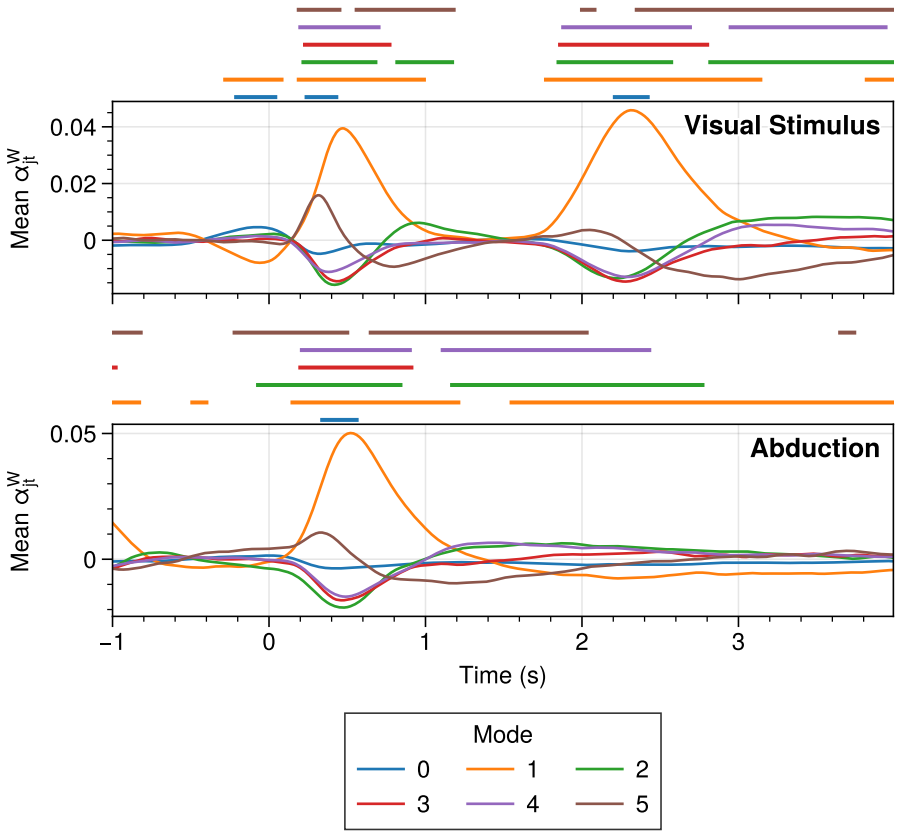


Figure 6.14: The mean of the trial averaged alphas for each metamode. The meta-modes were found by maximising the total correlation between mode alphas within each metamode. The coloured lines above each plot represent timepoints for each metamode where the p-value of the task correlation is less than 0.05 according to a sign-flipping permutation test. We see high task correlation within each metamode.

tation of the visual stimulus.

The power maps in Figure 6.20 show that modes 0, 1 and 4 remain roughly unchanged between the six and eight meta-mode analyses. Mode 2 (which matches mode 3 in the 6 metamode analysis) shows a more distributed activation in the motor cortex and mode 3 (which matches mode 2 from the metamode analysis) is more focussed in the visual cortex in the eight meta-mode analysis. Mode 5 from the six meta-mode analysis is split between modes 5, 6 and 7 in the eight meta-mode analysis.

The PSD of the metamodes are shown in Figure 6.21. Metamodes 0 - 5 show good frequency separation, with peaks at 1.95, 9.38, 16.02, 23.44, 26.56 and 41.02 Hz. Metamodes 6 and 7 have overlapping peaks which correspond to the peak

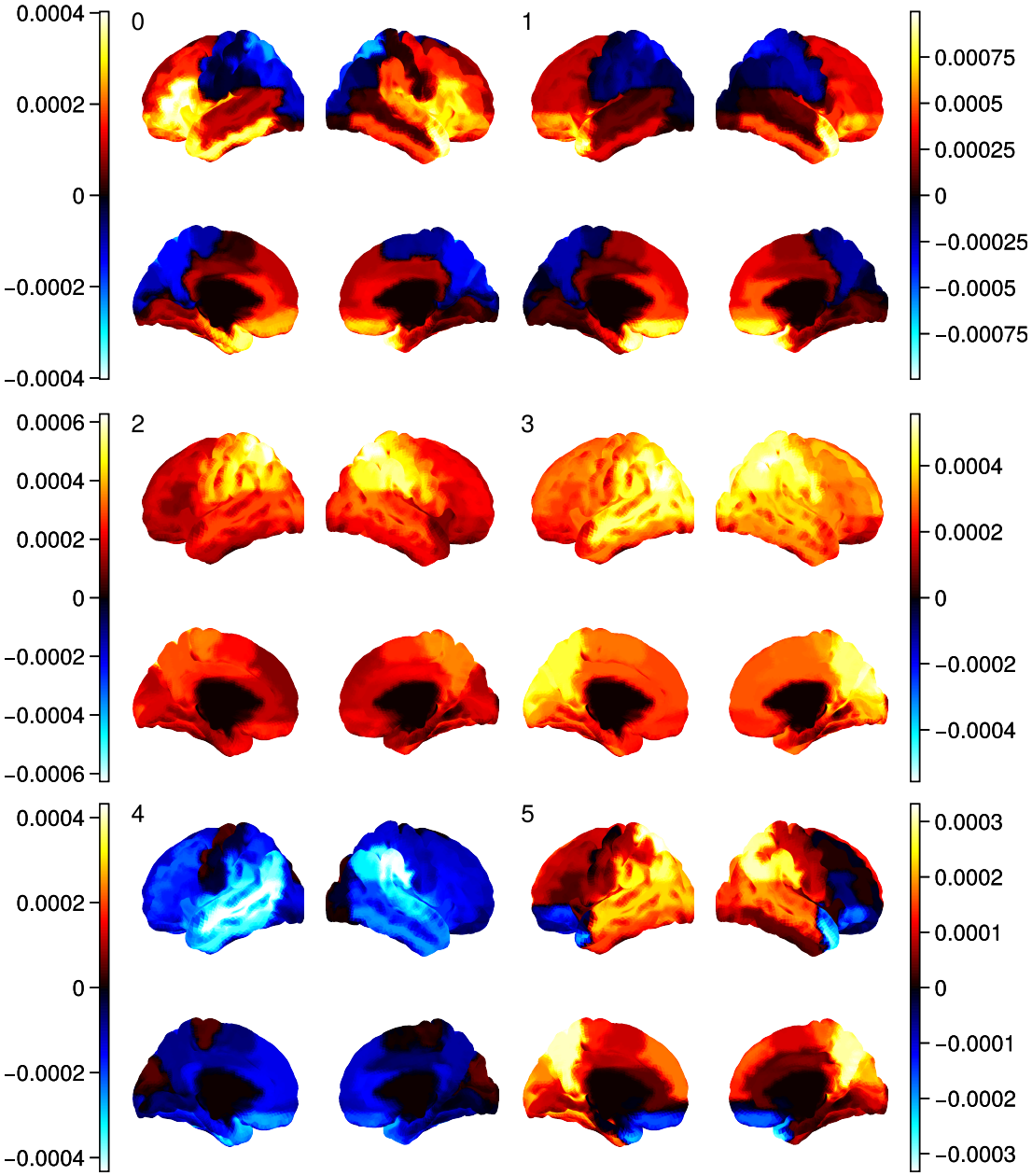


Figure 6.15: The power maps of the metamodes found by maximising the total correlation between mode alphas within each metamode. The variance of each metamode is projected onto the surface of the brain. The metamodes show clusters of nearby parcels activating together, even though DyNeMo is only able to train on each parcel in isolation.

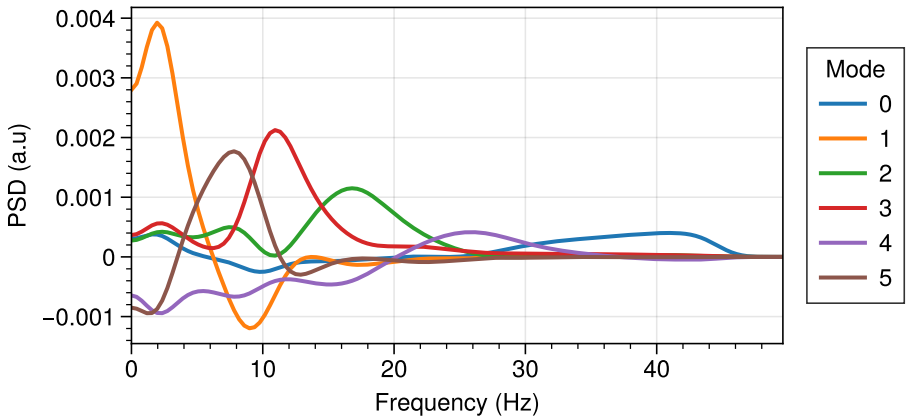


Figure 6.16: The PSD of the metamodes found by maximising the total correlation between mode alphas within each metamode. The metamodes show good frequency separation, with peaks at 1.95, 7.81, 10.94, 16.80, 25.78 and 41.02 Hz.

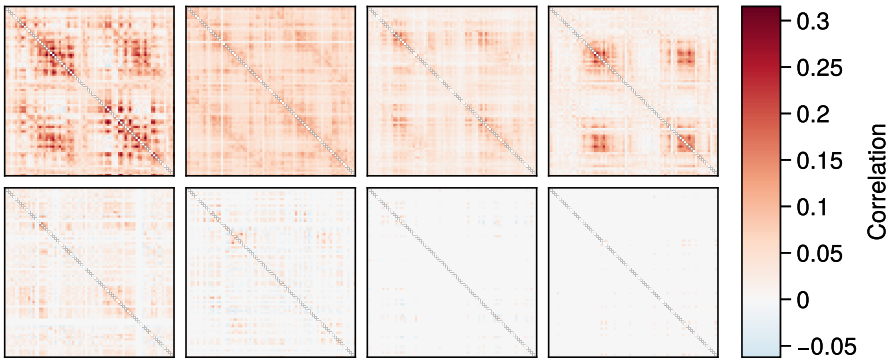


Figure 6.17: Adding dummy variables to the alpha correlation method allows solutions to be found which separate the inferred mode activity into more cohesive units. The solution space is expanded by allowing metamodes to include non-contributing terms. This also allows any modes which are not incorporated into one of the first 6 metamodes to be grouped with other more similar modes. In this figure, blank lines indicate the presence of a dummy variable. The latter metamodes are primarily used to collect the modes which are most uncorrelated with the other metamodes and are less interpretable on their own.

in mode 5 (7.81 Hz) from the six meta-mode analysis (Figure 6.16).

6.3.2.6 Frequency Clustering

Plausible meta-modes can be identified based on grouping frequency content. The frequency profile for each parcel-mode was obtained twice. Once by calculating the PSDs from the autocorrelation matrices, and once by regressing over the alphas. In both instances, following the application of the KMeans clustering algorithm, 6

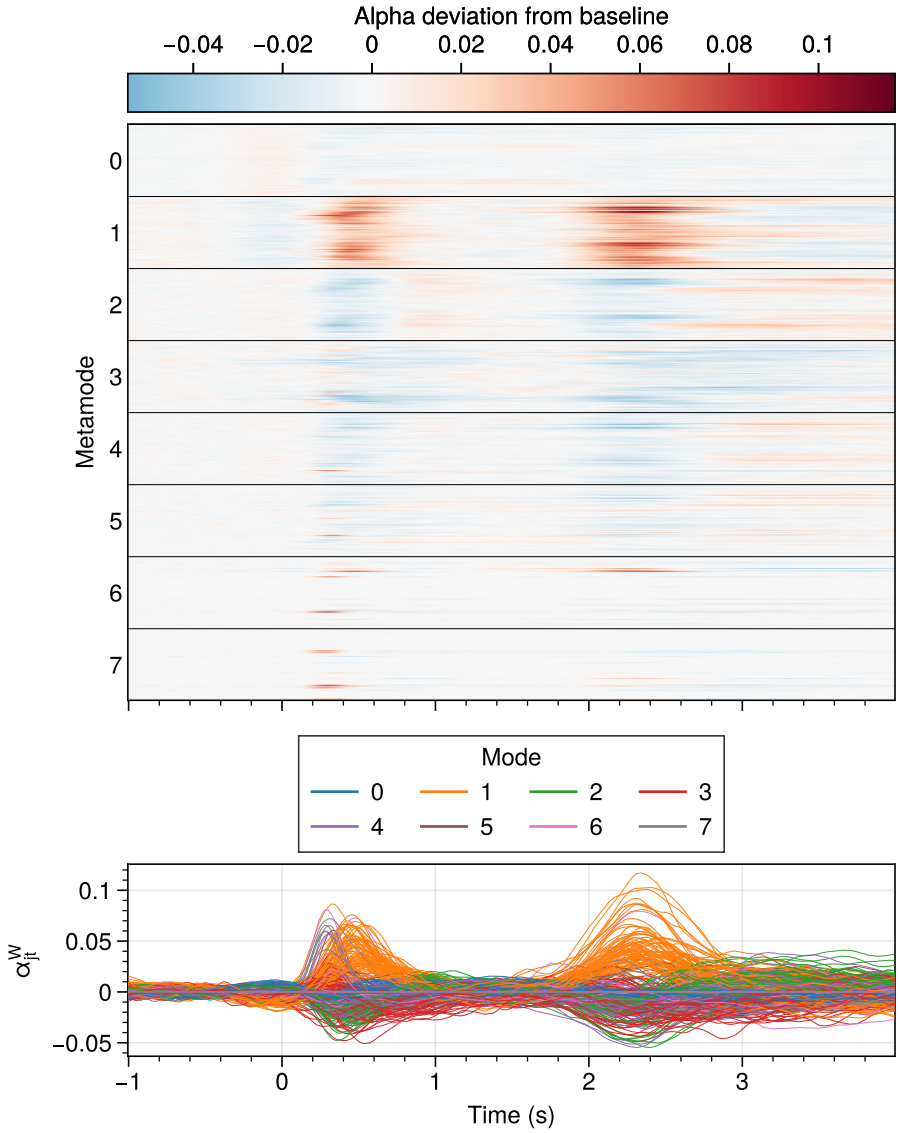


Figure 6.18: Trial averaged alphas for parcel-modes. The groupings of parcel-modes into metamodes is performed by maximising the total correlation between mode alphas within each metamode. Two dummy variables were added to each parcel. The similarity of the alphas within each metamode demonstrates that the metamodes are capturing consistent patterns of activation across parcels. The latter metamodes are primarily used to collect the modes which are most uncorrelated with the other metamodes and are less interpretable on their own.

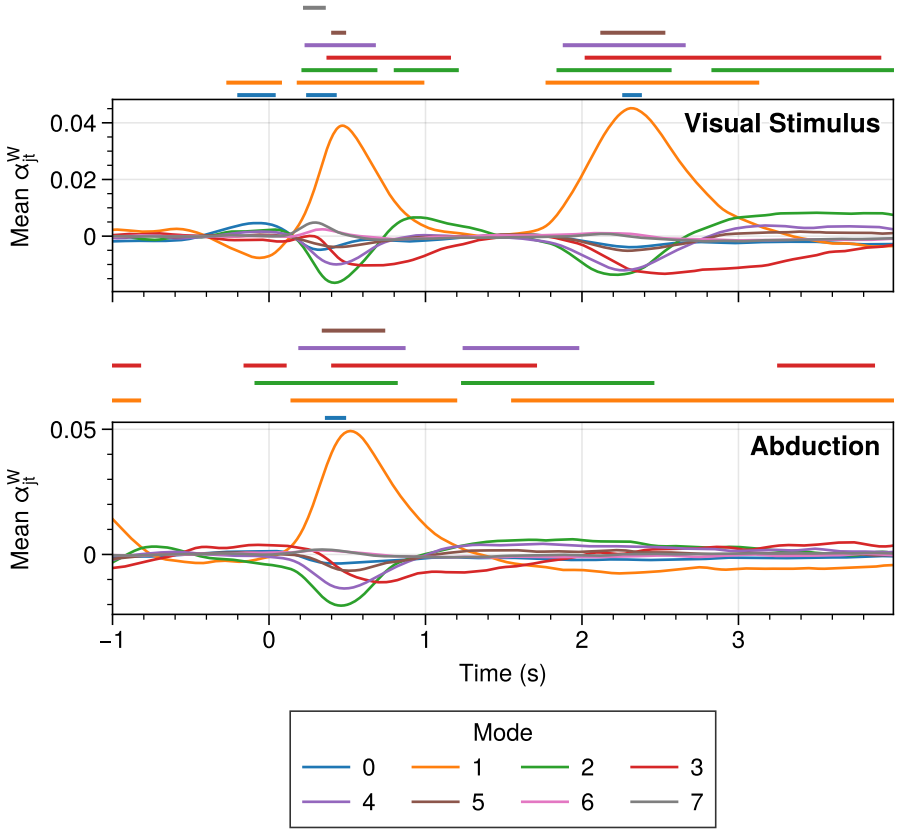


Figure 6.19: The mean of the trial averaged alphas for each metamode. The metamodes were found by maximising the total correlation between mode alphas within each metamode. Two dummy variables were added to each parcel. The coloured lines above each plot represent timepoints for each metamode where the p-value of the task correlation is less than 0.05 according to a sign-flipping permutation test. We see high task correlation within each metamode. The latter metamodes are primarily used to collect the modes which are most uncorrelated with the other metamodes and are less interpretable on their own.

meta-modes were created. These meta-modes were sorted by their peak frequency.

6.3.2.6.1 Regressed PSDs

The clusters of spectra in Figure 6.22 show that the clustering algorithm is successfully splitting the inferred parcel-modes into distinct spectral groups with peaks at 1.95, 6.64, 9.375, 10.93, 16.80 and 25.39 Hz respectively.

Grouping the parcel-mode alphas by meta-mode, we see consistent activation across trials (Figure 6.23). We also see in Figure 6.24, that each meta-mode is positively correlated with its spectral neighbours' alphas and anti-correlated with

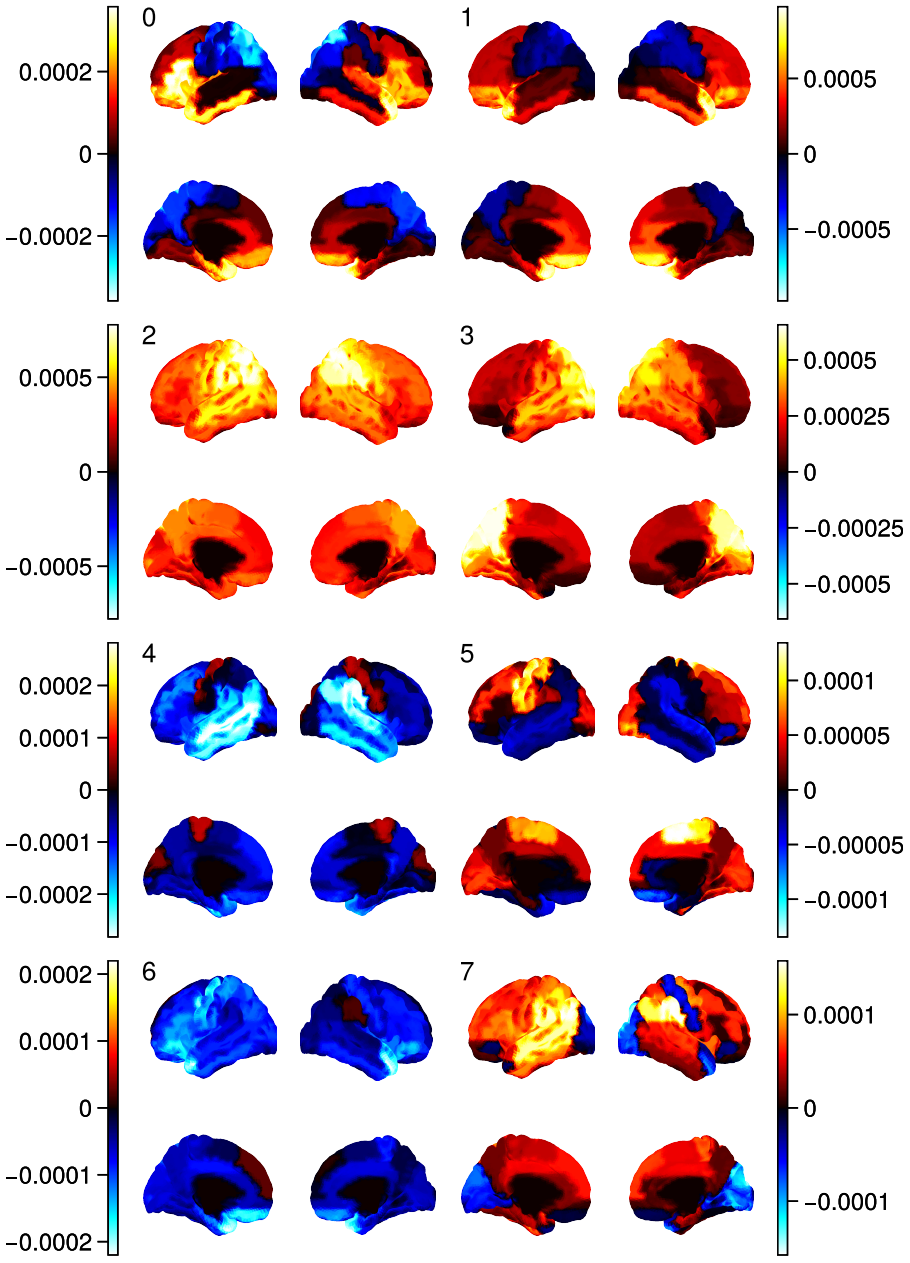


Figure 6.20: The power maps of the metamodes found by maximising the total correlation between mode alphas within each metamode, with two dummy variables per parcel. The variance of each metamode is projected onto the surface of the brain. The latter metamodes show less activation than the first six, but still capture some of the variance in the data. There is less spatial cohesion in meta-modes 5, 6 and 7, but they still show some structure.

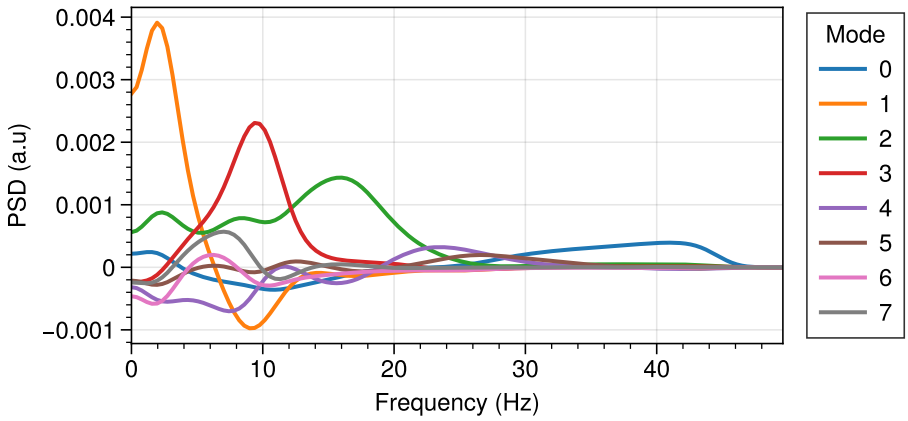


Figure 6.21: The PSDs of the metamodes found by maximising the total correlation between mode alphas within each metamode, with two dummy variables per parcel. Peaks 0 - 5 show good frequency separation, with peaks at 1.95, 9.38, 16.02, 23.44, 26.56 and 41.02 Hz. Peaks 6 and 7 have overlapping peaks which correspond to the peak in mode 5 (7.81 Hz) from the six meta-mode analysis (Figure 6.16).

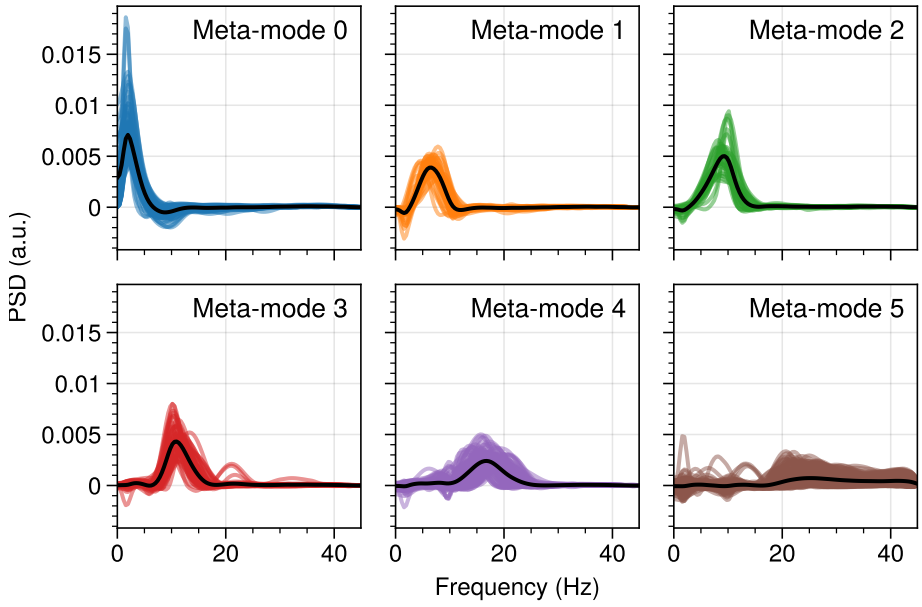


Figure 6.22: The clusters of spectra found by applying KMeans clustering to the parcel-mode PSDs. The clusters are sorted by their peak frequency. They show distinct spectral peaks at 1.95, 6.64, 9.375, 10.93, 16.80 and 25.39 Hz respectively.

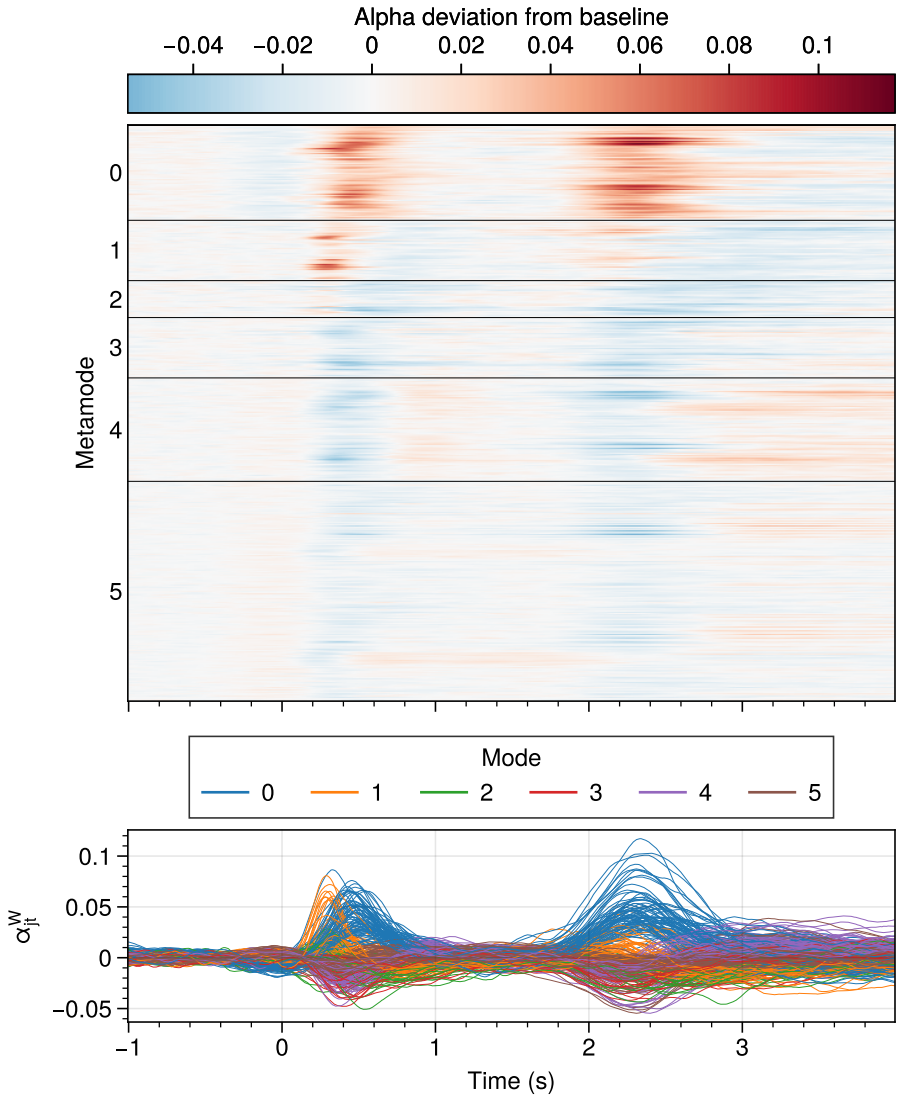


Figure 6.23: The parcel-mode alphas grouped by meta-mode for single-channel DyNeMo grouped by frequency cluster. The alphas are consistent across trials and show a clear separation between meta-modes. The second plot shows the mean over trials and parcel-modes, while the third shows the parcel-mode alphas averaged over trials. This shows that the alphas are consistent across both trials and parcel-modes.

other meta-modes. This makes sense, because as shown in Figure 6.25, although the spectra are indentifiably different, they overlap.

Regressing the meta-mode alphas over the time-frequency content of the dataset, we see that the meta-modes retain the same peaks as the KMeans centroids (Figure 6.25) albeit that the shapes are somewhat distorted.

The surface power maps (Figure 6.26) show distinct spatial activations. The low frequency mode (0 and 1) are both frontal states. Mode 2 (in the alpha

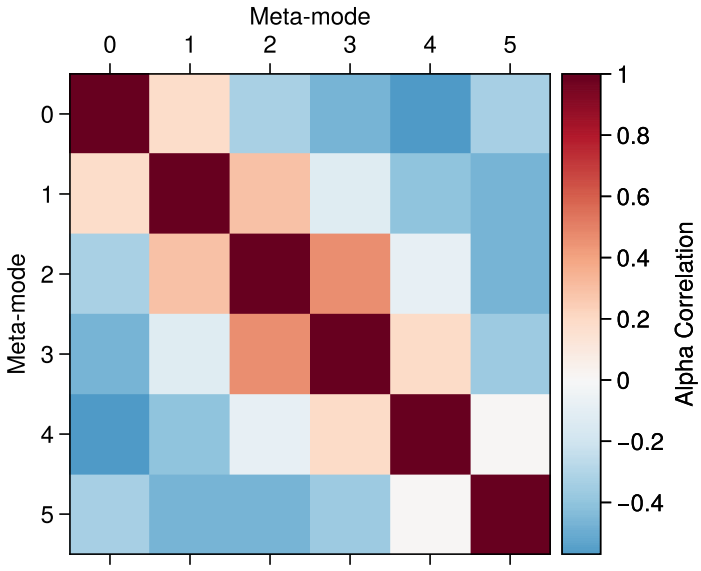


Figure 6.24: The correlation between the meta-mode alphas for single-channel DyNeMo grouped by frequency cluster. The meta-modes are positively correlated with their spectral neighbours and anti-correlated with other meta-modes.

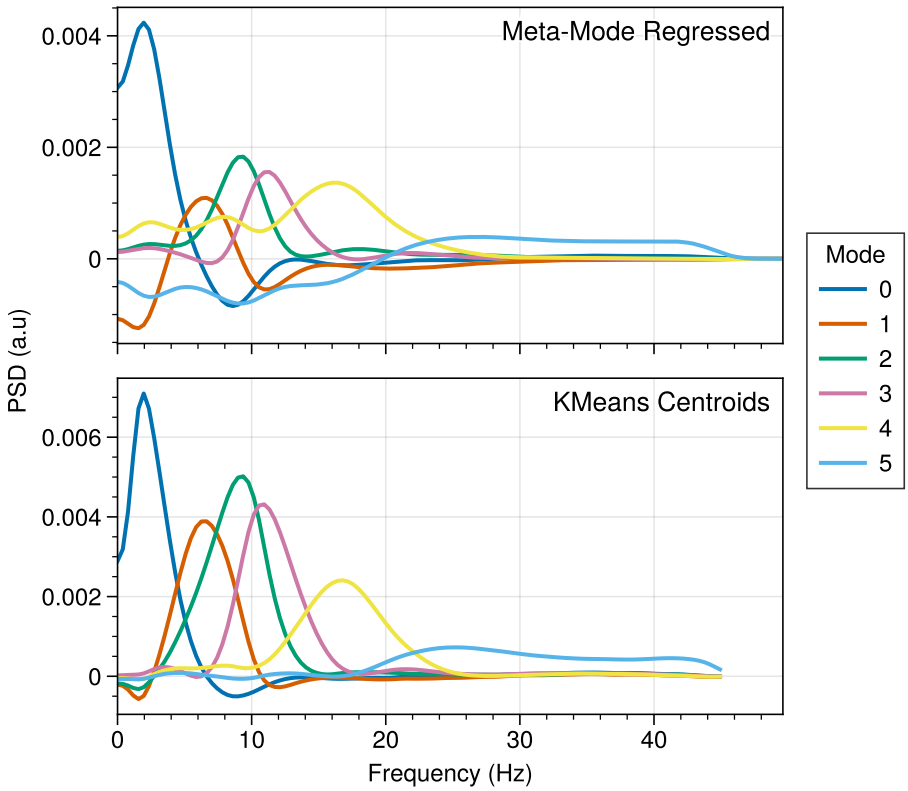


Figure 6.25: The first plot shows the PSDs of the meta-modes found by regressing the meta-mode alphas over the time-frequency content of the dataset. The second plot shows the PSDs of the KMeans centroids. The peaks of the spectra are the same, but the shapes are somewhat distorted.

band) occupies the visual cortex. Mode 3 and 4 (high alpha and low beta) show activation in the motor cortex. Mode 5 (high beta and up) shows deactivation in the motor cortex.

6.3.2.6.2 Autocorrelation Functions

The clusters of spectra in Figure 6.27 have peaks at 3.13, 6.25, 10.94, 15.63, 23.44 and 42.19 Hz. The frequency resolution of the clusters is reduced from that of the regressed PSDs because all of the information is encapsulated within the 23 data points relating to each lag. The parcel-mode alphas also show consistent trial-wise activity and good correlation to the trial structure (Figure 6.28). We see in Figure 6.29 that the meta-mode alpha correlations are positive between nearest spectral neighbours and negative elsewhere. Again, the PSDs found by regressing against the meta-mode alphas match the peaks of the spectra found in the KMeans clusters (Figure 6.30).

Finally, we turn to the power maps (Figure 6.31). Modes 0, 1 and 2 are consistent with those found using the regressed PSDs. Mode 3 matches with regressed mode 4. Mode 4 is most similar to the multichannel mode 4 seen in Figure 6.20. Mode 5 shows a reduced activation in the motor and visual cortices and some activity in more frontal regions.

The assembled metamodes showed strong task dependence and distinct spectral content. The spatial maps are also sensible. In particular it can be seen that the spatial map associated with the motor cortex is most represented in the beta band and is suppressed during the abduction.

6.3.2.7 Alpha RMSD

For the alpha root mean square deviation (RMSD) objective function, we'll start by considering the surface power maps of the meta-modes (Figure 6.32). Comparing them to the power maps of the alpha correlation meta-modes (Figure 6.15), we see some similarities. Mode 0 is approximately the same in both analyses. Modes 1 and 2 have split the content of mode 4 from the alpha correlation analysis, and gained some motor cortex activation. Mode 3 and 5 are distributed among modes

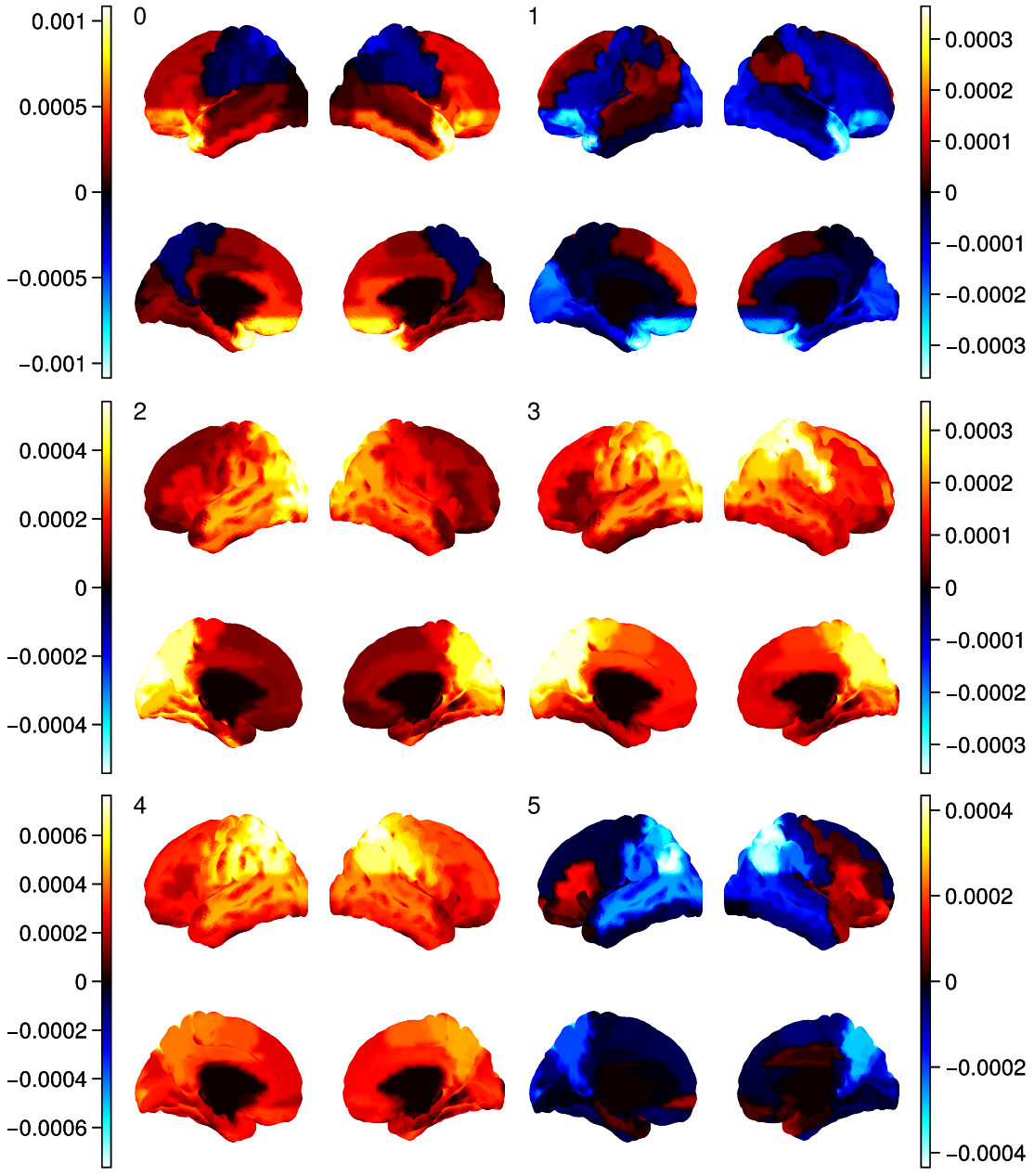


Figure 6.26: The power maps of the meta-modes found by regressing the meta-mode alphas over the time-frequency content of the dataset and performing KMeans clustering on the spectra produced. The variance of each meta-mode is projected onto the surface of the brain. The meta-modes show distinct spatial activations. Notably, modes 3 and 4 (low alpha and low beta) show activation in the motor cortex, while mode 2 (in the alpha band) occupies the visual cortex.

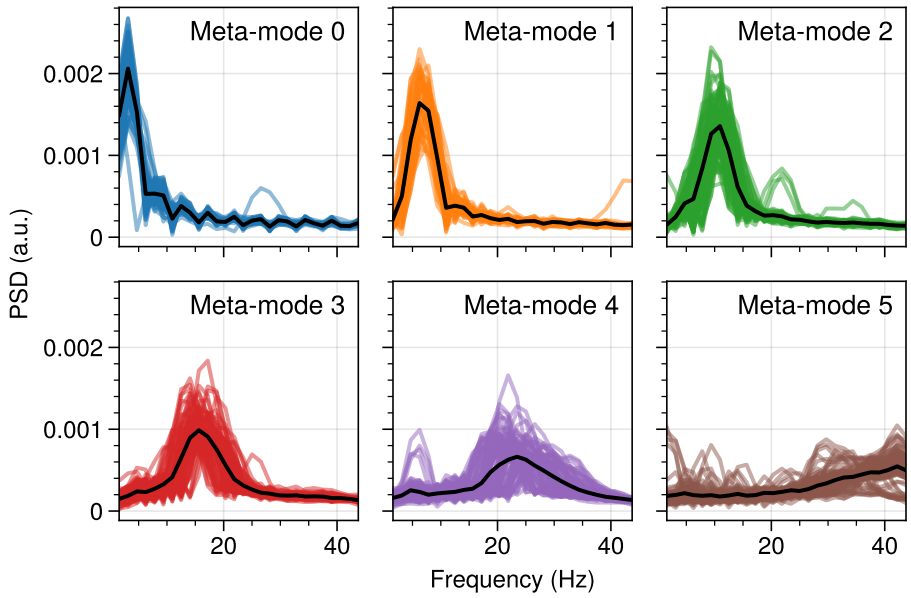


Figure 6.27: The clusters of spectra found by applying KMeans clustering to the parcel-mode autocorrelation functions. The clusters are sorted by their peak frequency. They show distinct spectral peaks at 3.13, 6.25, 10.94, 15.63, 23.44 and 42.19 Hz respectively. These clusters have a lower frequency resolution than those found by regressing the PSDs against the meta-mode alphas.

2, 3 and 4 from the alpha correlation analysis. Finally, mode 4 is similar to mode 1 from the alpha correlation analysis.

The parcel-mode alpha groupings shown in Figure 6.33 show far less consistency than the alpha correlation analysis. However, the metamode mean alphas shown in Figure 6.34 still show high task correlation.

The PSDs of the meta-modes are shown in Figure 6.35. They show poor frequency separation in comparison to the alpha correlation analysis. There are peaks at 1.95, 9.38, 13.67, 21.88 and 28.91 Hz. This ignores mode 0 which has two peaks, one at 1.56 Hz and one at 41.02 Hz.

6.3.2.8 Post-hoc Trial Matching

Meta-modes identified by grouping based on trial activation patterns form pairs of inverse modes. The parcel-mode alphas were epoched by the onset of the visual stimulus with a window of one second before and eight seconds after the stimulus was presented. Following a baseline correction using the second before the stimulus,

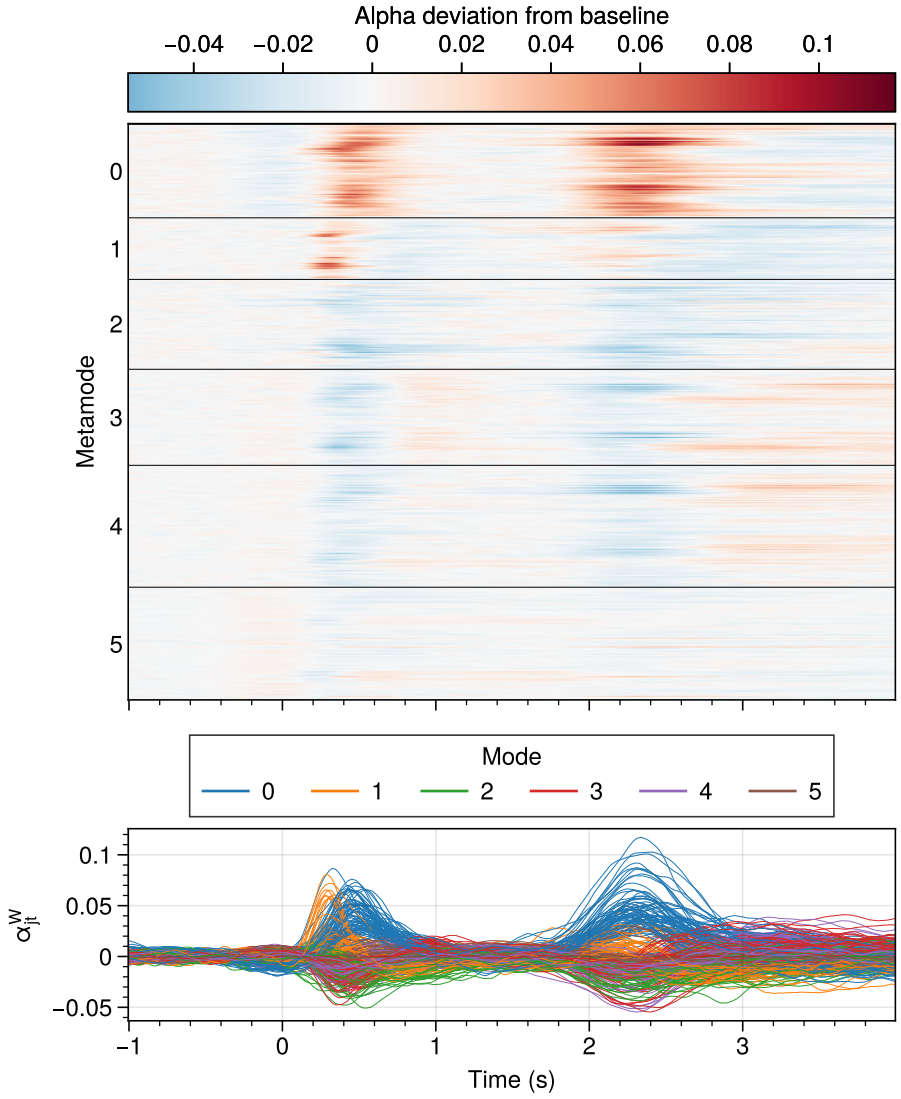


Figure 6.28: The parcel-mode alphas grouped by meta-mode for single-channel DyNeMo grouped by frequency cluster (autocorrelation functions). The alphas are consistent across trials and show a clear separation between meta-modes. The second plot shows the mean over trials and parcel-modes, while the third shows the parcel-mode alphas averaged over trials. This shows that the alphas are consistent across both trials and parcel-modes.

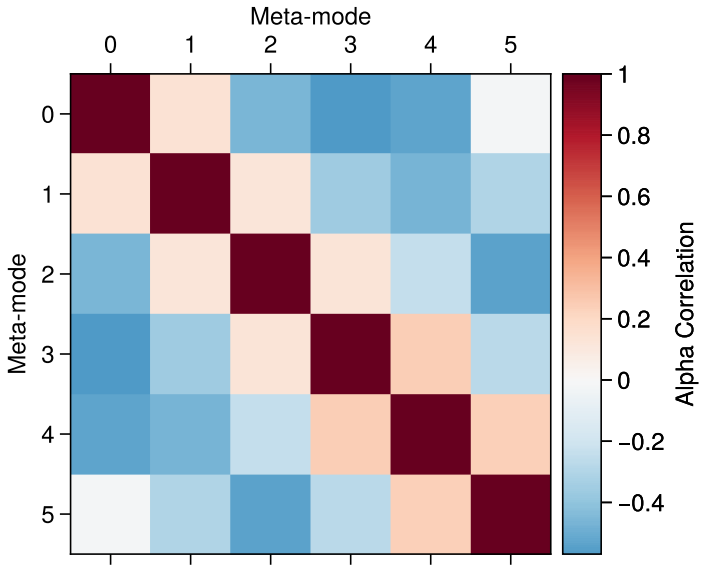


Figure 6.29: The correlation between the meta-mode alphas for single-channel DyNeMo grouped by frequency cluster (autocorrelation functions). The meta-modes are positively correlated with their spectral neighbours and anti-correlated with other meta-modes.

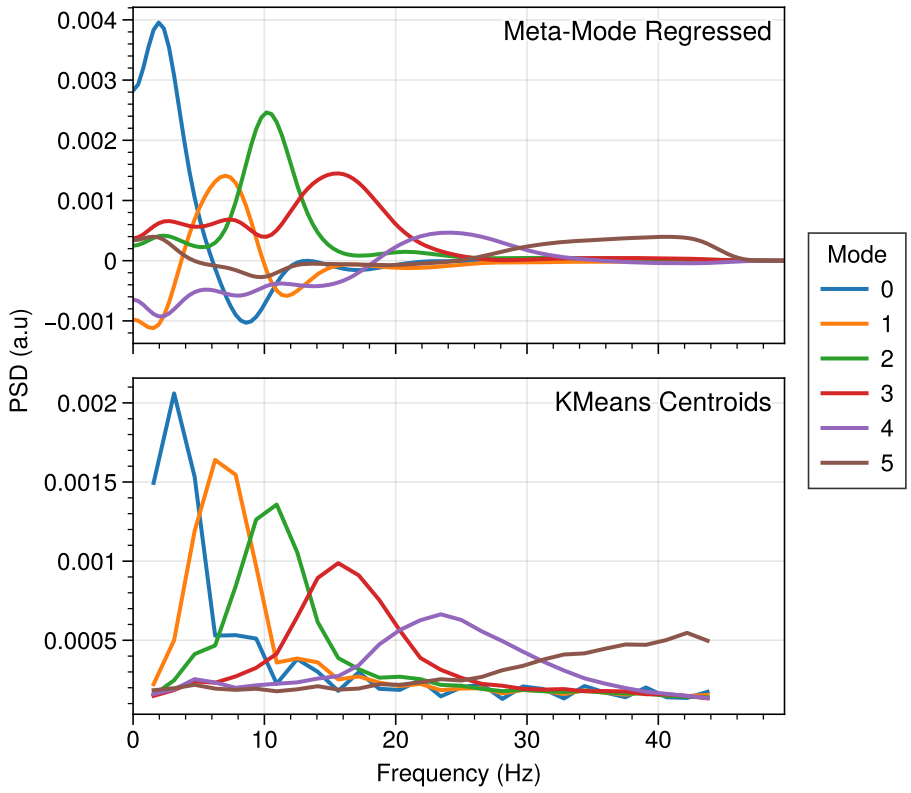


Figure 6.30: The first plot shows the PSDs of the meta-modes found by regressing the meta-mode alphas over the time-frequency content of the dataset. The second plot shows the PSDs of the KMeans centroids of the autocorrelation functions. Even though the frequency resolution is lower, the peaks of the KMeans spectra are the same as those found using regression.

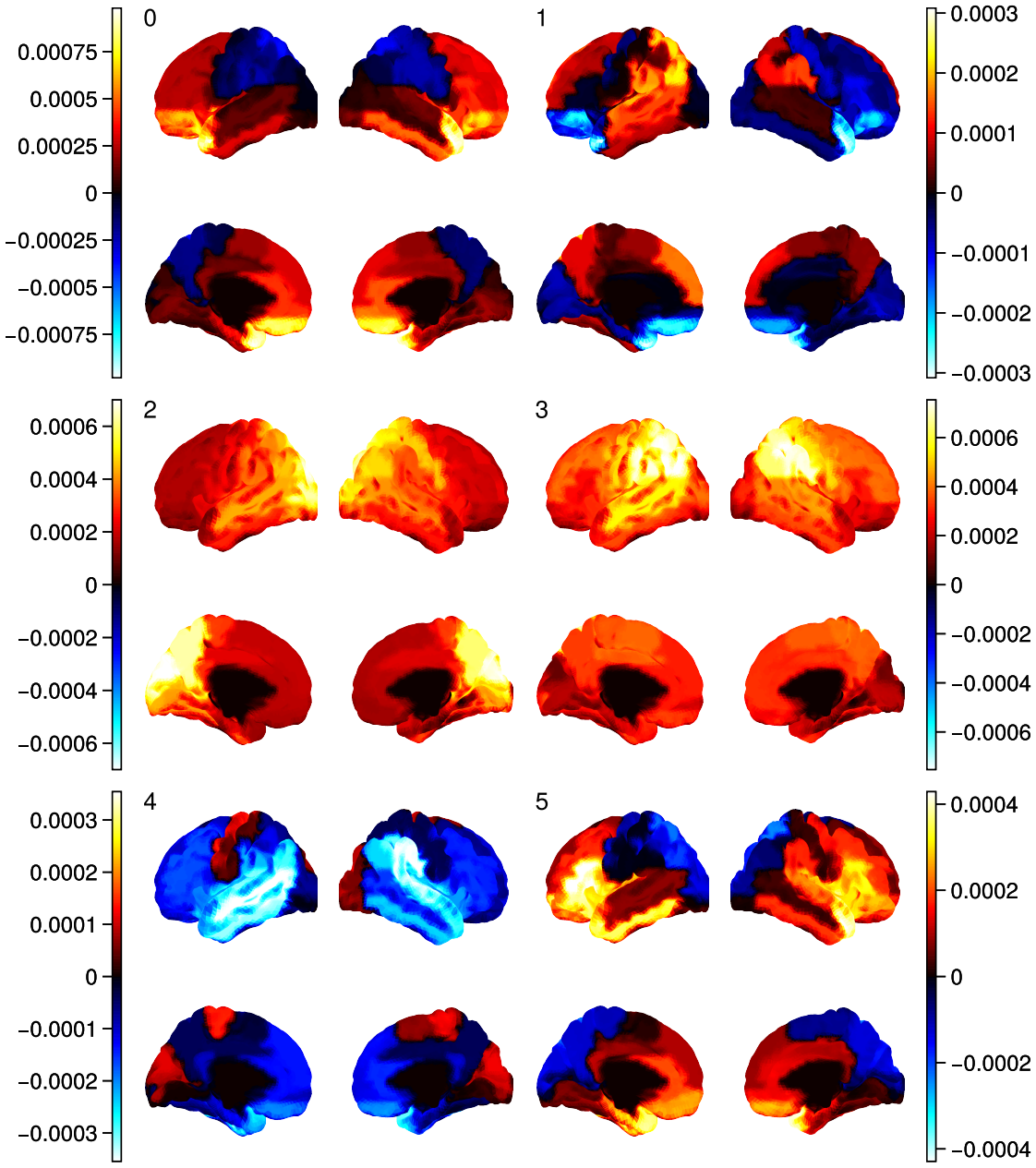


Figure 6.31: The power maps of the meta-modes found by regressing the meta-mode alphas over the time-frequency content of the dataset. The variance of each meta-mode is projected onto the surface of the brain. The meta-modes were found by grouping the parcel-modes by spectral content. These power maps are consistent with those found using the regressed PSDs.

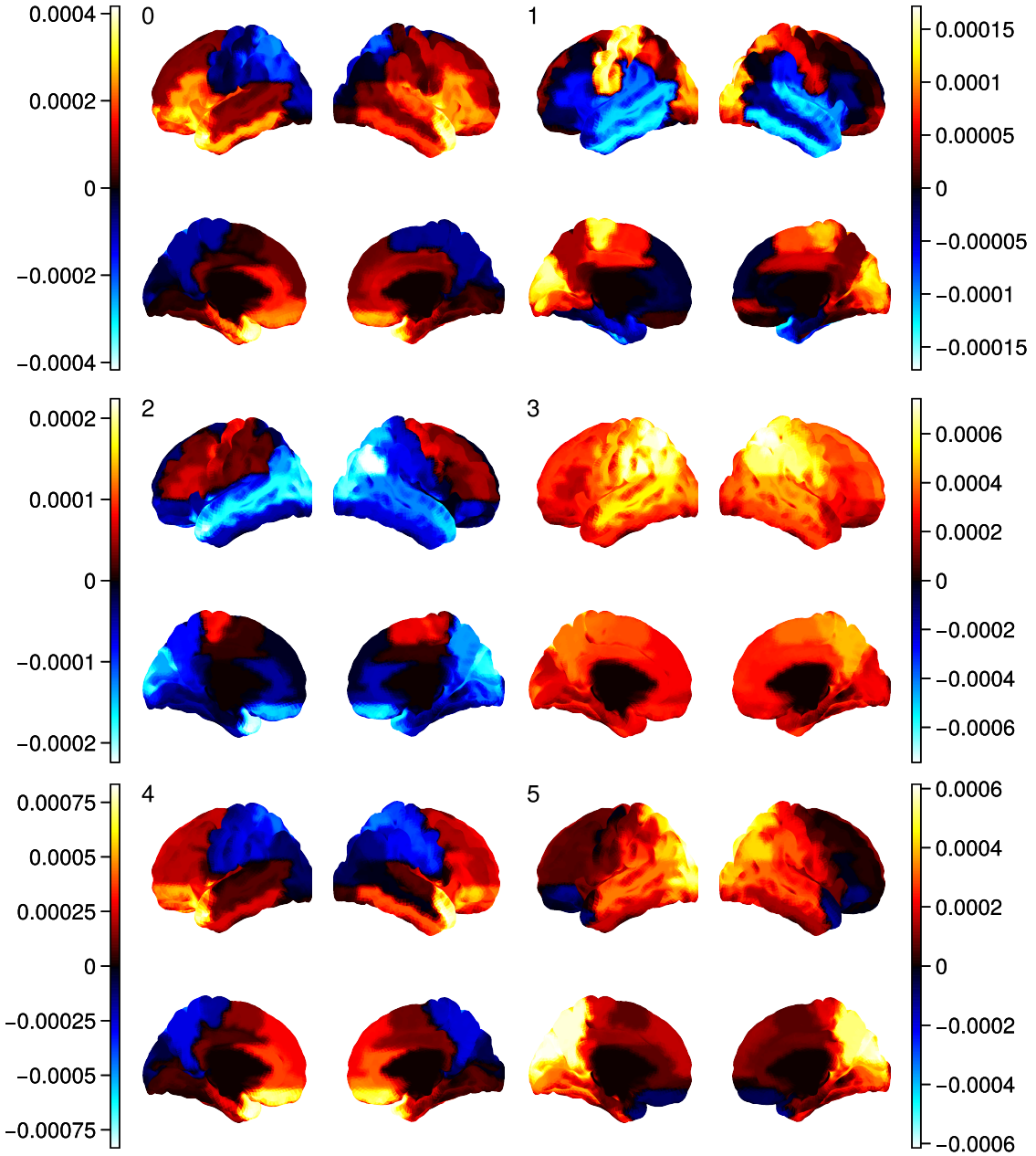


Figure 6.32: The power maps of the meta-modes found by minimising the RMSD between the alphas within meta-modes. The variance of each meta-mode is projected onto the surface of the brain. Modes 1 and 3 show activation in the motor cortex, while mode 5 shows activation in the visual cortex. These modes show less spatial cohesion than those found by maximising the alpha correlation.

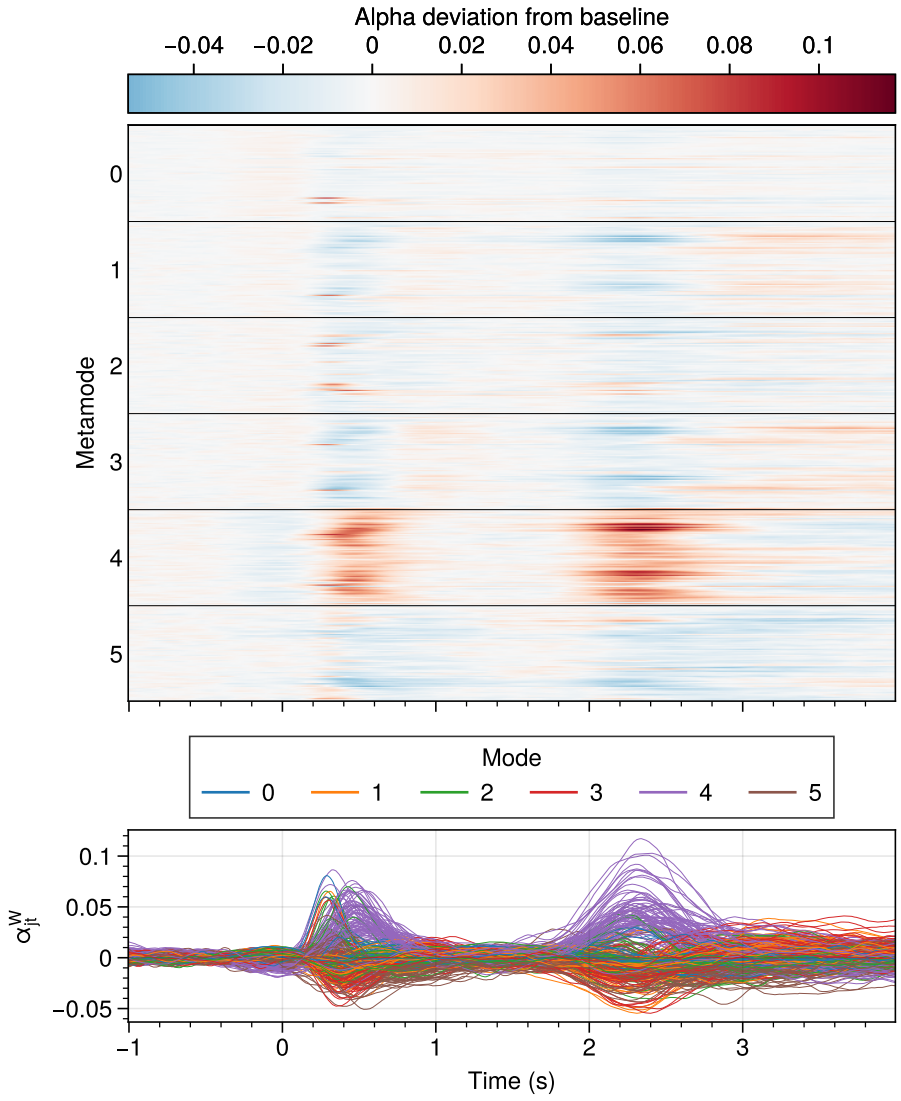


Figure 6.33: Trial averaged alphas for parcel-modes. The groupings of parcel-modes into metamodes is performed by minimising the RMSD between the alphas within meta-modes. While there are some consistent patterns of activation within each meta-mode, the meta-modes show less consistency than those found by maximising the alpha correlation.

a KMeans algorithm with six clusters was applied. The centroids of the KMeans clusters showed groupings of the task alpha response (Figure 6.36). We see that all six centroids have two peaks in their profiles. Both the relative and absolute heights of these peaks change between the clusters. As with the frequency clustering method, KMeans and simulated annealing using the trial alpha correlation will find near identical results.

Showing all the parcel-mode alphas in Figure 6.37, we see good trial consis-

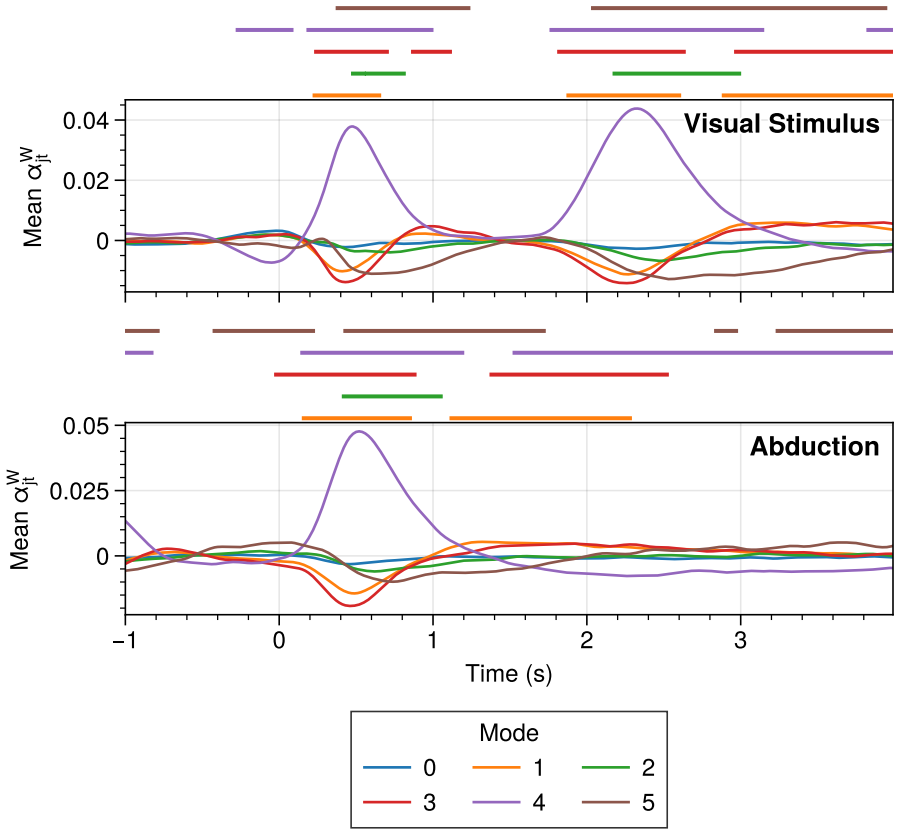


Figure 6.34: The mean of the trial averaged alphas for each metamode. The metamodes were found by minimising the RMSD between the alphas within meta-modes. The coloured lines above each plot represent timepoints for each metamode where the p-value of the task correlation is less than 0.05 according to a sign-flipping permutation test. We see high task correlation within each metamode except for mode 0 which shows no task correlation.

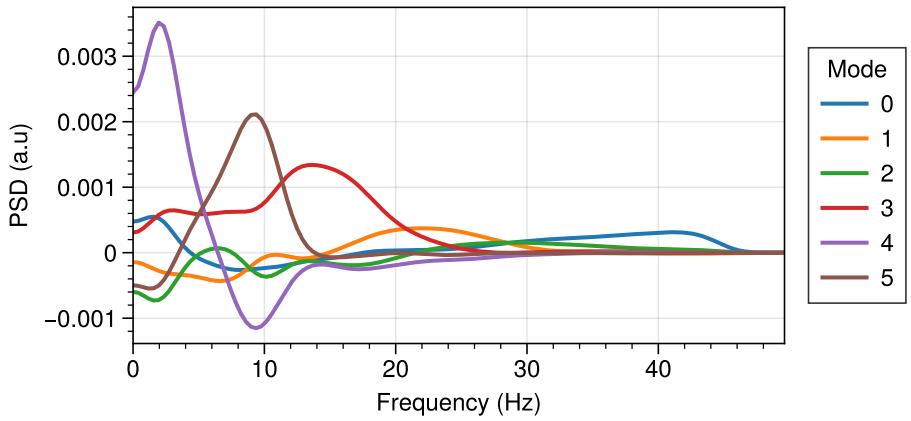


Figure 6.35: The PSD of the metamodes found by minimising the RMSD between the alphas within meta-modes. The metamodes show poor frequency separation, with peaks at 1.95, 9.38, 13.67, 21.88 and 28.91 Hz. Mode 0 has two peaks, one at 1.56 Hz and one at 41.02 Hz.

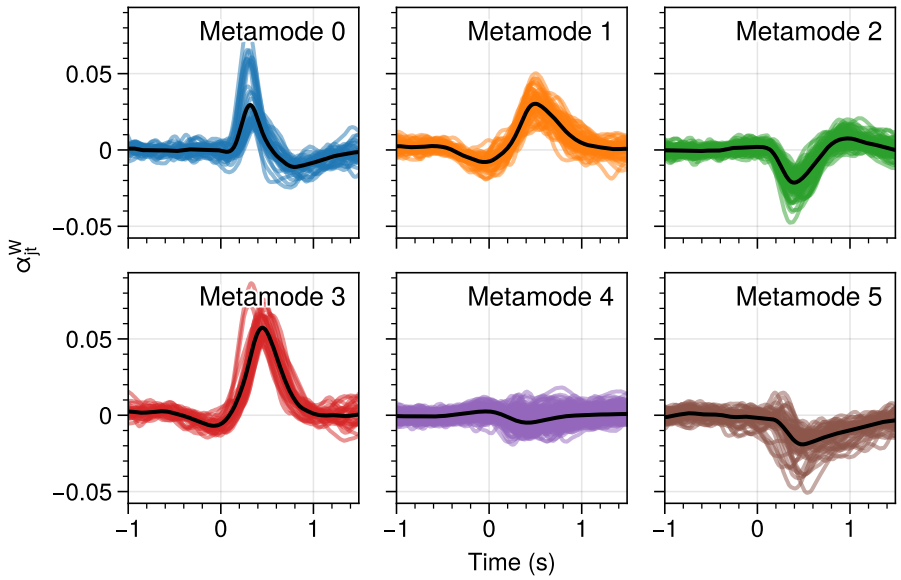


Figure 6.36: Clustered trial responses for single-channel DyNeMo. The centroids of the KMeans clusters show groupings of the task alpha response. The clustering is only performed around the visual stimulus. The clustering algorithm has successfully found six clusters with different responses to the stimulus.

tency and a mean of trials with good task correlation in Figure 6.38. This is unsurprising as we chose the meta-modes based on their similarity to each other during the trial window.

The power maps (Figure 6.39) created through this method are somewhat different to the ones created through the other methods in Section 6.3. There is a much greater focus on the areas responsible for the task response. There are also a series of paired, inverted modes. Modes 0 and 2 show activation in the motor cortex, with mode 1 being an inverted copy of mode 2. Mode 3 is a frontal state. Modes 4 and 5 respectively show an increase and decrease of power in the visual cortex.

Compared to the orderly PSDs we have become used to seeing in the meta-modes assembled through frequency clustering, the frequency distribution of these modes is less distinct (Figure 6.40). Modes 0, 1 and 2 show their biggest deviation from the baseline in the alpha range. Mode 3 is active in the low frequency range. Mode 4 peaks in the alpha, and again in the beta band. Mode 5 is highest in the high beta and above.

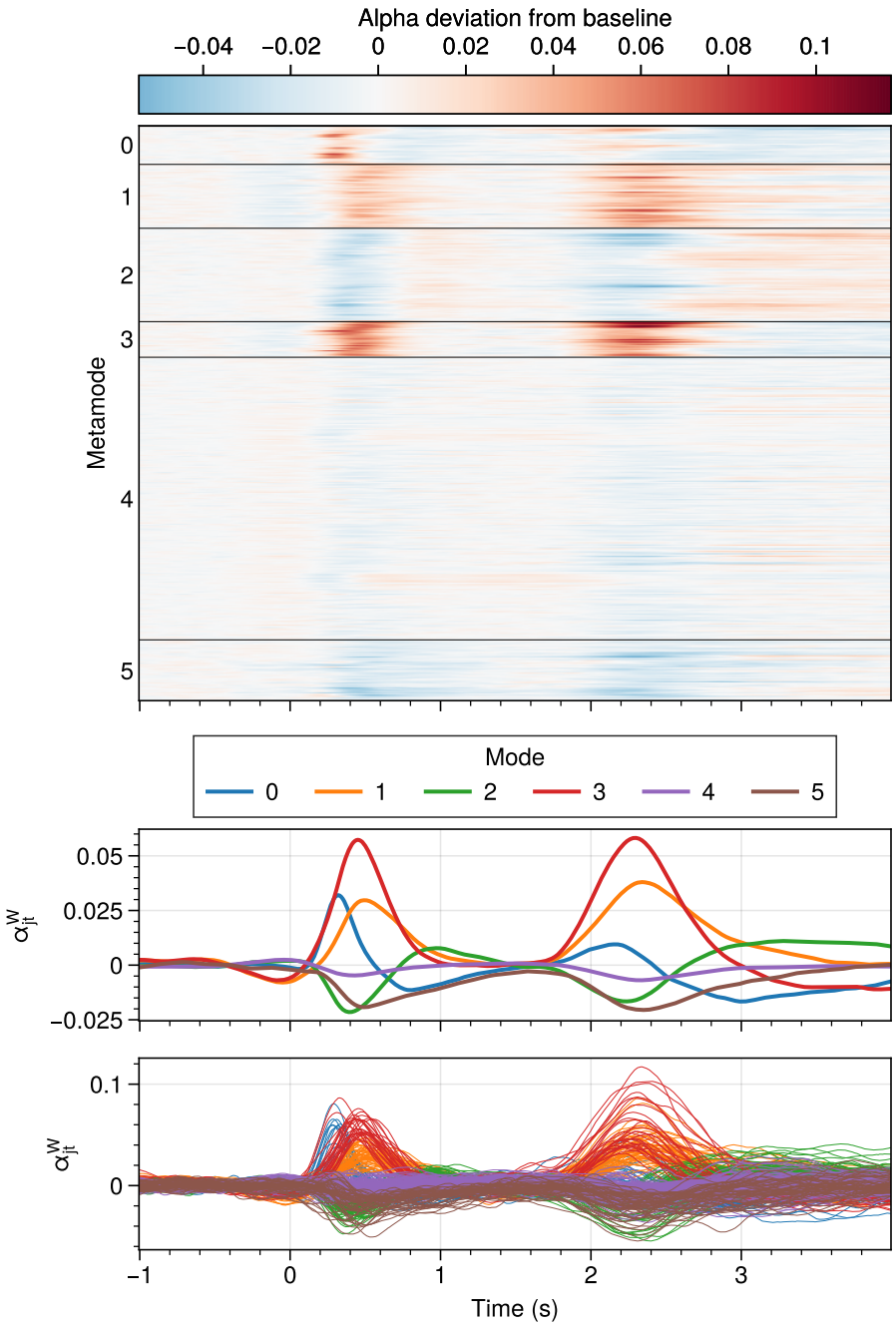


Figure 6.37: The parcel-mode alphas grouped by meta-mode for single-channel DyNeMo grouped by trial response. The alphas are consistent across trials and show a clear separation between meta-modes. The second plot shows the mean over trials and parcel-modes, while the third shows the parcel-mode alphas averaged over trials. This shows that the alphas are consistent across both trials and parcel-modes.

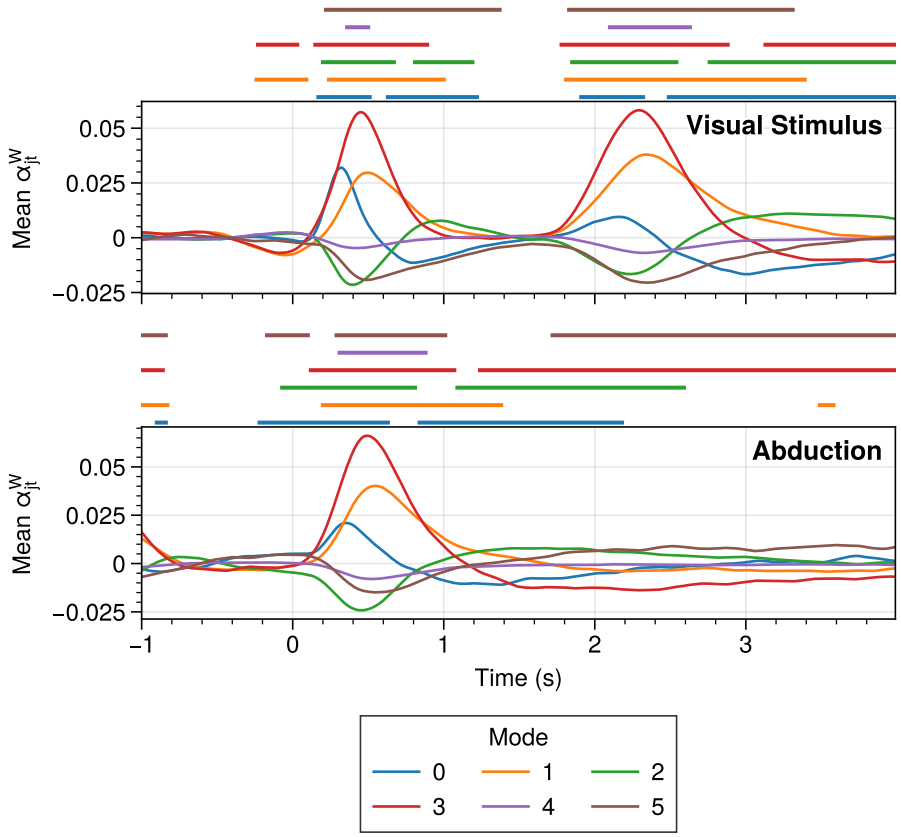


Figure 6.38: The mean of the trial averaged alphas for each metamode for single-channel DyNeMo grouped by trial response. The metamodes were found by regressing the meta-mode alphas over the time-frequency content of the dataset. The coloured lines above each plot represent timepoints for each metamode where the p-value of the task correlation is less than 0.05 according to a sign-flipping permutation test. We see high task correlation within each metamode.

6.4 Summary

In this chapter, we have shown that single-channel DyNeMo can be used to find consistent patterns of activation across parcels. We have shown that the metamodes found by single-channel DyNeMo are consistent across trials and show high task correlation. Several methods for grouping the parcel-mode alphas into metamodes were presented. The metamodes found through these methods were more similar to each other than to the metamodes found using multichannel DyNeMo.

Of particular note is the method for grouping the parcel-modes by their frequency content. This method was able to find metamodes which showed a clear separation in spectral profile. This makes sense as single-channel DyNeMo learns

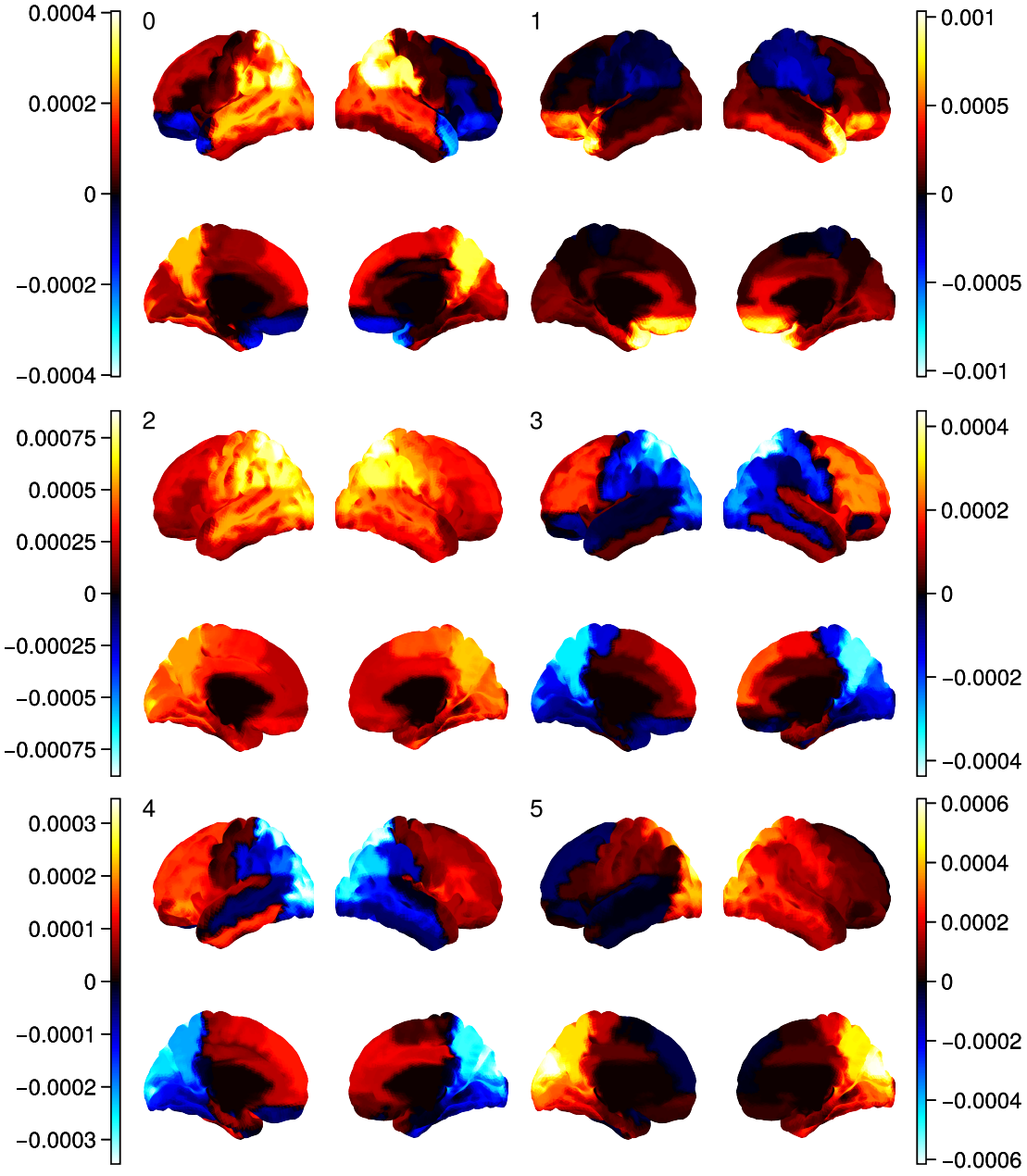


Figure 6.39: The power maps of the meta-modes found by regressing the meta-mode alphas over the time-frequency content of the dataset. Metamodes were found by grouping the parcel-mode alphas by trial response. The variance of each meta-mode is projected onto the surface of the brain. The modes show activation in the motor cortex, visual cortex and frontal state. Mode 2 is active in the motor cortex and corresponds to the beta frequency band. Its activity reduces during the task as expected for a motor task.

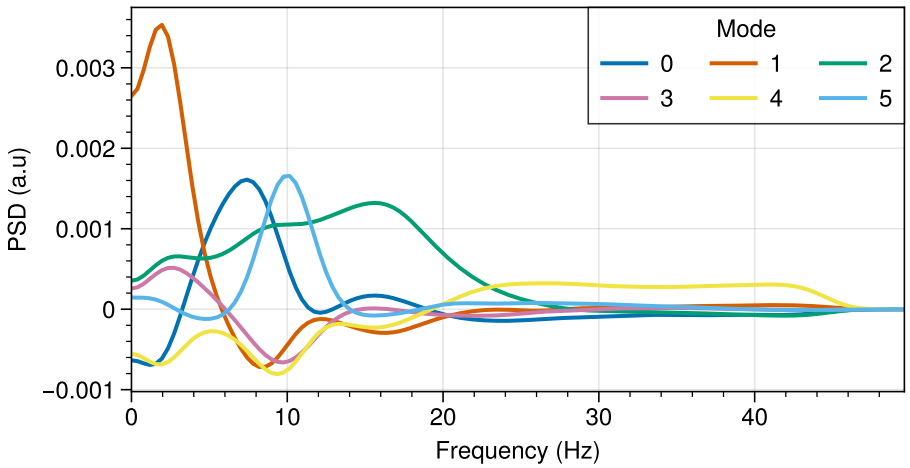


Figure 6.40: PSDs of the meta-modes determined by grouping the parcel-mode alphas by trial response. The PSDs of the meta-modes found by regressing the meta-mode alphas over the time-frequency content of the dataset. We see a much broader frequency response in mode 2 than in the other modes. This is the mode which includes the motor cortex and is active in the beta band. The other modes have more typical frequency separation.

modes based on the autocorrelation of the parcel-modes which is the Fourier transform of the power spectral density.

7
Conclusions

DyNeMo provides an opportunity to move away from the constraints of mutual exclusivity and memorylessness which are imposed by the Markovian property of the HMM. DyNeMo models are capable of inferring states which coactivate, allowing for improved inference of modes which overlap both spatially and temporally. The generative model can be used to demonstrate that DyNeMo can successfully learn long-term dependencies from input data. The presence of background modes allows for a cleaner representation of the data, as the model does not need to incorporate background effects into the modes which represent activity.

In the multi-channel regime, DyNeMo identifies known spatial maps with expected PSDs. The modes found when trained on task data correspond closely to those from resting state data. The task data also allowed the model to prove that it could produce modes with a strong task dependence. This helps to validate the usefulness of the model by giving an approximate "ground truth" within real data. Again using the task structure, DyNeMo shows a slightly improved spectral recovery in comparison to the HMM. The improvement is minor, but sets a lower bound from which improvements can be assessed.

DyNeMo can be trained on very large datasets thanks to the use of amortised inference. The `osl-dynamics` package also enables this through the use of memory mapped data files. The datasets examined in this work are small (in the order of

50 subjects), but the size of the model does not scale with the length of the dataset and so the only constraints on larger datasets are time and data storage.

Analysing DyNeMo results can be more difficult than those of the HMM. The lack of binarization makes some statistical methods from HMM analyses meaningless. To avoid this, post-hoc binarization methods and regression methods are provided. These allow analysis to be carried out in much the same way as for previous HMM analyses.

There are many hyperparameters involved in the training of DyNeMo. This space has not been widely explored. Instead, a set of reasonable parameters with consistent results was chosen. The use of multiple initialisations was also implemented to improve run-to-run variability.

In the single-channel regime, DyNeMo identifies parcel-modes which are individually correlated with the task structure. In order to reduce the analysis to a number of dimensions which is easy to understand, a set of metamodes were selected. Subsequent analysis of these metamodes showed that there are many ways to arrange the parcel-modes to understand differing properties of the data. In this work, alpha correlations, frequency clustering and task activity clustering were used to generate metamodes. Each showed a different perspective of the data, though none matched the multi-channel DyNeMo modes particularly closely.

Single-channel DyNeMo represents the opportunity for training on much smaller hardware than multi-channel DyNeMo. This means more researchers will be able to use it even if they do not have GPUs with large amounts of memory. It also provides the ability to modify the target of the analysis after training using different metamode objective functions. This flexibility is useful as the same models can be used for multiple separate analyses.

We have shown that in this regime, DyNeMo can be used to find modes with distinct autocorrelation functions. This means that more focus is placed on the power spectral shape of the modes, rather than the spatial maps and variance which dominate the multi-channel regime. This is a useful property as it allows

for the identification of modes which are not necessarily spatially distinct, but which have distinct spectra.

The opportunities presented by DyNeMo for more in depth analysis of neuroimaging data are plentiful. With many technical hurdles overcome, the foundation has been laid for interesting research and further development in this vein.

7.1 Future Work

There are many potential directions for the continuation of this project. Some extensions to DyNeMo and related models are already being pursued by members of the lab. These include subject embedded models and models with separated dynamics for power and connectivity.

DyNeMo can be used to model anywhere the HMM can be used. This includes modelling the dynamics of other types of data, such as EEG and fMRI or even non-neuroimaging data. As it presents a descriptive, low dimensional representation of the data, it may be useful in decoding analyses. In decoding, the goal is to predict some property extrinsic to the data, such as a behavioural score or a clinical label. The low dimensionality of the DyNeMo modes may make them more amenable to decoding than high dimensional MEG data.

A related algorithm, Temporally Adaptive SourcE Reconstruction (TASER), has been developed by Ryan Timms.[95] TASER uses the same underlying model to perform source reconstruction in a data driven manner. Combining TASER with DyNeMo could form a pipeline for source reconstruction and analysis of the reconstructed sources. This work could be used to investigate raw MEG data without some of the assumptions made during the data preprocessing stage.

Subject embedded models are a natural extension of DyNeMo. These models would allow for the modelling of subject specific dynamics, which may be useful in clinical applications. Training on concatenated data from multiple subjects may hide differences between subjects. Although these can be partially recovered during post-hoc analysis, it may be that more accurate results can be obtained by training an embedded model.

Finding an appropriate parameterisation for the covariance matrices of the modes is an important step in improving the model. If the multi-channel model could be modified to use a block Toeplitz structure, it would be possible to use the same parameterisation in the single-channel regime. This could lead to memory savings large enough to allow for the triaining of models without dimensionality reduction, or with a much larger number of embeddings.

There is also additional work to be done in the single-channel regime. In particular, to determine whether a set of single-channel models can be reliably combined to replicate the training of a multi-channel mode. It may be that this is not possible with the current state of the model, not least because it doesn't model the variance of each mode during training. The frequency clustering approach presented in this work is a promising avenue. Combined with a method for finding the optimal number of clusters, it may turn out to be a useful way to find modes which are consistent across subjects.

References

[1] Ashlea Segal et al. "Regional, Circuit and Network Heterogeneity of Brain Abnormalities in Psychiatric Disorders". In: *Nature Neuroscience* 26.9 (9 Sept. 2023), pp. 1613–1629. ISSN: 1546-1726. DOI: 10.1038/s41593-023-01404-6.

[2] Manish Jha and Madhukar Trivedi. "Personalized Antidepressant Selection and Pathway to Novel Treatments: Clinical Utility of Targeting Inflammation". In: *International Journal of Molecular Sciences* 19.1 (Jan. 12, 2018), p. 233. ISSN: 1422-0067. DOI: 10.3390/ijms19010233.

[3] Vince Calhoun et al. "Exploring the Psychosis Functional Connectome: Aberrant Intrinsic Networks in Schizophrenia and Bipolar Disorder". In: *Frontiers in Psychiatry* 2 (2012). ISSN: 1664-0640.

[4] Diego Vidaurre et al. "Spectrally Resolved Fast Transient Brain States in Electrophysiological Data". In: *NeuroImage* 126 (Feb. 2016), pp. 81–95. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2015.11.047.

[5] Diego Vidaurre, Stephen M. Smith, and Mark W. Woolrich. "Brain Network Dynamics Are Hierarchically Organized in Time". In: *Proceedings of the National Academy of Sciences* 114.48 (Nov. 28, 2017), pp. 12827–12832. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1705120114.

[6] Diego Vidaurre et al. "Discovering Dynamic Brain Networks from Big Data in Rest and Task". In: *NeuroImage* 180 (Oct. 2018), pp. 646–656. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2017.06.077.

[7] Diego Vidaurre et al. "Spontaneous Cortical Activity Transiently Organises into Frequency Specific Phase-Coupling Networks". In: *Nature Communications* 9.1 (July 30, 2018), p. 2987. ISSN: 2041-1723. DOI: 10.1038/s41467-018-05316-z.

[8] Andrew J. Quinn et al. "Task-Evoked Dynamic Network Analysis Through Hidden Markov Modeling". In: *Frontiers in Neuroscience* 12 (2018). ISSN: 1662-453X.

[9] Joana Cabral et al. "Cognitive Performance in Healthy Older Adults Relates to Spontaneous Switching between States of Functional Connectivity during Rest". In: *Scientific Reports* 7.1 (1 July 11, 2017), p. 5135. ISSN: 2045-2322. DOI: 10.1038/s41598-017-05425-7.

[10] Diego Vidaurre et al. "Temporally Unconstrained Decoding Reveals Consistent but Time-Varying Stages of Stimulus Processing". In: *Cerebral Cortex* 29.2 (Feb. 1, 2019), pp. 863–874. ISSN: 1047-3211. DOI: 10.1093/cercor/bhy290.

[11] Cameron Higgins et al. "Spatiotemporally Resolved Multivariate Pattern Analysis for M/ EEG". In: *Human Brain Mapping* 43.10 (July 2022), pp. 3062–3085. ISSN: 1065-9471, 1097-0193. DOI: 10.1002/hbm.25835.

[12] Semir Zeki. "A Massively Asynchronous, Parallel Brain". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 370.1668 (May 19, 2015), p. 20140174. DOI: 10.1098/rstb.2014.0174.

[13] Adam P Baker et al. "Fast Transient Networks in Spontaneous Human Brain Activity". In: *eLife* 3 (Mar. 25, 2014), e01867. ISSN: 2050-084X. DOI: 10.7554/eLife.01867.

[14] C. J. P. Grimmelikhuijzen and J. A. Westfall. "The Nervous Systems of Cnidarians". In: *The Nervous Systems of Invertebrates: An Evolutionary and Comparative Approach: With a Coda Written by T.H. Bullock*. Ed. by O. Breidbach and W. Kutsch. Experientia Supplementum. Basel: Birkhäuser, 1995, pp. 7–24. ISBN: 978-3-0348-9219-3. DOI: 10.1007/978-3-0348-9219-3_2.

[15] Detlev Arendt et al. "The Evolution of Nervous System Centralization". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 363.1496 (Apr. 27, 2008), pp. 1523–1528. ISSN: 0962-8436. DOI: 10.1098/rstb.2007.2242. pmid: 18192182.

[16] Tae-Yoon S. Park et al. "Brain and Eyes of Kerygmachela Reveal Protocerebral Ancestry of the Panarthropod Head". In: *Nature Communications* 9 (Mar. 9, 2018), p. 1019. ISSN: 2041-1723. DOI: 10.1038/s41467-018-03464-w. pmid: 29523785.

[17] J. B. Smaers et al. "The Evolution of Mammalian Brain Size". In: *Science Advances* 7.18 (Apr. 28, 2021), eabe2101. DOI: 10.1126/sciadv.abe2101.

[18] David Attwell and Simon B. Laughlin. "An Energy Budget for Signaling in the Grey Matter of the Brain". In: *Journal of Cerebral Blood Flow & Metabolism* 21.10 (Oct. 1, 2001), pp. 1133–1145. ISSN: 0271-678X. DOI: 10.1097/00004647-200110000-00001.

[19] Suzana Herculano-Houzel. "The Human Brain in Numbers: A Linearly Scaled-up Primate Brain". In: *Frontiers in Human Neuroscience* 3 (2009). DOI: 10.3389/neuro.09.031.2009.

[20] Frederico A.C. Azevedo et al. "Equal Numbers of Neuronal and Nonneuronal Cells Make the Human Brain an Isometrically Scaled-up Primate Brain". In: *Journal of Comparative Neurology* 513.5 (2009), pp. 532–541. ISSN: 1096-9861. DOI: 10.1002/cne.21974.

[21] Christopher S. von Bartheld, Jami Bahney, and Suzana Herculano-Houzel. "The Search for True Numbers of Neurons and Glial Cells in the Human Brain: A Review of 150 Years of Cell Counting". In: *Journal of Comparative Neurology* 524.18 (2016), pp. 3865–3895. ISSN: 1096-9861. DOI: 10.1002/cne.24040.

[22] "Gray, White, and Reticular Matter". In: Bryan Kolb. *Fundamentals of Human Neuropsychology*. In collab. with Ian Q. Whishaw. 5th ed. Series of Books in Psychology. New York, NY: Worth Publishers, 2003, p. 49. ISBN: 978-0-7167-5300-1.

[23] Ricardo Vieira Teles. "Phineas Gage's Great Legacy". In: *Dementia & Neuropsychologia* 14.4 (Dec. 2020), pp. 419–421. ISSN: 1980-5764. DOI: 10.1590/1980-57642020dn14-040013. pmid: 33354296.

[24] Kieran O'Driscoll and John Paul Leach. ""No Longer Gage": An Iron Bar through the Head". In: *BMJ : British Medical Journal* 317.7174 (Dec. 19, 1998), pp. 1673–1674. ISSN: 0959-8138. pmid: 9857119.

[25] Paul Broca. "Comments Regarding the Seat of the Faculty of Spoken Language, Followed by an Observation of Aphemia (Loss of Speech) (1861)". In: *Broca's Region*. Oxford University Press, May 2006, pp. 291–304. ISBN: 978-0-19-517764-0. DOI: 10.1093/acprof:oso/9780195177640.003.0018. eprint: https://academic.oup.com/book/9725/chapter/156901248/chapter-ag-pdf/44946150/book_9725_section_156901248.ag.pdf.

[26] Paul Broca. "Remarques Sur Le Siège de La Faculté Du Langage Articulé, Suivies d'une Observation d'aphémie (Perte de La Parole)". In: *Bulletin et Memoires de la Societe anatomique de Paris* 6 (1861), pp. 330–357.

[27] Ennis Ata Berker, Ata Husnu Berker, and Aaron Smith. "Translation of Broca's 1865 Report: Localization of Speech in the Third Left Frontal Convolution". In: *Archives of Neurology* 43.10 (Oct. 1986), pp. 1065–1072. ISSN: 0003-9942. DOI: 10.1001/archneur.1986.00520100069017. eprint: https://jamanetwork.com/journals/jamaneurology/articlepdf/585803/archneur_43_10_017.pdf.

[28] C. Wernicke. "THE APHASIC SYMPTOM-COMPLEX: A Psychological Study on an Anatomical Basis*". In: *Archives of Neurology* 22.3 (Mar. 1970), pp. 280–282. ISSN: 0003-9942. DOI: 10.1001/archneur.1970.00480210090013. eprint: https://jamanetwork.com/journals/jamaneurology/articlepdf/569614/archneur_22_3_013.pdf.

[29] C. Wernicke. *Der Aphasische Symptomencomplex: Eine Psychologische Studie Auf Anatomischer Basis*. Max cohn & weigert, 1874.

[30] Richard Caton. "Electrical Currents of the Brain". In: *The Journal of Nervous and Mental Disease* 2.4 (1875), p. 610.

[31] Hans Berger. "Über Das Elektroenkephalogramm Des Menschen". In: *Archiv für psychiatrie und nervenkrankheiten* 87.1 (1929), pp. 527–570.

[32] I. I. Rabi et al. "A New Method of Measuring Nuclear Magnetic Moment". In: *Physical Review* 53.4 (Feb. 1938), p. 318. ISSN: 1536-6065. DOI: 10.1103/PhysRev.53.318.

[33] F. Bloch. "Nuclear Induction". In: *Physical Review* 70.7-8 (Oct. 1946), pp. 460–474. ISSN: 1536-6065. DOI: 10.1103/PhysRev.70.460.

[34] P. C. Lauterbur. "Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance". In: *Nature* 242.5394 (Mar. 1973), pp. 190–191. ISSN: 1476-4687. DOI: 10.1038/242190a0.

[35] P. C. Lauterbur. "Magnetic Resonance Zeugmatography". In: *Pure and Applied Chemistry* 40.1-2 (Jan. 1974), pp. 149–157. ISSN: 1365-3075. DOI: 10.1351/pac197440010149.

[36] P. Mansfield and P. K. Grannell. ""Diffraction" and Microscopy in Solids and Liquids by NMR". In: *Physical Review B* 12.9 (Nov. 1975), pp. 3618–3634. ISSN: 2469-9969. DOI: 10.1103/PhysRevB.12.3618.

[37] W. S. Hinshaw, P. A. Bottomley, and G. N. Holland. "Radiographic Thin-Section Image of the Human Wrist by Nuclear Magnetic Resonance". In: *Nature* 270.5639 (Dec. 1977), pp. 722–723. ISSN: 1476-4687. DOI: 10.1038/270722a0.

[38] David Cohen. "Magnetoencephalography: Evidence of Magnetic Fields Produced by Alpha-Rhythm Currents". In: *Science (New York, N.Y.)* 161.3843 (Aug. 1968), pp. 784–786. ISSN: 0036-8075. DOI: 10.1126/science.161.3843.784.

[39] Sanjay P. Singh. "Magnetoencephalography: Basic Principles". In: *Annals of Indian Academy of Neurology* 17 (Suppl 1 Mar. 2014), S107. DOI: 10.4103/0972-2327.128676.

[40] A. H. Silver and J. E. Zimmerman. "Quantum Transitions and Loss in Multiply Connected Superconductors". In: *Physical Review Letters* 15.23 (Dec. 1965), pp. 888–891. ISSN: 1079-7114. DOI: 10.1103/PhysRevLett.15.888.

[41] A. I. Ahonen et al. "122-Channel Squid Instrument for Investigating the Magnetic Signals from the Human Brain". In: *Physica Scripta* 1993 (T49A Jan. 1993), p. 198. ISSN: 1402-4896. DOI: 10.1088/0031-8949/1993/T49A/033.

[42] Juha Virtanen. *EEG Combined with MEG and TMS in Studies of Human Brain Function*. Helsinki University of Technology, 1998, p. 23.

[43] Sofie S. Meyer et al. "Flexible Head-Casts for High Spatial Precision MEG". In: *Journal of Neuroscience Methods* 276 (Jan. 30, 2017), pp. 38–45. ISSN: 0165-0270. DOI: 10.1016/j.jneumeth.2016.11.009.

[44] Emma L. Hall et al. "The Relationship between MEG and fMRI". In: *NeuroImage*. Multimodal Data Fusion 102 (Nov. 15, 2014), pp. 80–91. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2013.11.005.

[45] S. P van den Broek et al. "Volume Conduction Effects in EEG and MEG". In: *Electroencephalography and Clinical Neurophysiology* 106.6 (June 1, 1998), pp. 522–534. ISSN: 0013-4694. DOI: 10.1016/S0013-4694(97)00147-8.

[46] "Nerve Cells and Their Connections". In: G. Pocock, C.D. Richards, and D.A. Richards. *Human Physiology*. Oxford Core Texts. OUP Oxford, 2013, pp. 102–103. ISBN: 978-0-19-957493-3.

[47] Sylvain Baillet. "Magnetoencephalography for Brain Electrophysiology and Imaging". In: *Nature Neuroscience* 20.3 (3 Mar. 2017), pp. 327–339. ISSN: 1546-1726. DOI: 10.1038/nn.4504.

[48] Matti Hämäläinen et al. "Magnetoencephalography—Theory, Instrumentation, and Applications to Noninvasive Studies of the Working Human Brain". In: *Reviews of Modern Physics* 65.2 (Apr. 1, 1993), pp. 413–497. DOI: 10.1103/RevModPhys.65.413.

[49] Seppo P. Ahlfors et al. "Sensitivity of MEG and EEG to Source Orientation". In: *Brain Topography* 23.3 (2010), pp. 227–232. ISSN: 0896-0267. DOI: 10.1007/s10548-010-0154-x. pmid: 20640882.

[50] G Tononi, O Sporns, and G M Edelman. "A Measure for Brain Complexity: Relating Functional Segregation and Integration in the Nervous System." In: *Proceedings of the National Academy of Sciences* 91.11 (May 24, 1994), pp. 5033–5037. DOI: 10.1073/pnas.91.11.5033.

[51] K J Friston et al. “Functional Connectivity: The Principal-Component Analysis of Large (PET) Data Sets”. In: *Journal of cerebral blood flow and metabolism* 13.1 (Jan. 1, 1993), pp. 5–14. ISSN: 1559-7016. DOI: 10.1038/jcbfm.1993.4. pmid: 8417010.

[52] Karl J. Friston. “Functional and Effective Connectivity in Neuroimaging: A Synthesis”. In: *Human Brain Mapping* 2.1-2 (1994), pp. 56–78. ISSN: 1097-0193. DOI: 10.1002/hbm.460020107.

[53] Francisco Varela et al. “The Brainweb: Phase Synchronization and Large-Scale Integration”. In: *Nature Reviews Neuroscience* 2.4 (4 Apr. 2001), pp. 229–239. ISSN: 1471-0048. DOI: 10.1038/35067550.

[54] R. D. Pascual-Marqui, C. M. Michel, and D. Lehmann. “Low Resolution Electromagnetic Tomography: A New Method for Localizing Electrical Activity in the Brain”. In: *International Journal of Psychophysiology* 18.1 (Oct. 1, 1994), pp. 49–65. ISSN: 0167-8760. DOI: 10.1016/0167-8760(84)90014-X.

[55] Joseph HA Guillaume et al. “Introductory Overview of Identifiability Analysis: A Guide to Evaluating Whether You Have the Right Type of Data for Your Modeling Purpose”. In: *Environmental Modelling & Software* 119 (2019), pp. 418–432.

[56] B.D. Van Veen et al. “Localization of Brain Electrical Activity via Linearly Constrained Minimum Variance Spatial Filtering”. In: *IEEE Transactions on Biomedical Engineering* 44.9 (Sept. 1997), pp. 867–880. ISSN: 1558-2531. DOI: 10.1109/10.623056.

[57] B.D. Van Veen and K.M. Buckley. “Beamforming: A Versatile Approach to Spatial Filtering”. In: *IEEE ASSP Magazine* 5.2 (Apr. 1988), pp. 4–24. ISSN: 1558-1284. DOI: 10.1109/53.665.

[58] M. X. Huang, J. C. Mosher, and R. M. Leahy. “A Sensor-Weighted Overlapping-Sphere Head Model and Exhaustive Head Model Comparison for MEG”. In: *Physics in Medicine & Biology* 44.2 (Feb. 1999), p. 423. ISSN: 0031-9155. DOI: 10.1088/0031-9155/44/2/010.

[59] J Sarvas. “Basic Mathematical and Electromagnetic Concepts of the Biomagnetic Inverse Problem”. In: *Physics in Medicine and Biology* 32.1 (Jan. 1, 1987), pp. 11–22. ISSN: 0031-9155, 1361-6560. DOI: 10.1088/0031-9155/32/1/004.

[60] G.L. Colclough et al. “A Symmetric Multivariate Leakage Correction for MEG Connectomes”. In: *NeuroImage* 117 (Aug. 2015), pp. 439–448. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2015.03.071.

[61] Sreevalsan S. Menon and K. Krishnamurthy. “A Comparison of Static and Dynamic Functional Connectivities for Identifying Subjects and Biological Sex Using Intrinsic Individual Brain Connectivity”. In: *Scientific Reports* 9.1 (1 Apr. 5, 2019), p. 5729. ISSN: 2045-2322. DOI: 10.1038/s41598-019-42090-4.

[62] R. Matthew Hutchison et al. “Dynamic Functional Connectivity: Promise, Issues, and Interpretations”. In: *NeuroImage*. Mapping the Connectome 80 (Oct. 15, 2013), pp. 360–378. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2013.05.079.

[63] Catie Chang and Gary H. Glover. “Time–Frequency Dynamics of Resting-State Brain Connectivity Measured with fMRI”. In: *NeuroImage* 50.1 (Mar. 1, 2010), pp. 81–98. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2009.12.011.

[64] Maria Giulia Preti, Thomas AW Bolton, and Dimitri Van De Ville. "The Dynamic Functional Connectome: State-of-the-art and Perspectives". In: *NeuroImage*. Functional Architecture of the Brain 160 (Oct. 15, 2017), pp. 41–54. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2016.12.061.

[65] C.F. Beckmann and S.M. Smith. "Probabilistic Independent Component Analysis for Functional Magnetic Resonance Imaging". In: *IEEE Transactions on Medical Imaging* 23.2 (Feb. 2004), pp. 137–152. ISSN: 0278-0062. DOI: 10.1109/TMI.2003.822821.

[66] Stephen M. Smith et al. "Temporally-Independent Functional Modes of Spontaneous Brain Activity". In: *Proceedings of the National Academy of Sciences* 109.8 (Feb. 21, 2012), pp. 3131–3136. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1121329109.

[67] Danilo Bzdok, Naomi Altman, and Martin Krzywinski. "Statistics versus Machine Learning". In: *Nature Methods* 15.4 (4 Apr. 1, 2018), pp. 233–234. ISSN: 1548-7105. DOI: 10.1038/nmeth.4642.

[68] Y. Ephraim and N. Merhav. "Hidden Markov Processes". In: *IEEE Transactions on Information Theory* 48.6 (June 2002), pp. 1518–1569. ISSN: 1557-9654. DOI: 10.1109/TIT.2002.1003838.

[69] "Markov Models". In: Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer, 2006, pp. 607–610. ISBN: 978-0-387-31073-2.

[70] "Hidden Markov Models". In: Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer, 2006, pp. 610–635. ISBN: 978-0-387-31073-2.

[71] Harini Eavani et al. "Unsupervised Learning of Functional Network Dynamics in Resting State fMRI". In: *Information Processing in Medical Imaging*. Ed. by James C. Gee et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 426–437. ISBN: 978-3-642-38868-2. DOI: 10.1007/978-3-642-38868-2_36.

[72] Gabriel G. Infante López, Holger Hermanns, and Joost-Pieter Katoen. "Beyond Memoryless Distributions: Model Checking Semi-Markov Chains". In: *Process Algebra and Probabilistic Methods. Performance Modelling and Verification*. Ed. by Luca de Alfaro and Stephen Gilmore. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, pp. 57–70. ISBN: 978-3-540-44804-4. DOI: 10.1007/3-540-44804-7_4.

[73] Nelson J. Trujillo-Barreto, David Araya, and Wael El-Deredy. *The Discrete Logic of the Brain - Explicit Modelling of Brain State Durations in EEG and MEG*. May 10, 2019. DOI: 10.1101/635300. URL: https://www.biorxiv.org/content/10.1101/635300v1 (visited on 01/06/2024). preprint.

[74] Petra Ritter et al. "State-Dependencies of Learning across Brain Scales". In: *Frontiers in Computational Neuroscience* 9 (2015). ISSN: 1662-5188.

[75] Jürgen Schmidhuber. "Deep Learning in Neural Networks: An Overview". In: *Neural Networks* 61 (Jan. 1, 2015), pp. 85–117. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2014.09.003.

[76] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks Are Universal Approximators". In: *Neural Networks* 2.5 (Jan. 1, 1989), pp. 359–366. ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90020-8.

[77] Yann LeCun et al. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Advances in Neural Information Processing Systems*. Ed. by D. Touretzky. Vol. 2. Morgan-Kaufmann, 1989.

[78] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (7553 May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

[79] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *Nature* 323.6088 (6088 Oct. 1986), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0.

[80] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[81] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. Dec. 10, 2022. arXiv: 1312.6114 [cs, stat]. URL: http://arxiv.org/abs/1312.6114 (visited on 08/06/2023). preprint.

[82] Karl Friston, James Kilner, and Lee Harrison. "A Free Energy Principle for the Brain". In: *Journal of Physiology-Paris*. Theoretical and Computational Neuroscience: Understanding Brain Functions 100.1 (July 1, 2006), pp. 70–87. ISSN: 0928-4257. DOI: 10.1016/j.jphysparis.2006.10.001.

[83] Guido Van Rossum. *The Python Library Reference, Release 3.9.17*. Python Software Foundation, 2023.

[84] TensorFlow Developers. *TensorFlow*. Version v2.9.0. Zenodo, May 16, 2022. DOI: 10.5281/zenodo.6555127.

[85] Charles R. Harris et al. "Array Programming with NumPy". In: *Nature* 585.7825 (7825 Sept. 2020), pp. 357–362. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2.

[86] Martín Abadi et al. *TensorFlow: Large-scale Machine Learning on Heterogeneous Systems*. 2015.

[87] Chetan Gohil et al. "Mixtures of Large-Scale Dynamic Functional Brain Network Modes". In: *NeuroImage* 263 (Nov. 1, 2022), p. 119595. ISSN: 1053-8119. DOI: 10.1016/j.neuroimage.2022.119595.

[88] Benjamin A E Hunt et al. "Attenuated Post-Movement Beta Rebound Associated With Schizotypal Features in Healthy People". In: *Schizophrenia Bulletin* 45.4 (June 18, 2019), pp. 883–891. ISSN: 0586-7614. DOI: 10.1093/schbul/sby117.

[89] Mark Jenkinson, Mickael Pechaud, Stephen Smith, et al. "BET2: MR-based Estimation of Brain, Skull and Scalp Surfaces". In: *Eleventh Annual Meeting of the Organization for Human Brain Mapping*. Vol. 17. 3. Toronto. 2005, p. 167.

[90] Stephen M. Smith. "Fast Robust Automated Brain Extraction". In: *Human Brain Mapping* 17.3 (2002), pp. 143–155. ISSN: 1097-0193. DOI: 10.1002/hbm.10062.

[91] Zelekha A. Seedat et al. “The Role of Transient Spectral ‘Bursts’ in Functional Connectivity: A Magnetoencephalography Study”. In: *NeuroImage* 209 (Apr. 2020), p. 116537. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2020.116537.

[92] N. Tzourio-Mazoyer et al. “Automated Anatomical Labeling of Activations in SPM Using a Macroscopic Anatomical Parcellation of the MNI MRI Single-Subject Brain”. In: *NeuroImage* 15.1 (Jan. 1, 2002), pp. 273–289. ISSN: 1053-8119. DOI: 10.1006/nimg.2001.0978.

[93] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (May 13, 1983), pp. 671–680. DOI: 10.1126/science.220.4598.671.

[94] Peng Tian, Huanchen Wang, and Dongme Zhang. “Nonlinear Integer Programming by Simulated Annealing”. In: *IFAC Proceedings Volumes*. 7th IFAC Symposium on Large Scale Systems: Theory and Applications 1995, London, UK, 11-13 July, 1995 28.10 (July 1, 1995), pp. 629–633. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)51590-6.

[95] R. Timms. “Time-Varying Source Reconstruction”. http://purl.org/dc/dcmitype/Text. University of Oxford, 2022.