Institute of Fundamental Technological Research,
Polish Academy of Sciences, Warsaw

Doctoral Dissertation

Locally specific human brain dynamics automatically modeled using spectral features of MEG/EEG signals

Michał Konrad Komorowski
Supervisor: Prof. dr hab. Włodzisław Duch

Department of Informatics, Institute of Engineering and Technology,
Faculty of Physics, Astronomy and Informatics,
Nicolaus Copernicus University

Toruń, 2022

I dedicate this thesis to my Family.

Declaration

I hereby declare that except for a specific reference made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification at this or any other university. This dissertation is my own work and does not contain anything which would be the outcome of work done in collaboration with others, except for what is specified in the text and the Acknowledgements.

Michał Konrad Komorowski
June 2022

Acknowledgments

I wish to thank my Supervisor, Prof. Włodzisław Duch, for his broad support and advice during my Ph.D. studies and research. For the longtime mentorship, I am grateful to Dr. Tomasz Piotrowski and Dr. Joanna Dreszer.

My deepest gratitude goes to my mother Ewa, father Zbigniew, and my grandparents Wanda and Kazimierz, for supporting me in every possible way while being extremely patient. Your contribution is beyond words. I want to thank my brother, Mariusz, for simply being the best. I am deeply indebted to Dr. Karolina Finc and Mirosław Manelski for living the values that inspire me to be a better scientist and giving me the courage to be the best version of myself. Thank you from my heart. I owe my well-being to Ewa Ratajczak and Marcin Hajnowski — my virtual siblings who balance my serious work with their flawless wit. I thank Karol Dziankowski for being a demanding but supportive teacher and friend.

Much appreciation goes to Dr. Anne Keitel for invaluable guidance in understanding Spectral Fingerprinting, and to Dr. Krzysztof Rykaczewski for massively improving the mathematical notation of the method algorithms. I am also grateful to Dr. Katarzyna Jurewicz and Jakub Wojciechowski for coauthoring the currently prepared articles. For giving me the stimulating habitat for my scientific growth at the Centre for Modern Interdisciplinary Technologies as well as at the Department of Informatics, I would like to thank Prof. Jerzy Łukaszewicz, Prof. Wiesław Nowak, Prof. Rafał Adamczak, and the Authorities and Administration of Faculty of Physics, Astronomy, and Informatics, Nicolaus Copernicus University in Toruń. For excellent training opportunities, I thank Prof. Andrzej Cichocki and Prof. Tomasz Wolak.

Finally, any endeavor of this kind would not possible without the support of those great people:

Paweł Balasiński,
Dr. Bibianna Bałaj,
Paweł Bieżuński,
Jędrzej Brajer,
Małgorzata Brzóskiewicz,
Przemysław Gulczyński,
Bartłomiej Kiljanek,
Paweł Jakub Klimek,
Aleksandra Kozupa,
Szymon Kozupa,
Mateusz Kubiak,
Aleksandra Kulesz,
Krystian Kulesz,
Agnieszka Łoza,
John Paul,
Natalia Pawlaczyk,
Natalia Sierakowska,
Dominik Stańczak,
Marcin Surga,
Magdalena Szmytke,
Agnieszka Szymańska,
Mateusz Warda.

Thank you for taking part in this long journey.

I would also like to thank for the following financial support:

- SYMFONIA 4 grant (UMO-2016/20/W/NZ4/00354),
  National Science Center, Poland.
- "Granty dla Młodych" (1037-F/2017) funded by
  Faculty of Physics, Astronomy, and Informatics,
  Nicolaus Copernicus University, Poland
- University Centre of Excellence
  "Dynamics, Mathematical Analysis and Artificial Intelligence"
  (IDUB CD DAMSI, 9001960020/2021/0971)

Abstract

The human brain is a dynamic system whose oscillatory activity has been investigated for many decades, which has led to a better understanding of the intricate mosaic of our experience. The concept of *brain fingerprint* has many definitions. In this dissertation, I refer to it as to the set of brain activity features (derived from measured signals) that allow for identification of brain structures: regions or networks, involved in particular cognitive process.

I have focused specifically on one of the approaches based on frequency analysis — the *Spectral Fingerprinting* method. In its original form proposed by Keitel and Gross in 2016, it allows modeling the activity of local cortical and subcortical structures, delineated using an anatomical atlas of the human brain, based on resting multichannel signals measured with a magnetoencephalograph, as sets of spectral curves that uniquely identify each brain region. At the time, the method, while promising, was inapplicable to the exploration of other datasets than the one provided by the authors, and the option to adjust modeling parameters was missing.

My dissertation aims to fill this gap by offering a new implementation of the Spectral Fingerprinting in the form of an open-source modular toolbox, developed in MATLAB, called the **Toolbox for Frequency-based Fingerprinting (ToFFi)**. It allows automatic construction of spectral markers of brain activity, characteristic for a selected atlas of brain areas, at the level of individual subjects or a selected group of subjects, based on modeling parameters adjusted by the user. The presented software enables parallel computations, supports reproducibility of the obtained results, and facilitates interpretation of the output data using a visualization module.

The dissertation begins by introducing the concept of oscillation. Then, the neuron and its basic mechanisms leading to the oscillatory activity of the human brain are presented. Magnetoencephalography and electroencephalography are discussed as two main modalities used in the author's publications. The Spectral Fingerprinting method is then presented against the background of contemporary studies in the domain, after which the first results obtained by Keitel and Gross are compared with the results of other researchers, including authorial results as well. The central part of the thesis is the discussion of the ToFFi Toolbox architecture, complemented by the complexity analysis of implemented algorithms, as well as considerations of parallel computing and reproducibility of results. In addition, the thesis includes paragraphs briefly demonstrating the operation of the program. The application of the toolbox to the study of the brain resting-state networks is illustrated in the final chapter of the dissertation.

The content of the appendices includes a comprehensive mathematical description of the implemented algorithms, the original Spectral Fingerprinting code proposed by Keitel and Gross in 2016, and a guide to the ToFFi Toolbox repository, where interested readers will find the information needed to use the toolbox in their own research.

Streszczenie

Mózg człowieka stanowi układ dynamiczny, którego oscylacyjna aktywność badana jest od wielu dekad, prowadząc do coraz lepszego rozumienia skomplikowanej mozaiki naszych doświadczeń. Koncepcję *brain fingerprint* określa się na wiele sposobów. W niniejszej rozprawie, definiuje to pojęcie jako zbiór cech aktywności mózgu (uzyskany z mierzonych sygnałów), pozwalający na identyfikację struktur mózgowych: obszarów lub sieci, zaangażowanych w wybrany proces poznawczy.

Przyjrzałem się szczególnie jednemu z podejść bazujących na analizie częstotliwości — metodzie *Spectral Fingerprinting*. W jej oryginalnym wydaniu zaproponowanym przez Keitel i Grossa w 2016 roku, pozwala ona na modelowanie aktywności lokalnych struktur korowych i podkorowych, wyznaczonych za pomocą anatomicznego atlasu ludzkiego mózgu, w oparciu o spoczynkowy wielokanałowy sygnał mierzony magnetoencefalografem, jako jednoznacznie identyfikowalne zbiory krzywych spektralnych, opisujące każdy badany obszar mózgu z osobna. Metoda w ówczesnej postaci, choć obiecująca, była niemożliwa do zastosowania w eksploracji innych niż udostępniony przez autorów zbiór danych oraz pozbawiona opcji dostosowywania parametrów modelowania.

Moja rozprawa ma na celu wypełnienie tej luki poprzez zaoferowanie nowej implementacji metody Spectral Fingerprinting w postaci otwartego modularnego pakietu narzędziowego opracowanego w języku MATLAB, występującego pod nazwą the **Toolbox for Frequency-based Fingerprinting (ToFFi)**. Pozwala on na automatyczne konstruowanie spektralnych markerów aktywności mózgowej charakterystycznych dla wybranego atlasu obszarów mózgu na poziomie indywidualnych osób badanych bądź wybranej grupy osób w oparciu o parametry modelowania wybrane przez użytkownika. Przedstawione oprogramowanie umożliwia obliczenia równoległe, wspiera reprodukowalność uzyskiwanych wyników, oraz ułatwia interpretację danych wyjściowych za pomocą modułu do wizualizacji.

Rozprawa rozpoczyna się od wprowadzenia pojęcia oscylacji. Kolejno, przedstawiany jest neuron i podstawowe mechanizmy jego działania prowadzące do wytwarzania oscylacyjnej aktywności ludzkiego mózgu. Omówione zostają magnetoencefalografia oraz elektroencefalografia jako dwie główne modalności wykorzystane w autorskich publikacjach. Metoda Spectral Fingerprinting jest następnie przedstawiona na tle współczesnych studiów o podobnej tematyce, po czym pierwsze wyniki uzyskane z jej pomocą są zestawione z rezultatami innych badaczy, w tym autora niniejszej rozprawy. Główną część pracy stanowi omówienie architektury pakietu ToFFi, uzupełnione o analizę złożoności wykorzystywanych algorytmów oraz rozważania na temat obliczeń równoległych i reprodukowalności wyników. Dodatkowo załączono paragrafy demonstrujące pokrótce działanie programu. Zastosowanie pakietu narzędziowego w badaniu właściwości sieci spoczynkowych ludzkiego mózgu ilustruje ostatni rozdział rozprawy.

Treść dodatków stanowi: wyczerpujące opracowanie matematyczne zastosowanych algorytmów; pierwotny kod metody zaproponowanej przez Keitel i Grossa, oraz opis repozytorium otwartego kodu, gdzie zainteresowany czytelnik znajdzie informacje potrzebne by wykorzystać niniejszy pakiet narzędziowy we własnych badaniach.

Table of contents

List of Tables

List of Figures

Listings

Nomenclature

Acronyms / Abbreviations

AAL Automatic Anatomical Labeling

ADC Analog-to-Digital Converter

BOLD Blood-Oxygenation-Level-Dependent

BSS Blind Source Separation

CFC Cross-Frequency Coupling

CPU Central Processing Unit

CV Cross-Validation

DAN Dorsal Attention Network

DICS Dynamic Imaging of Coherent Sources

DMN Default Mode Network

DPSS Discrete Prolate Spheroidal Sequences

ECoG Electrocorticography

EEG Electroencephalography

EM Expectation-Maximization

EMD Earth Mover's Distance

EOG Electrooculography

EPSP Excitatory Postsynaptic Potential

ERD Event-Related Desynchronization

ERP Event-Related Potential

ERS Event-Related Synchronization

EVD Eigenvalue Decomposition

EXE Executive Control Network

FFT Fast Fourier Transform

FID Free Induction Decay

fMRI Functional Magnetic Resonance Imaging

FOI Frequencies of Interest

FPN Frontoparietal Network

GM Gaussian Mixture

GMM Gaussian Mixture Model

HCP Human Connectome Project

hdEEG High-density Electroencephalography

HPI Head Position Indicator

ICA Independent Component Analysis

iEEG Intracranial Electroencephalography

IF Individual Fingerprint

IPSP Inhibitory Postsynaptic Potential

LCMV Linearly Constrained Minimum Variance

LIM Limbic Network

LORETA Low-Resolution Brain Electromagnetic Tomography

MEG Magnetoencephalography

MNE Minimum Norm Estimate

MNI Montreal Neurological Institute

MRI Magnetic Resonance Imaging

MTMFFT Multi-Taper Method for Fast Fourier Transform

MV-PURE Minimum-Variance Pseudo-Unbiased REduced-rank

NIRS Near-Infrared Spectroscopy

NLL Negative Log-Likelihood

OPM Optically Pumped Magnetometer

RAM Random Access Memory

RF Radiofrequency

RNG Random Number Generator

ROI Region of Interest

RSN Resting-State Network

SAL Salience Network

SB/CB Subcortical/Cerebellum Network

SE Spin-Echo

SF Spectral Fingerprint

SMN Somatomotor Network

SNR Signal-to-Noise Ratio

SQUID Superconducting Quantum Interference Device

SS Sum of Squares

SVD Singular Value Decomposition

ToFFi Toolbox for Frequency-based Fingerprinting

ULF Ultra-Low Field

UPGMA Unweighted Pair Group Method with Arithmetic Mean

VIS Visual Network

YIGM Yttrium-Iron Garnet Films

Introduction

Seeking to understand rhythms in nature led us to many discoveries that improved our lives, starting from the knowledge of how to rhythmically swing our legs to be able to walk, the comprehension of patterns of season and weather to know when to sow and when to harvest, ending on noticing the life cycle of our organizations, companies, and schools — rhythms of all the abstract systems we invent to take control of our evolution [1].

The enormously complicated structure of the human brain defines its function, which in turn shapes our perception of everyday experience. The myriads of neural networks create complex activity patterns that repeat over time. The neuron and the oscillation are the fundamental building blocks neuroscience has studied for decades [2].

The ongoing progress in modern neuroscience can be attributed largely to neuroinformatics [3]. The field advances by proposing inventions for a broad range of topics, from theoretical and computational modeling of various cognitive systems (like vision) or processes (e.g., mental rotation), designing new applications for hardware that mimics the behavior of neurobiological architectures (neuromorphic chips), providing solutions in medicine (e.g., paralysis rehabilitation system utilizing neuroimaging; epileptic source localization), and finding applications for deep learning [4][5]. But why do we care so much?

Brain diseases affect 180 billion Europeans — over one fourth of the people in our continent. The size of the resulting economic burden is about 800 billion euros per year [6]. Progress in understanding the specificity of neural rhythms, in the context of these numbers, is the vital quest for modern research because harnessing brain oscillations could be one day a game-changer in medical diagnosis and treatment [7].

Brain fingerprinting is a relatively young concept used in many ways in the literature [138][172][9][173]. In this thesis I use the meaning related to finding and interpreting meaningful signal patterns correlated with specific cognitive processes and linking them to a given brain structure or network. Many reports confirm that individual human brain areas have characteristic oscillatory signatures that can be modulated under changing external conditions. In particular, Donner and Siegel attempted to define so-called *spectral fingerprints* of sets of regions acting in concert (brain networks) [8], however, the method invented by Keitel and Gross was the first that addressed the issue of characterizing individual brain areas and testing it directly [9]. Following their definition of brain (spectral) fingerprints, I focus on characterizing particular brain

regions.

In 2016, *Spectral Fingerprinting* existed only as a limited beta-version script, seemingly simple and intuitive, however at a closer look, it turned out that deep understanding of the code and some programming skills are expected from the user to adjust the program for custom applications. However, not every scientist is a skilled programmer at the same time, which limits the pool of potential beneficients of the method. There were no open-source, versatile tools to calculate spectral fingerprints systematically and efficiently, until now. The goal of this thesis was to fill the gap.

I have addressed the problem by creating modular, highly-configurable MATLAB **Toolbox for Frequency-based Fingerprinting** (ToFFi[†]) [10]. It can transform MEG/EEG signals into unique spectral representations of brain areas defined by anatomical or functional division of choice. The toolbox design supports reproducibility and parallel computations to solve research problems related to finding characteristic signatures of the human brain oscillatory activity. The content of this dissertation explains the background behind the toolbox, present its structure and capabilities, and shows how it can be used in neuroscience research.

Chapter 1 starts with the concept of *oscillation*, and further introduces the basics of brain anatomy and function. The link between physiology and physics is made to facilitate understanding of modern brain imaging methods: *magnetoencephalography (MEG)*, *electroencephalography (EEG)*, and *functional magnetic resonance imaging (fMRI)*.

Chapter 2 reviews the studies of characteristic patterns found in oscillations. Exciting possibilities that brain fingerprinting offers are illustrated by the review of literature on source analysis, graph-theoretical methods, and spectral approaches focused on finding useful oscillatory patterns. Notably, this part of the thesis sets the stage for Spectral Fingerprinting.

Chapter 3 focuses on the main method being employed in my research — Spectral Fingerprinting. The background, methods, and results of two main works utilizing the approach are discussed: the pioneering article by Keitel and Gross, 2016, based on MEG resting-state data [9], and the work of Lubinus et al., 2019, exploring differences between spectral fingerprints of healthy and congenitally blinded participants [11]. I also present the results published in the study of Komorowski, et al., 2018 [12], which I conducted to (1) reproduce the area identification accuracy reported by Keitel and Gross, and (2) to test specificity of spectral representations constructed from other data by replicating the procedure on EEG recordings.

Chapter 4 describes in depth the architecture and functionalities of the Toolbox for

[†]I think the toolbox's name is kind of a "happy accident", as it summarizes both its domain of application and the lenght of its development. The verdict is left to the reader's taste.

Frequency-based Fingerprinting. Employed algorithms were analyzed from a perspective of *time complexity* and computational *speedup* when implemented concurrency feature is used. Numerous flowcharts, listings, pseudocodes, and references to the appendix containing mathematical formulae complete the description. ToFFi toolbox is also a part of open-software publication in Neurocomputing journal, currently under review. The preprint is available for reading at arXiv repository [10].

Finally, **Chapter 5** illustrates how Spectral Fingerprinting and ToFFi Toolbox can aid solving real-life scientific problems by presenting an exploratory study in progress about the hierarchy of oscillations across the cortex organized by the *resting-state brain networks (RSNs)* framework. Discovered relations between fMRI-based cortical principal gradients [13] and constructed spectral components provide valuable insight for analyses planned in the future.

The content of **Appendix A** includes the comprehensive mathematical description of the implemented algorithms. The original Spectral Fingerprinting code proposed by Keitel and Gross in 2016 can be found in **Appendix B**. Interested readers seeking for information enabling usage of the ToFFi Toolbox's code in their own research are referred to **Appendix C**.

This thesis, although inspired by neuroscience, is heavily grounded in computer science. Concepts related to brain function and brain imaging are introduced only to the extent sufficient to provide a necessary context. For the most part, my work describes in-depth the toolbox: its architecture, performance, and utilized algorithms.

Considering all the above, I hope that this dissertation will catalyze the progress towards new discoveries in brain science and neuroinformatics. The ground of brain fingerprinting is fertile and already under cultivation, promising tools that address current burning demands — faster and more accurate medical diagnosis, comprehending cognition and consciousness, regulating learning and emotions. The Toolbox for Frequency-based Fingerprinting combines the modern knowledge of the human brain functions, signal processing, and software development with the hope of contributing to this ambitious goal. Along the way, we, experts from different branches of science, need to cross our boundaries and synchronize interdisciplinary effort to understand problems of the day, invent creative solutions, and, ultimately, bring them to life.

1 Understanding Human Brain Oscillations

Periodicity characterizes animate and inanimate beings. Periodicity of seasons, day and night, music and dance, death and life. People revere periodicity by engaging in invented rituals and by adapting culture to the recurring echo of Nature. Recognizing rhythms made humans understand that something was, something is, and something will be — the arrow of time runs unmoved in only one direction, from the past to the present, which is immediately abandoned in favor of the future. Time is the only resource we cannot produce or recover. We cannot shape time. We can only "happen" in time. The cause and effect relationship exists because there is a temporal dimension.

The absolute time axis contains events — single reference points between which time intervals are measured (*durations*). In a repetitive process, we distinguish a single *oscillation* by designating two corresponding points — the beginning and the end of a cycle. An oscillation is an ordered sequence of events that occurs between the beginning and the end and then repeats itself. The amount of time needed to complete one cycle is called the *period*. Different processes have different oscillation periods. It is helpful to ask how many oscillations (n) will fit into a given interval of time (T). Then, the concept of *frequency* emerges, conventionally denoted by the letter f, expressed as:

$$f = \frac{n}{T} \quad \Big[\frac{1}{s} = \mathrm{Hz}\Big].$$

Fig. 1.1 shows how different types of oscillations can have the same frequency. Frequency value need not be an integer, of course. Frequency is often expressed as *angular frequency*:

$$\omega = 2\pi f = \frac{2\pi}{T} \quad \Big[\frac{\mathrm{rad}}{s}\Big].$$

The equations above show the inverse relationship between time and frequency. In the context of brain science, it is believed that the hierarchical organization of oscillations at multiple *time scales* might constitute the cornerstone of the neural syntax [14][15].

Complex oscillatory patterns are mixtures of variation at different time scales that can be disentangled by measuring the proportion between basic frequencies present in the data. In 1807, the mathematician and engineer Joseph Fourier introduced a method for representing any periodic signal $x(t)$ as the sum of sines and cosines, possibly the most natural depiction of an oscillation [16]. *Trigonometric Fourier series* is defined as follows [17]:

$$x(t) = a_0 + \sum_{k=1}^{\infty} \left(a_k \cos k\omega t + b_k \sin k\omega t\right), \tag{1.1}$$

where:

$$a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)\mathrm{d}t,$$

$$a_k = \frac{2}{T}\int_{t_0}^{t_0+T} x(t)\cos k\omega t \ \mathrm{d}t,$$

$$b_k = \frac{2}{T}\int_{t_0}^{t_0+T} x(t)\sin k\omega t \ \mathrm{d}t.$$

Eq. 1.1 opened up the possibility of studying periodic phenomena using *spectral* methods. An example of such a method can be found in Algorithm 1, discussing the mathematics of the Spectral Fingerprinting method.

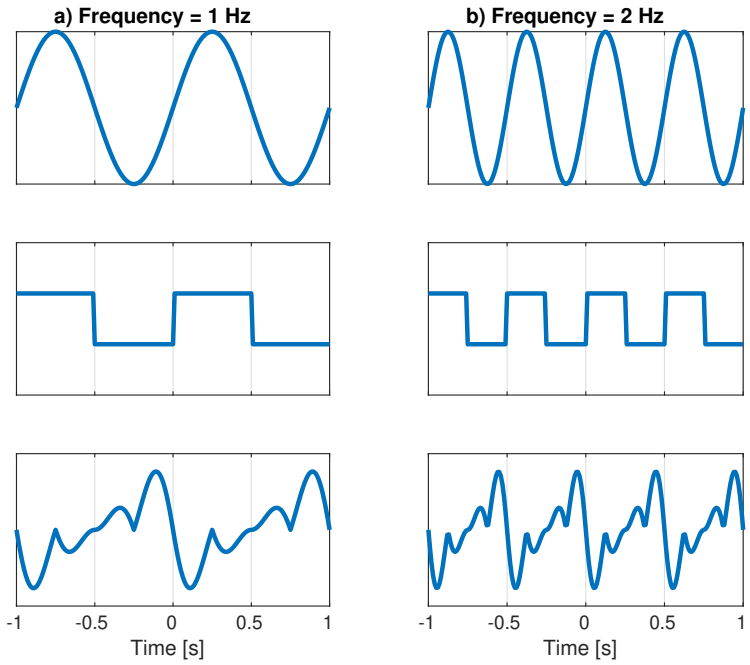


Fig. 1.1. Different examples of oscillations sharing the same frequency — **(a)** $f = 1\,\mathrm{Hz}$; **(b)** $f = 2\,\mathrm{Hz}$.

The concept of frequency is closely related to the concept of *synchrony*. One of the definitions of the word "synchronous" that The American Heritage Dictionary gives is *"having identical periods"*[1]. However, the possession of the same period by two objects A and B exhibiting synchronicity does not immediately imply that they are linked or anyhow related since the alleged relationship may be pure coincidence (Fig. 1.2a). In addition to coincidental synchrony, there are at least three more cases when synchrony occurs. *Reciprocal synchrony* occurs when objects A and B are mutually influenced by

[1]https://www.ahdictionary.com/word/search.html?q=synchronous ; access: 1.03.2022

each other (Fig. 1.2b). When only one object affects the other (e.g., A affects B), we have the *driven synchrony* (Fig. 1.2c). *Induced synchrony* happens when A and B are synchronized by some external energy flow, e.g., the third object that drives both A and B (Fig. 1.2d).

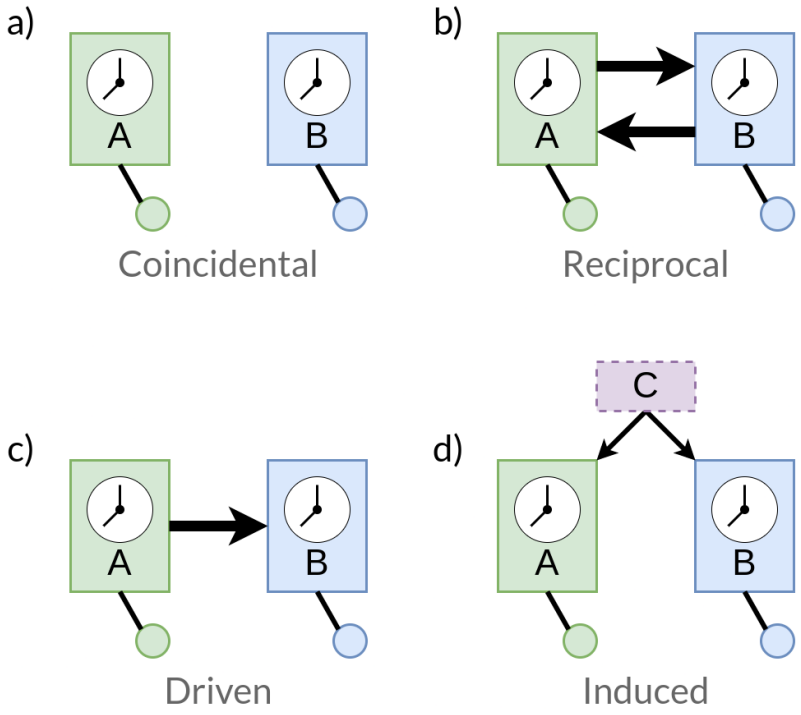


Fig. 1.2. Four types of synchrony. See text for details. Based on: [18].

1.1 Neuron - the oscillating "atom"

If one ever asks what is the primary building block of the nervous system and the brain itself, the word "neurons" came to mind immediately, but there are basically two types of brain cells: *glia* and *neurons*.

Glia (also called *neuroglia*) come in several types and serve different functions. They are not the main actors of the systems, but without them, the whole system would collapse. Glia can shield neurons from external interference [19]. They can produce their own molecules and can buffer signal transmission chemicals (*neurotransmitters*) released by neurons to free them at some other time, which in turn results in synchronizing activity with neighboring neural cells [20][21]. Other vital functions of glia are: dilating blood vessels to enhance the nutrition of active brain regions [22][23], removing dead or damaged neurons, viruses, and fungi from the brain (*microglia*; [24][25]), insulation of the neural axons (*oligodendrocytes* and *Schwann cells*; [26]), helping to place neurons in

desired location during development (*radial glia*; [27]).

In this section, however, we will focus on neurons (Fig. 1.3), which are the most important building block of the whole nervous system. Each one of approximately 100 billion neurons [28] is an individuated discrete unit (as discovered by Santiago Ramón y Cajal in the late 1800s) that sends electrical impulses to other neurons in an organized way. Networks of neurons connect to all major organs in the human body to control their behavior.

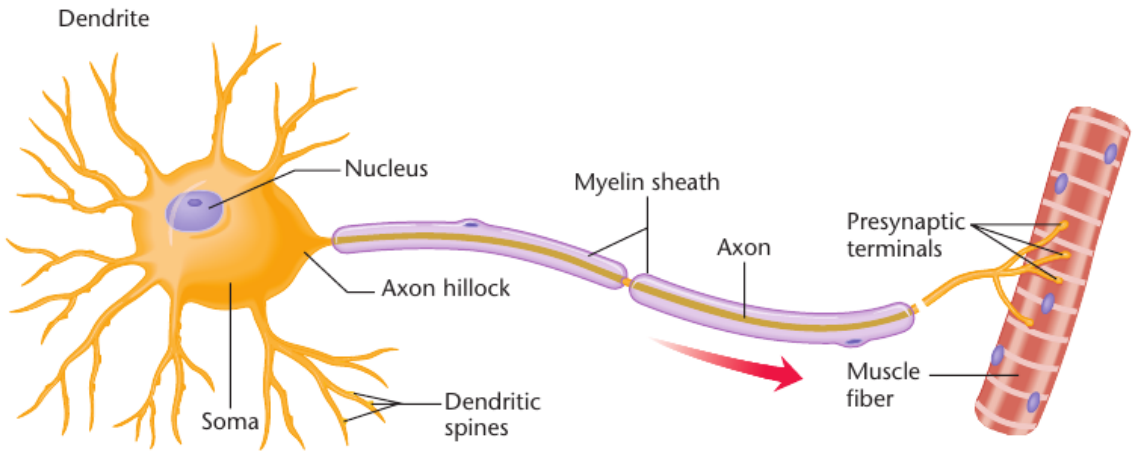


Fig. 1.3. A neuron. Source: [29].

Neurons are built mainly from the elements typical to most animal cells. The surface of a cell is its *membrane* equipped with protein channels that control the inflow and the outflow of water, oxygen, ions (sodium, potassium, calcium, chloride), and other molecules. Among structures typical for an animal cell present in the neuron is the *nucleus*, wherein *chromosomes* store genetic material, *mitochondria* provide the cell with the energy needed, *ribosomes* synthesize new proteins.

What differentiates neurons from other cells is their distinctive shape. The *soma* is the "star-shaped" body of the cell, containing the *nucleus*. The soma's branching outgrowths, lined with specialized *synaptic receptors*, are called *dendrites*. A single specific relatively long fiber is called the *axon*, which can branch at the end. Dendrites come in various shapes and numbers, but there is always only one axon per neuron that has many branches. It conveys the impulse toward other neurons, organs, or muscles. The end of each branch has a swelling called a *presynaptic terminal* (also known as *end bulb* or *bouton*). The end of one neuron and the other neuron do not touch each other[2] — the region encompassing the presynaptic terminal and the synaptic receptor on the receiving neuron is called the *synapse*. Whenever we have at least two neurons

[2]In the case of *electrical synapses*, contrary to *chemical synapses*, there is a physical connection between two neurons.

connected via a synapse, the neuron that delivers transmission is called the *presynaptic neuron*, and the one that receives it is the *postsynaptic neuron*. At the presynaptic terminal, the axon releases neurotransmitters that cross through the space called the *synaptic cleft* towards another cell, inducing changes in the flow of ions to and from the interior of the neuron, thus passing the information forward. Details of the neural transmission will be discussed in the next section.

Some axons may be covered with the myelin sheath with interruptions called *nodes of Ranvier*. They speed up the transfer of the electric signal, utilizing the saltatory conduction phenomenon [30][31][32]. The longest axons of the human body are those from the spinal cord to feet, having more than a meter in length. The smallest neurons may have no axon at all or have it only in residual form. In that case, such neuron (called a *local anaxonic neuron*) transmits a signal to the neighboring neurons only using dendrites via smaller depolarizations called *graded potentials*, not action potentials. Different potentials and their mechanisms are covered in the upcoming paragraphs. Some neurons are contained entirely within a single organ (*interneurons*), and some neurons span across two or more organs. *Afferent* (or *sensory*) neurons collect stimuli received by receptors throughout the body, including the skin, eyes, ears, nose, tongue, and pain and other receptors in the internal organs. The central nervous system controls the rest of the body via *efferent* neurons.

Pyramidal neuron. There is one special type of neuron called the *pyramidal neuron* (Fig. 1.4). Its alignment is perpendicular to the brain's cortical surface, and one of its dendrites is very long (called *apical dendrite*), whereas the cell's soma is located deeper into the cortical sheet. The net current from ensembles of pyramidal neurons is strong enough to produce a detectable electromagnetic field that can be measured by EEG (*electroencephalogram*) and MEG (*magnetoencephalogram*) instruments.

1.2 Neural communication mechanisms

Wherever there is communication, some flow of energy is required. Neurons rely heavily on glucose, so they need a steady supply of oxygen and thiamine (vitamin B1) to be able to metabolize glucose in order to get the energy needed.

Evolution has not equipped us with copper wirework inside our bodies for a reason. *"That way, the strength of an impulse would decay rapidly as it traveled. A touch on your shoulder would feel stronger than a touch on your abdomen. Short people would feel their toes more strongly than tall people could - if either could feel their toes at all"* [29]. Luckily, axons, instead of conducting an electrical impulse, regenerate an impulse at each point. The difference between the arrival times of the impulses from

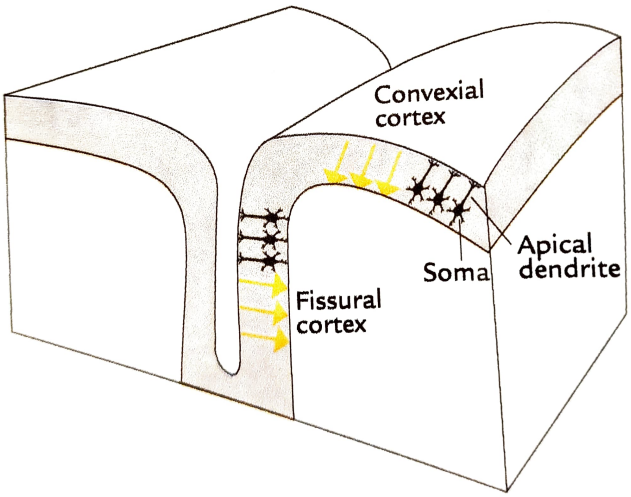


Fig. 1.4. Schematic representation of neurons (black) with the central axis oriented perpendicular to the cortical surface. Excitation of the apical dendrites cause flow of the current, from dendrite to soma or reversely (excitation or inhibition, respectively), which can be modeled as a current dipole (yellow arrow). Source: [33].

different parts of the body is compensated by the brain. Simply, it does not register small differences in arrival time from touch sense receptors. Only recently, in 2021, David Julius and Ardem Patapoutian have won the Nobel Prize award for identifying molecular mechanisms that allow the body's cells to sense temperature and touch.

To understand how our brain acquires information both from the external environment as well as from inside our body, we first need to consider several types of membrane potential (resting potential, action potential, spontaneous potential, postsynaptic potentials) and describe the inner workings of the synapse.

Resting membrane potential. The *membrane potential* is the difference between two electric potentials (*voltage*) that can be measured using two microelectrodes - one on the outside (as the reference microelectrode) and the other inside the neural cell. Messages the brain receives from nervous systems develop from the disturbance of so-called *resting membrane potential*, which is about -70 millivolts (mV). This state results from the presence of sodium and potassium ions inside and outside the neural cell. Outside the membrane, there is an abundance of positively charged sodium ions (Na^+) and relatively fewer potassium ions (K^+). On the other side of the membrane, there are many potassium ions as well. They are surrounded by negatively charged large proteins and chloride ions (Cl^-), which keep the cell's interior electrically negative. As a result, the membrane is *polarized* to its resting potential of -70 mV. The only way to even out this ionic concentration gradient is to move ions through *sodium-potassium pumps* and ionic channels. In resting state, they are held closed, preventing the migration of

charges. Sodium-potassium pumps respond and open to maintain the resting potential default value. An important role of this steady voltage difference is to keep neurons ready to fire up immediately, analogously to the archer pulling the bow in advance to be ready to shoot when needed quickly.

Action potential. Due to external stimuli or stimulation by other neurons, resting potential can be further *hyperpolarized* (i.e., further increase of the negative charge inside the cell or the positive charge outside the cell). The *depolarization* is the opposite event, which results in the increase of the membrane resting potential. If depolarization reaches the threshold value of $-55\ mV$, the *action potential* occurs. The depolarization makes voltage-gated ion channels open for a short time allowing for the fast inflow of the sodium and potassium ions, further depolarizing the membrane. The action potential is possible only in axons because there are no voltage-gated channels in dendrites or soma. Inside the neural body, instead of the default overall negative charge, now the positive charges dominate. At about $+40\ mV$, the action potential reaches its peak. Even at the peak of the action potential, sodium ions continue to be far more concentrated outside the neuron than inside, thus the overall potential outside the cell remains positive throughout the process. Then, the voltage-gated sodium channels close and cannot be opened during some time (called the *refractory period*), even by the very strong stimuli. Sodium stops flowing in. Nevertheless, potassium channels remain open, allowing an escape of the K^+ and Na^+ ions out of the membrane, driven by the force of concentration gradient and the electrical gradient. Membrane potential eventually drops to the value slightly below its resting value because there are still some excessive sodium ions remaining. Sodium-potassium pumps take care of that and restore the membrane potential to its resting value after some longer time, relative to the depolarization time, which lasts in total 1-2 milliseconds. Ignition of the action potential is local, but the movement of sodium ions into the cell during the action potential cause depolarization to spread further along the axon, but only towards its presynaptic terminal, not in the direction of the soma, thanks to the refractory period preventing already triggered but now closed channels to re-open too early. This mechanism of progressive longitudinal depolarization of the axon allows the action potential to be regenerated with the same strength, without weakening along the way, which is why it is better to have neurons than copper wires inside the body. The only drawback of such a solution is that copper wire transmits signals at the speed of hundreds of thousands of meters per second. For comparison, non-myelinated neuron transmits signals with a velocity less than $10\ m/s$, and myelinated one (i.e., equipped with nodes of Ranvier) at about $43\ m/s$ [34]. For thicker axons, the electrical resistance is lower, so the transmission velocity is higher, and additionally, the frequency of action potential subsequent occurrences can

be higher. One of the most important properties of the action potential is that its peak value, $+40\ mV$, and speed are invariant to the stimulus strength - it only needs to be strong enough to trigger the action potential. This is called the *All-or-None Law*. An important question is then raised: considering the constant value of the action potential amplitude, how is it possible to differentiate stronger and weaker stimuli? The more occurrences of action potential in a unit of time, the stronger the given stimulus is perceived.

Some neurons rest peacefully in their resting potentials, but some of them produce a *spontaneous activity* in the form of periodic patterns of action potentials, even without synaptic input.

Activity at synapses. Another class of important potentials is the *postsynaptic potentials* that come in two variants:

- *excitatory postsynaptic potentials (EPSP)*,
- *inhibitory postsynaptic potentials (IPSP)*.

These potentials result from the action potential triggered by the presynaptic neuron, and it is an electrical excitation, converted to the chemical signal and transferred to the postsynaptic neuron through a synapse. EPSP depolarizes postsynaptic neuron (via inflow of positives charge into postsynaptic membrane), and IPSP hyperpolarizes it (via outflow of positives charge into postsynaptic membrane), which lowers and raises the threshold of triggering the action potential, respectively. The character of the postsynaptic potential depends on the kind of neurotransmitter released at the postsynaptic terminal and the kind of synaptic receptor that neurotransmitter latches into. Neurotransmitters are the chemical synthesized either at the presynaptic terminal (e.g., acetylcholine) or inside the nucleus (e.g., endorphins), depending on the size of its molecule. Activated receptors can then trigger ionic channels to open in the postsynaptic neurons encouraging charges to move and produce currents. Certain hormones produced in the thyroid, pituitary gland, or the neurons themselves, secreted into the bloodstream, affect human organism globally, including the neural transmission at synapses.

There are around 160 trillion ($160 \cdot 10^{12}$) synapses in the human body [35]. Their operation can be briefly overviewed as follows (Fig. 1.5):

1. Action potential reaches the presynaptic terminal,
2. Neurotransmitters are released into the space of synaptic cleft,
3. Released neurotransmitter molecules are partially captured by the receptors of the postsynaptic neuron (the rest of them float in the synaptic space),

4. Activated receptors trigger chemical events and potentials in the postsynaptic neuron, which modulates the behavior of this neuron and dependent organs
5. After about 1 ms, captured neurotransmitters are released back to the synaptic cleft,
6. Some floating neurotransmitters are recycled, some are absorbed back to the presynaptic neurons to be used again (this is called the *reuptake*), and some stay in the cleft.

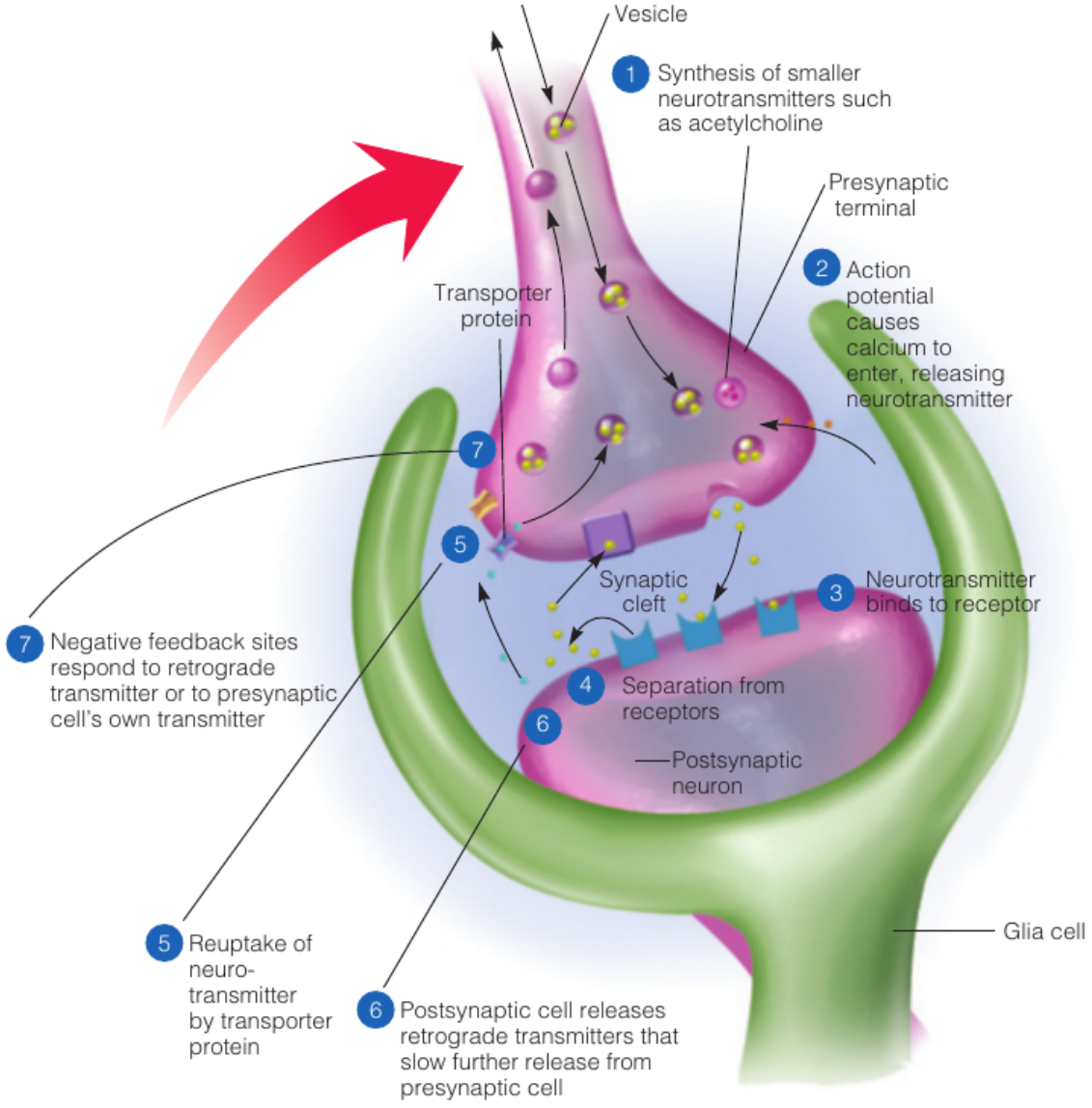


Fig. 1.5. Transmission at a synapse. Source: [29].

Summation of potentials. Each postsynaptic neuron can have many synaptic connections to many presynaptic neurons. Postsynaptic potentials undergo both *spatial*

summation (total excitation/inhibition of the postsynaptic neuron is determined by the sum of potential amplitudes reaching that neuron) and *temporal summation* (rapid succession of potentials superimpose their amplitudes). It means that synaptic inputs from separate locations and repeated stimuli within a brief time produce a cumulative effect that may or may not trigger the action potential in the postsynaptic neuron [36]. Fig. 1.6 shows that different synaptic configurations between neurons can realize simple Boolean logic functions. Specifically, it is possible to model the NOR gate, a serial connection of OR gate with an inverter (NOT gate), which possesses a *functional completeness* property, i.e., every Boolean logic function is possible to be combined solely from NOR gates [37]. Consequently, neuronal systems, thanks to their architecture, can express any logical behavior needed.

Overall, it can be concluded that the behavior of each neural network depends heavily on its internal state and inter-neuronal configuration. For an artificial neural network, its function is encoded in its hyperparameters — the number of neurons, the number of layers, the architecture of connections, transfer functions, and the weights [38]. For biological neurons, the type of neurons, the configuration of their synapses (their number, type, and location) heavily affects on the overall human mood, character, mental and physical performance, and well-being in general. Coffee, cigarettes, recreational drugs, psychotropic drugs — all of them owe their impressive effects to the ability to modify synaptic transmission (e.g., by blocking the receptors of certain neurotransmitters or preventing reuptake). Does it mean that we, individuated units of consciousness, are no more than just individuated sets of synapses [39]?

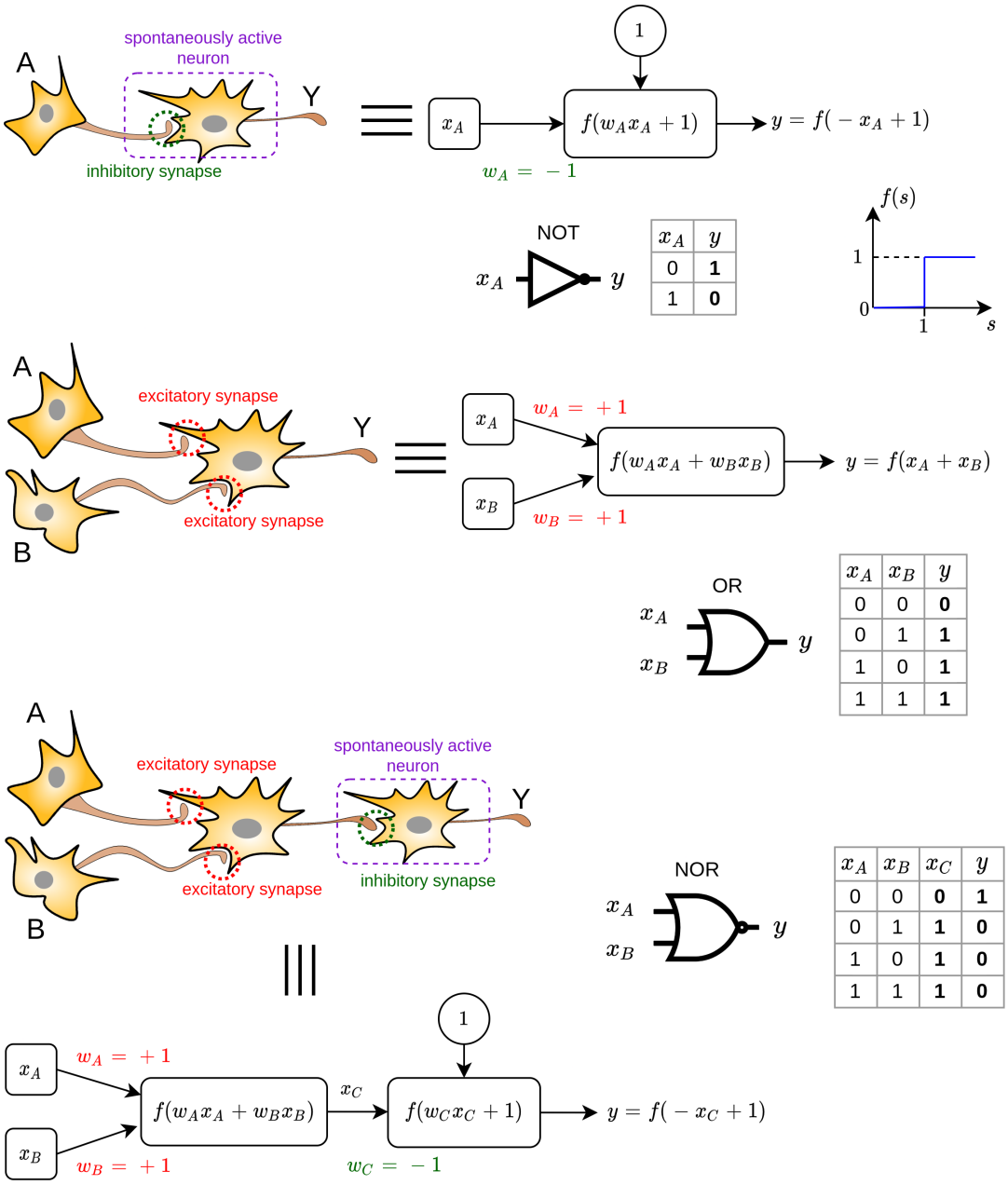


Fig. 1.6. A simplified explanation of the significance of excitatory/inhibitory types of synaptic connections. Using artificial neural networks notation, it is possible to model a system that can produce the output of logical gates NOT, OR, and NOR. Synapse types are modeled by edge weights w connecting neurons between layers. Spontaneously active neurons are represented using bias terms (circles). Temporal and spatial summation of synaptic activity is realized by the function f, which thresholds the weighted linear combination of its inputs. NOR gate can be constructed as a serial connection between OR and NOT architectures. Appropriate combinations of NOR gates can realize any desired logical function (*functional completeness*, [37]).

1.3 "The Great Neural Bell" - Cerebral Cortex

Church towers used to be one of the most important reference points in the town. The periodical ringing of the church bell organized life and guided the way. One of the biggest bells in the world is the Great Bell (Big Ben) in London, UK.

Analogously, the cerebral cortex produces the "loudest" signal in the electromagnetic measurements of the brain's activity due to the proximity of the sensors to the head surface. Of course, cortical activity is driven by the subcortical structures (e.g., via connections called *thalamocortical pathways* [40][41][42]). The magnitude of the generated field decreases with the distance of subcortical neural ensembles to the sensors [43], so results regarding the cortical sheet are more likely to be sound than those concerning beneath structures.

From the standpoint of this thesis, cerebral cortex is the most important part of the brain. It facilitates both sensory and associative functions of human cognition. The cortical sheet is only few millimeters thick, and yet, inside it happen neural computation related to primary and secondary sensory processing, understanding language, shifting attention, and more. Of course, some of these functions are mediated by the subcortical structures like the thalamus, but most of the inputs to the cortex happen to start from the cortex itself [44][45]. The output from the cortex is directed to the basal ganglia, thalamus, spinal cord, and, evidently, to the cortex itself. These mutual connections between different brain structures are the wirework (*anatomical connectivity*) for different aspects of cognition, organized into distinct functional patterns that we call *brain networks* [46]. To maximize the usage of space, the cortical sheet is heavily folded, as is the case for most large mammals like primates, dolphins, whales, cows, sheep, pigs, cats, dogs, and ferrets. Most of the folding happens between 27th and 37 week of the human prenatal development and during that time is mediated by tensions related to different growth speeds and directions of the white matter, subcortical and cortical layers [47].

Cortical neurons. Maturated cortex is built of approximately 16 billion neurons - much more than it happens to be in the spinal cord, but about five times less than in the cerebellum. About 70 % of the neurons are of pyramidal type [48]. There are six horizontal layers of neurons between the pial surface (innermost layer of the meninges) and the white matter (Fig. 1.8). Layers differ from one another by the amount and type of neurons within. These cytoarchitectural differences served as a basis for Korbinian Brodmann to differentiate approximately 52 distinct areas of the human brain [49][50]. Radially oriented columns gather neurons that serve a similar purpose (Fig. 1.7).

Cortical landmarks and divisions. Brain hemispheres are separated by the

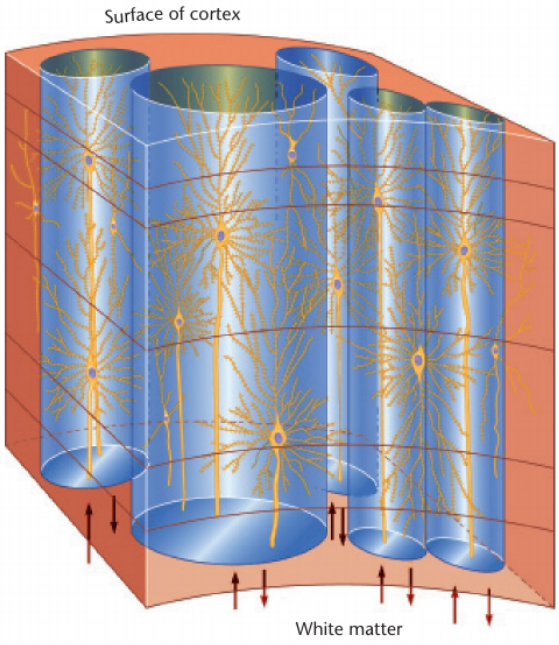


Fig. 1.7. Neurons in the cerebral cortex are organized in columns, sharing similar functions. Source: [51].

longitudinal fissure but bridged by two bundles of axons: the *corpus callosum* and the *anterior commissure* (Fig. 1.9). There are also smaller commissures that connect different subcortical structures as well. In Fig. 1.10, we can see highlighted areas called *lobes*.

The left part of the occipital lobe receives the input from both eyes. It is also true for the right part of the occipital lobe. Interestingly, even with eye or optical nerve damage, visual imagery remains possible as long as the visual cortex is healthy [53][54] but only for people who have learned to see before the damage.

Despite being the primary target for auditory information, the *temporal lobe* aids in forming advanced visual representation, including the perception of movement and recognition of faces. The temporal lobe also contains the Wernicke's area, which is essential for comprehending written and spoken language, in contrast to the Broca's area, which is a part of the *frontal lobe* involved in language production. Damage to the temporal lobe and medial temporal lobe structures (including the amygdala, hippocampus and its surroundings) produces a set of behaviors known as Klüver-Bucy syndrome, which distorts the processing of emotions, fear and impairs motivation [55][56].

The *parietal lobe* lies between the occipital lobe and the *central sulcus*. Receptors from muscles, skin, and joints transmit sensations to the *primary somatosensory cortex*, which occupies the area of the *postcentral gyrus*. The parietal lobe is also responsible

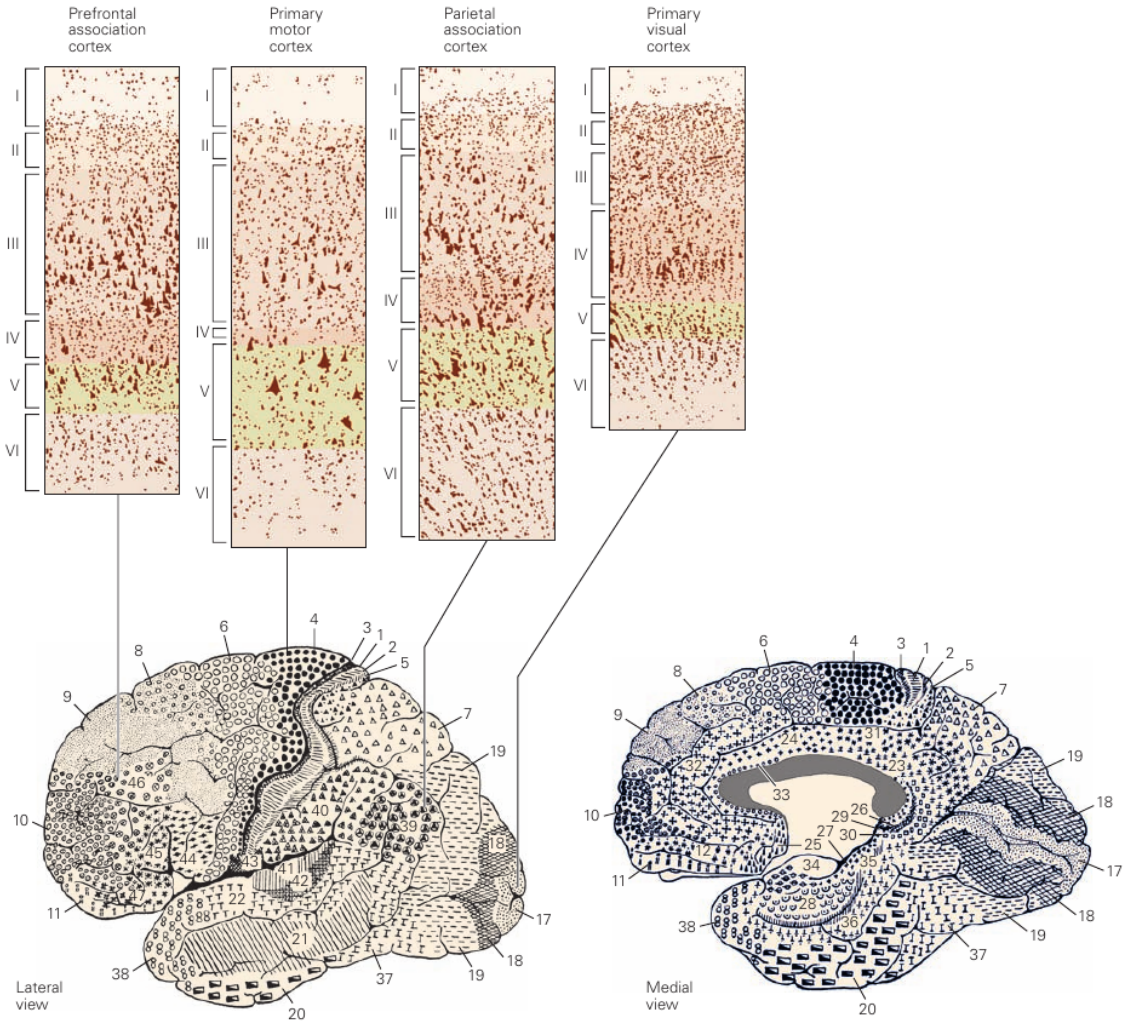


Fig. 1.8. Variability of different cortical layer compositions throughout the cortex. Roman numerals refer to layers. Numbers differentiate between various cytoarchitectonic regions according to Brodmann's 1909 subdivision. Source: [51].

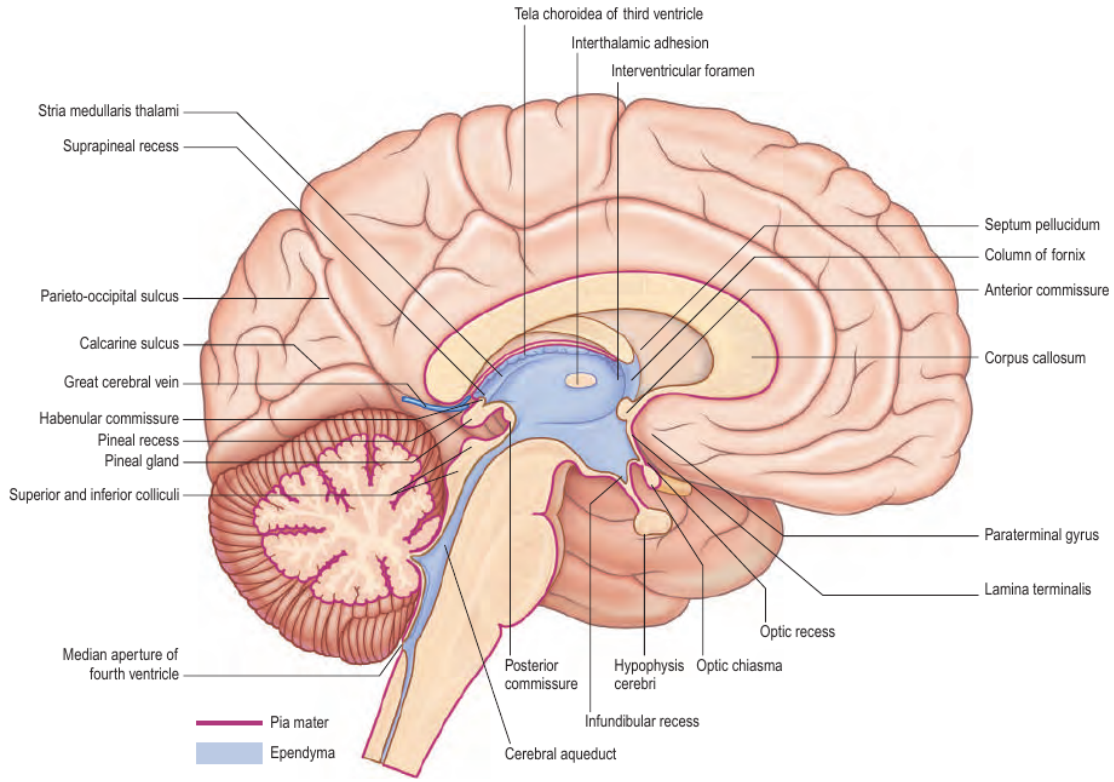


Fig. 1.9. Sagittal hemisection of the brain. Source: [52].

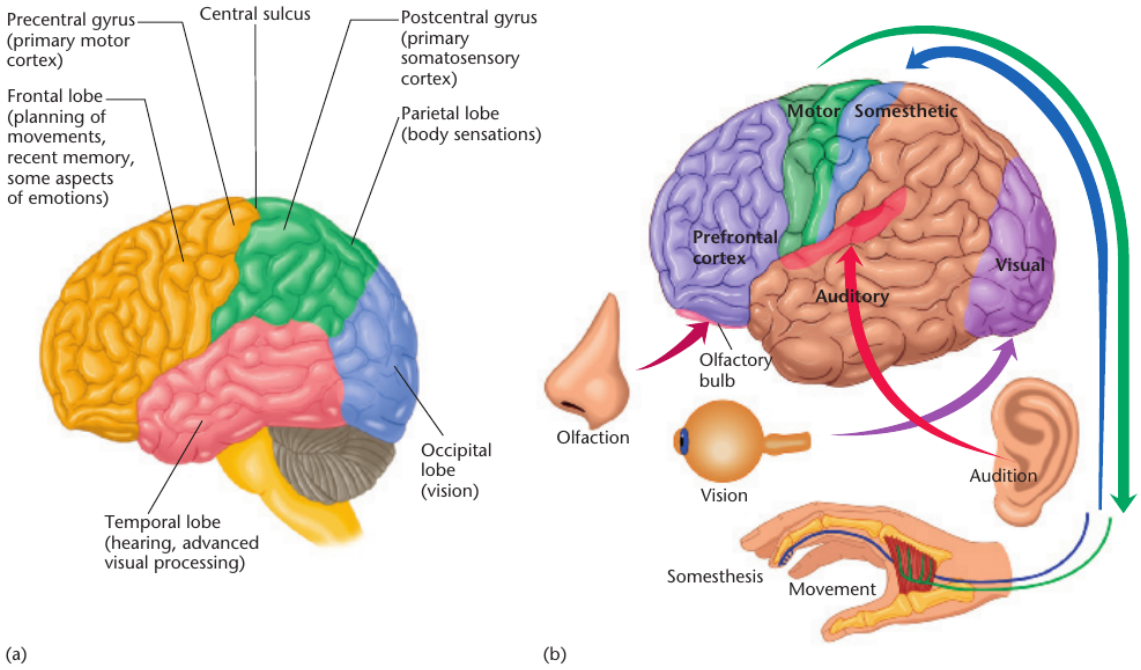


Fig. 1.10. **(a)** Areas of the human cortex and **(b)** their general functional roles. Source: [29].

for monitoring the position of eyes, hands, and body in general, and passes information to the movement planning and control regions of the brain. Interestingly, there exist some parietal neural circuits that are responsible for the mental link between numbers and space [57].

Next to the postcentral gyrus, there is the *precentral gyrus*, which is the part of the *frontal lobe*. It functions as the *primary motor cortex*, which controls mostly contralateral parts of the body (with some minor exceptions).

Lastly, the *prefrontal cortex*, which is a part of the frontal lobe, can be divided into three zones [58]:

- posterior part - mainly responsible for movement as it coincides with the primary motor cortex,
- middle part - in charge of working memory, emotional reactions, and cognitive control,
- anterior part - calculating difficulty, outcome, and value of actions to facilitate making informed decisions.

What is also a remarkable function of the brain is its *neural plasticity*. Particular functions of brain lobes are not written in stone, i.e., seemingly lost functions of the damaged brain areas can be taken over by another site and restored to some extent [59][60]. This desired ability of the brain to change itself according to environmental and internal pressures is the substrate of learning - a crucial capacity for each of us [61].

1.4 From synapses to measurable electromagnetic fields

Measurable neural events originate in synaptic activity (see Section 1.2). Information between neurons is transmitted along the axon in the form of electrical pulses (action potentials), reaching a synapse and provoking a release of neurotransmitter molecules that, after interaction with postsynaptic receptors, finally triggers the movement of ionic currents in the postsynaptic neuron (see Fig. 1.11) — so-called *primary currents* inside a neural cell, and *volume currents* (sometimes called *return currents*), representing the flow of charges outside the membrane, as a "reaction" to the primary current.

Let us think about primary currents and volume currents in terms of the current density $\vec{J}$ defined as:

$$\vec{J} = nq\vec{v}_d \qquad \Big[\frac{C}{m^2 s} = \frac{A}{m^2}\Big], \tag{1.2}$$

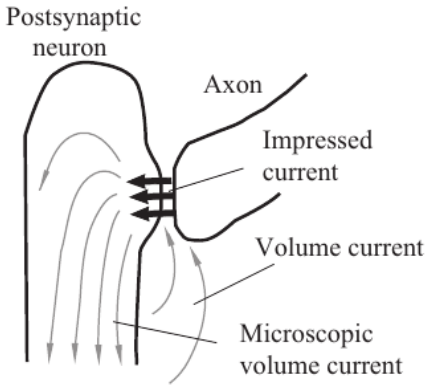


Fig. 1.11. The movement of neurotransmitter molecules from the presynaptic neuron to the postsynaptic neuron (captured by receptors) triggers the flow of the ionic currents. Source: [62].

where n is the number of charge carriers, each having a charge q, passing unit area perpendicular to the flow of the charge; $\vec{v}_d$ is the average drift velocity of charge carriers. Primary current density will be denoted as $\vec{J}^p$, and the volume current as $\vec{J}^v$. Current I is defined as the total amount of charge, Q, moving in the unit of time:

$$I = \frac{dQ}{dt} \qquad \left[\frac{C}{s} = A\right], \tag{1.3}$$

which we can think of as a total flow of charge flowing through some surface S (see Fig. 1.12):

$$I = \int_S \vec{J} \circ \hat{n}\, dS, \tag{1.4}$$

where $\circ$ denotes the dot product, $\hat{n}$ is a unit vector normal to the infinitesimal surface segment dS.

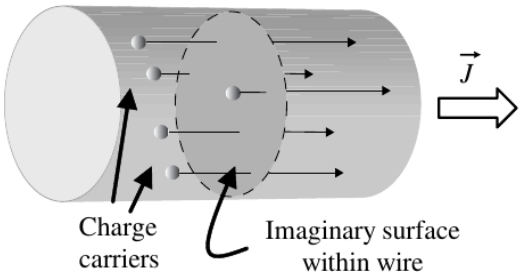


Fig. 1.12. Current density as the amount of charge flowing through an imaginary surface in a conductor. Source: [63].

These currents change the spatial distribution of the charges, denoted here as ρ, and thus electric and magnetic fields surrounding neuron change, according to the set of four famous *Maxwell's Equations* - the cornerstone of modern electronic technology,

pioneered by Maxwell in 1850s and developed two decades later by Olivier Heaviside and Heinrich Hertz [63]. To see how neural currents contribute to the electromagnetic field, let us have a look at Fig. 1.13. Postsynaptic primary currents, $\vec{J^p}$, cause the change of the charges density ρ inside and outside the membrane according to the continuity equation for electromagnetism (no charges are being created or annihilated):

$$\frac{d\rho}{dt} = -\vec{\nabla} \circ \vec{J},$$

where $\vec{\nabla} = \hat{i}\frac{d}{dx} + \hat{j}\frac{d}{dy} + \hat{k}\frac{d}{dz}$ denotes gradient operator, $\hat{i}, \hat{j}, \hat{k}$ are the unit vectors of the axes of the Cartesian coordinate system, and $\vec{\nabla} \circ \vec{J}$ is the divergence of the current density vector field $\vec{J}$. From Maxwell's first equation (*Gauss's Law for Electric Fields*), electric field $\vec{E}$ is produced by charges that act like the field's sources and sinks. Electric field drives the volume currents $\vec{J^v} = \sigma\vec{E}$ proportionally to the medium conductivity σ. Volume currents affect the charge density ρ, which constitutes a feedback loop (this change of charge density will affect the electric field, which will affect volume currents, which will affect charges density, which will affect the electric field, which ...). At the same time, both primary and volume currents, by Maxwell's fourth equation (*Ampère-Maxwell's Law*), will produce a circulating magnetic field. In neuroscience, the Ampere-Maxwell's Law is simplified by dropping the second term related to time-varying changes of the electric fields, which is believed to be an excellent approximation [62][64]:

$$\vec{\nabla} \times \vec{B} = \mu_0 \left(\vec{J} + \varepsilon_0 \frac{\partial \vec{E}}{\partial t} \right) \approx \mu_0 \vec{J}, \tag{1.5}$$

where $\vec{\nabla} \times \vec{B} = \hat{i}\left(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z}\right) + \hat{j}\left(\frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x}\right) + \hat{k}\left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}\right)$ is the curl of the magnetic field $\vec{B} = [B_x, B_y, B_z]$, μ_0 is the magnetic permeability of free space ($\mu_0 \approx 4\ \pi\ \times\ 10^{-7}$ Vs / Am), ε_0 is the electric permittivity of free space ($\varepsilon_0 \approx 8.85\ \times\ 10^{-12}$ C / Vm), and $\vec{E} = [E_x, E_y, E_z]$ is the electric field. This simplification is called the *quasi-static approximation* for time-dependent electromagnetic phenomena.

How are these processes being expressed at the whole-brain scale? Imagine a brain divided into small (e.g., 1 mm^3) non-overlapping volumetric cubes called *voxels*. In such a macroscopic scale, the net contribution of the primary and volume currents to the electromagnetic field, measured by the array of sensors at the surface of the scalp, can be modeled as a set of *current dipoles*, which is used to approximate a concentration of the current in a small volume. Each dipole, located inside each cell of the 3-D brain grid, has its *dipole moment*, i.e., orientation in space and magnitude. Each grid cell with the dipole inside is called a *source*. Relation between sources of brain activity and

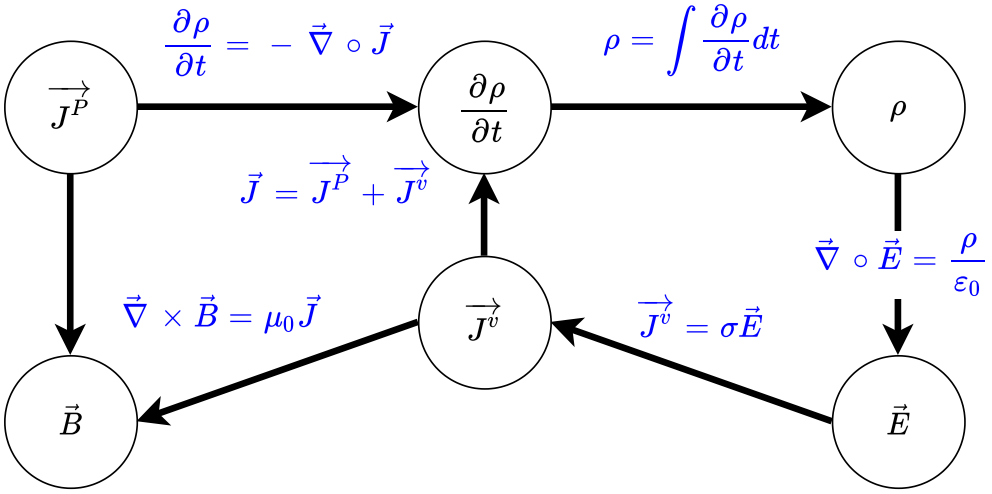


Fig. 1.13. Fields generated by the primary current (see text for the description). Adapted from: [62].

readings from the sensor array is modeled by the system of linear equations:

$$\mathbf{Y} = \mathbf{AX} + \eta. \tag{1.6}$$

Here, the data matrix $\mathbf{Y}$ denotes the set of signal values obtained from MEG or EEG measurements; η is the noise matrix. $\mathbf{A}$ is called *transfer*, *mixing*, or *lead field* matrix that describes the linear relationship between the source currents and the measured electromagnetic fields; $\mathbf{X}$ denotes the amplitudes or time-courses matrix of the unknown neuronal source currents. The element $y_{it}, i = 1, \ldots, N_{\text{ch}}, t = 1, \ldots, N_{\text{t}}$ of the $N_{\text{ch}} \times N_{\text{t}}$ matrix $\mathbf{Y}$ is the sensor signal recorded at time t by i th sensor (magnetic or voltage sensor). A typical magnetoencephalographic magnetometer measures the flux of a magnetic field through a loop of wire while an EEG voltage sensor is picking up the potential difference between two electrodes (Fig. 1.14):

$$\text{MEG:} \qquad \Phi_i = \int_{S_i} \vec{B}(\mathbf{r}) \circ d\vec{S}, \tag{1.7}$$

$$\text{EEG:} \qquad V_j = \int_{\mathbf{r}_{\text{ref}}}^{\mathbf{r}_j} \vec{E}(\mathbf{r}) \circ d\vec{l}, \tag{1.8}$$

where Φ_i is the magnetic flux through the wire loop S_i of magnetic sensor i and $V_j = V(\mathbf{r}_j) - V(\mathbf{r}_{\text{ref}})$ is the potential difference between electrode j located at $\mathbf{r}_j$ and reference electrode located at $\mathbf{r}_{\text{ref}}$.

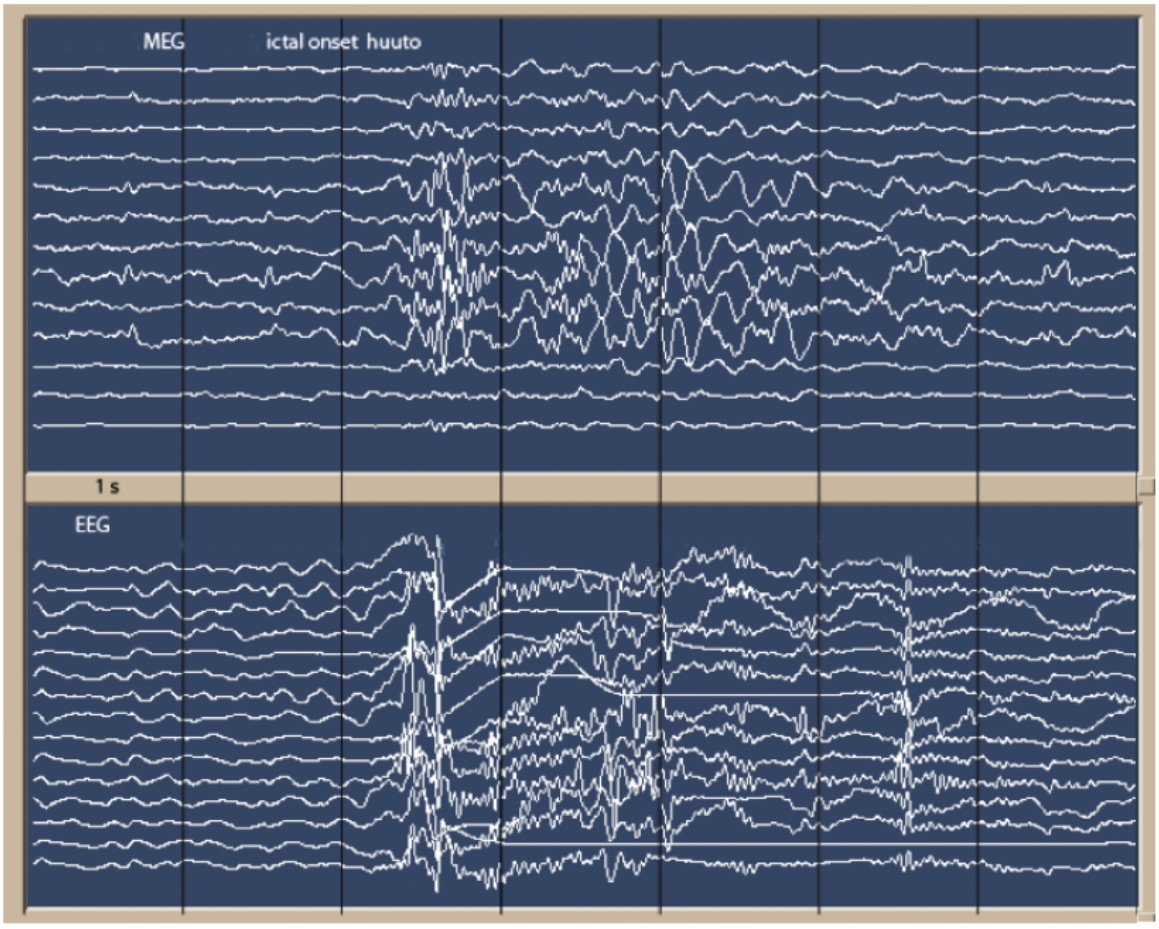


Fig. 1.14. MEG and EEG signals the onset of an epileptic seizure. Source: [62].

It can be assumed that the main generator of the MEG and EEG sources (Fig. 1.15) are solely the primary currents arising from the postsynaptic potentials (action potentials do not contribute significantly). Although the volume currents contribute to the tangential components, they are not explicitly needed for the field estimation. The contribution of the tangential components can be derived from the radial components, which can be computed purely from the primary current [62].

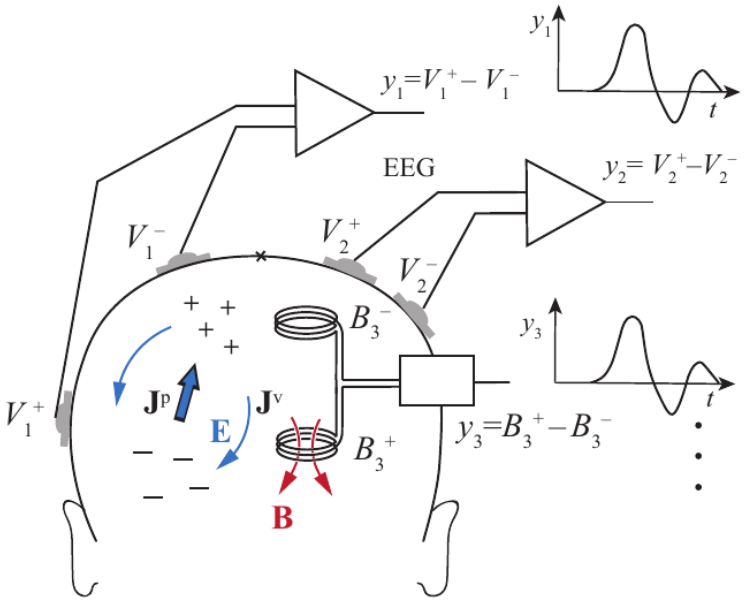


Fig. 1.15. One spatially fixed current $\mathbf{J}^{\mathrm{P}}$ causes the flow of the volume current $\mathbf{J}^{\mathrm{v}}$. EEG sensors pick up electric field changes as a difference between two potentials. Same currents produce additional magnetic flux sensed by flux transformers, here configured to measure the difference between two magnetic fluxes. Source: [62].

At location $\mathbf{r}$, joint primary source-current density $\mathrm{J}^{\mathrm{P}}(\mathbf{r}, t)$, which is a weighted sum of several component source-current density patterns $\mathrm{J}_s^{\mathrm{p}}, s = 1, \ldots, N_{\mathrm{s}}$, being spatially fixed and time-independent. Entries of the $\mathbf{X}$ matrix are the time-dependent amplitudes x_{st}. It follows that the joint primary current is expressed as:

$$\mathbf{J}^{\mathrm{P}}(\mathbf{r}, t) = \sum_{s=1}^{N_{\mathrm{s}}} x_{st} \mathbf{J}_s^{\mathrm{P}}(\mathbf{r}). \tag{1.9}$$

Element $A_{is}, i = 1, \ldots, N_{\mathrm{ch}}, s = 1, \ldots, N_{\mathrm{s}}$ of the mixing matrix $\mathbf{A}$ in Eq. 1.6 is the signal that would be recorded by sensor i if the source-current pattern $\mathbf{J}_S^{\mathrm{p}}(\mathbf{r})$ were acting alone with unit source amplitude $x_{st} = 1$. Elements of the matrix $\mathbf{A}$ are dependent not only on the spatial distribution of the source currents but also on the conductivity structure of the head (called the *head model*). There are many head models to be chosen from, with varying degrees of detail and the number of tissue classes considered

[64][65]. It is an essential choice for EEG studies as electric fields are more distorted by tissue compartments than magnetic flux. In case of MEG, which is the main method used in this work, it is recommended to use shell models rather than sphere models, preferably three-shell model [66].

1.5 Different perspectives on oscillatory events

Brain activity represented by recorded sensor signals consists of periodic patterns — a straightforward observation that unaided eye can make (Fig. 1.14). Consequently, in analog EEG, one of the first ideas to classify neural events was to visually examine the paper readout and detect the presence of specific classes of oscillations in the signal (called *brain rhythms*) by counting the number of cycles in fixed periods of time. However, one can also take a different perspective. Living organisms are constantly exposed to various stimuli that trigger different responses. Processes of conscious and subconscious reactions, decision-making, and executing chosen responses are accompanied by spiking changes of whole neural populations, leading to detectable fluctuations in generated electromagnetic fields. Changes that have more or less fixed time-delay to the stimulus are called *event-related potentials* (ERPs), and they can be discovered by simple linear techniques like averaging. Sadly, some neural populations' synchronicity is not phase-locked, thus cannot be detected by averaging but rather using time-frequency analysis methods. Such changes are called *event-related synchronization* and *desynchronization* (ERS/ERS). Let us have a closer look on all these types of oscillatory events.

Brain rhythms. In 1929-1933, German neurosurgeon Hans Berger published a series of papers documenting for the first time the joint electrical field produced by synchronous spiking of neural assemblies in the human cortex and led to the discovery of so-called *Berger's rhythm*. Berger's rhythm is the prominent presence of about 10 Hz spectral power peak in the brain signal when a subject closes his eyes. Adrian and Matthews further studied the effect in 1934 [67]. Soon after, several other brain "rhythms" emerged, each having different links to known cognitive processes. The first classification, introduced by the experts from the International Federation of Societies of Electroencephalography and Clinical Neurophysiology, was proposed in 1974 [1]. Much of the knowledge about the neural mechanisms of rhythmical brain activity was contributed by Romanian researcher Mircea Steriade [68][40][69]. Studying these oscillatory events is helpful for understanding not only humans as rhythmic activity is evolutionarily preserved across species [70].

Brain rhythms are traditionally grouped into coarsely specified bands. These bands include:

- delta δ (2–4 Hz),
- theta θ (4–8 Hz),
- alpha α (8–14 Hz),
- beta β (15–30 Hz),
- lower gamma γ_1 (30–80 Hz),
- upper gamma γ_2 (80–150 Hz).

Boundary frequency values may differ slightly depending on the literature. There are also oscillations in the subdelta and omega (up to 600 Hz) ranges, but they tend to be of weak power; hence the frequency bands most typically associated with cognitive processes in the literature are between 0.1 Hz and 150 Hz [1][71].

Frequency peaks inside presented ranges, despite being usually contained within bands described above, may differ from subject to subject slightly and are also not stationary [72][73]. During development, brain rhythms also undergo significant changes [74]. It is also known that every brain region has its preferable frequency of operation [73].

Alpha rhythm. This is the most widely known oscillation type, more or less observed across the whole cortex but most pronounced in the occipital lobe. Usually, the alpha rhythm peaks somewhere between 8-13 Hz. It is connected to vision - it is suppressed when eyes are open and stronger when eyes are closed. It is also briefly suppressed at the onset of visual stimuli [33]. More generally, alpha-band suppression is present in task-related active brain regions together with the alpha-band enhancements in non-task-related brain regions [75][76]. Engagement in internal tasks such as mental calculation or accessing working memory causes alpha-rhythm to enhance [77] and is closely linked to attentional processes [78]. Sometimes mu and tau rhythms are considered as different types of alpha oscillations, however they are associated with different processes and are present in different regions of the cortex [79].

Mu rhythm. The mu rhythm was described by Henri Gastaut in 1952 [80]. It is present near the central sulcus in the primary somatosensory cortex [81]. It is a mixture of 10 Hz and 20 Hz oscillations connected with the execution of the motor actions [82]. These two components are not exact harmonics [83]. Mu rhythm is first suppressed during voluntary movements, and then it phasically increases following the movement completion [84]. Alpha and mu rhythms share common frequencies however they differ in sites they are usually present (Fig. 1.16). To avoid confusion, it is advisable to indicate both frequency and location of the rhythm.

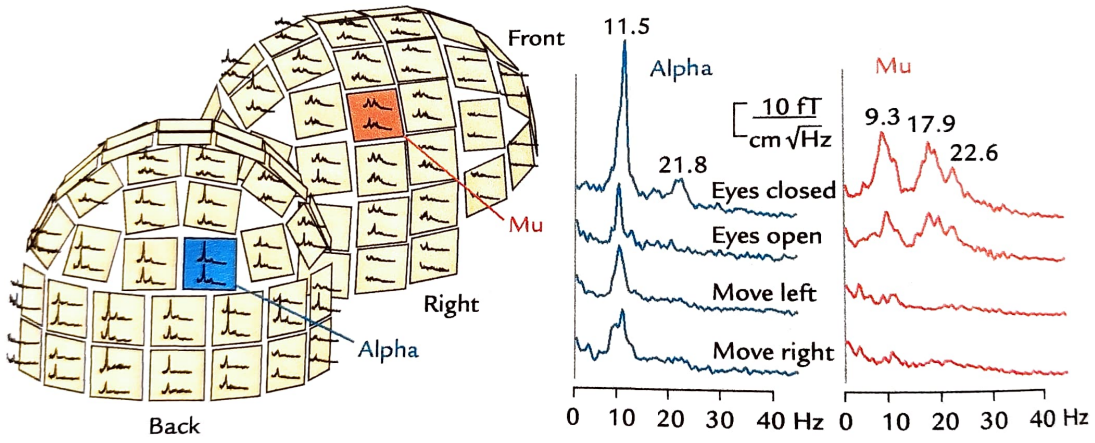


Fig. 1.16. Amplitude spectra of MEG alpha and mu rhythms. **Left**: A sensor array shown from two viewing angles (back where alpha is the strongest and right where mu rhythms dominate). On each sensor unit, spectra are displayed from two orthogonal planar gradiometers. **Right**: Magnified amplitude spectra from the left panel. Eye opening weakens alpha oscillations, and mu rhythm is abolished during hand movement. Source: [33].

Tau rhythm. The frequency band of the tau rhythm is 8-10 Hz, similar to alpha, but its neural sources are localized in the supratemporal auditory cortex [85][86][79]. Tau is suppressed when auditory stimulus arrives. Tau can be easily differentiated from alpha because eye opening/closing does not affect tau.

Beta rhythm. Beta rhythm has been ascribed to multiple functions [87] with varying frequencies and generation sites. It is not uncommon to divide the beta band into two separate classes like β_1 for 12-20 Hz and β_2 for 20-30 Hz subbands. The most common generators of beta rhythms are the primary motor cortex, near the the central sulcus (as a component of the mu rhythm), supplementary motor cortex, cerebellum, and also basal ganglia, and thalamus. Beta rhythms are involved in coordination among many parts of the neocortex, inhibition of movement, motor planning, language processing, making decisions provided enough evidence, focus, and learning [87].

Theta rhythm. Theta rhythms are slow 4-8 Hz oscillations often related to drowsiness, sleep, encoding and retrieval of spatial information from episodic memory [88], maintenance of working memory [89][90]. Studies localizing different frequencies across the human brain suggest that theta rhythm is present primarily in the frontal and frontotemporal areas of the cortex [91][9][92][79]. There is also evidence on cortico-hippocampal interplay engaging theta rhythms [93]. Additionally, studies found an interesting interplay between theta and gamma rhythms, where theta is proposed to enable effective quantization of information in the form of specific synchronicity between those two bands, called the *theta-gamma coupling* [89][94][95].

Gamma rhythm. There is no consensus in the literature on the span of the gamma

rhythms. The typical class considered is the 30-60 Hz frequency band, however there are reports considering other five subbands: Omega (60-120 Hz), high gamma (60-200 Hz), ripples (100-200 Hz), ultra-fast gamma (200-600 Hz), and fast ripples (250-600 Hz) [96], however in the scalp EEG studies gamma is harder to capture due to the distortions of the electric fields by the skull and scalp [97]. For such problems, *electrocorticography* (ECoG) comes in handy [98]. It is a type of electrophysiological monitoring that uses electrodes placed directly on the exposed surface of the brain to record electrical activity from the cerebral cortex. This way, distortions are greatly reduced.

Modulation of the gamma rhythm is linked with feature binding, multisensory integration, sensorimotor coordination, operation of working, and long-term memory [99][100][101][102]. Gamma oscillations can be found locally in different parts of the brain. Available reports point to the entorhinal cortex, amygdala, hippocampus, striatum, olfactory bulb, thalamus [103].

Delta rhythm and ultra-slow oscillations. In healthy humans, theta rhythm (frequencies below 3.5 Hz) occurs only during deep sleep. Its presence in an awake state is a sign of brain pathology. Slower rhythms may be visible in the human encephalograph, however they are prone to artifacts in the form of slow drifts. The study from Hughes in 2011 has shown the presence of ultra-slow awake activity in the thalamus [104], however, they are more likely to occur during sleep [105]. Some studies attribute delta rhythms present in the motor cortex to the processing of sensory temporal contextual information and linguistic information [106].

Rhythms coupling. During tasks and rest, different brain rhythms can interact. *Cross-frequency coupling* (CFC) is the phenomenon where the faster oscillations are modulated by slower rhythm (phase-phase CFC, phase-amplitude CFC, amplitude-amplitude CFC) [87]. Findings on CFC suggest that it mediates parsing neuronal computations into discrete chunks of activity to optimize attention, learning, memory and help process rhythmic external sensory and motor activity. As mentioned earlier, one prominent example of CFC is the *theta-gamma coupling* [94][95], which is used to organize information in the hippocampus and transmit it to targets in the prefrontal cortex and striatum during successful recall of the working memory. Various types of interactions are described in several reviews on the topic [107][108][109].

Evoked potentials. Other important classes of brain activity are the *evoked responses* and *event-related responses*. They are replicable stable signal patterns related to specific stimuli types, and can be estimated by averaging hundreds of repeats of recorded stimulus-locked responses (see Fig. 1.17b). These averaged time courses are called ERPs — *event related potentials*. EEG, MEG, and other neuroimaging techniques with fine time resolution are suitable for this purpose. A curious reader is directed

towards an excellent book on the topic [110].

Event-related synchronization/desynchronization (ERS/ERD). There are two important classes of neural responses: *phase-locked* and *non phase-locked*. The basic assumption behind phase-locked response is that it includes evoked activity with a more or less fixed time delay to the stimulus. Then other unrelated activity can be treated as additive noise. The power of the signal of interest compared to the power of the noise, known as the signal-to-noise ratio (SNR), can be enhanced by averaging signals across repetitions. This method, however, will not work with the non phase-locked responses, in which cycles are not aligned across repetitions to produce a meaningful average. It is similar to the destructive interference phenomenon. However, time-frequency analysis can reveal non-phase locked components (Fig. 1.17c).

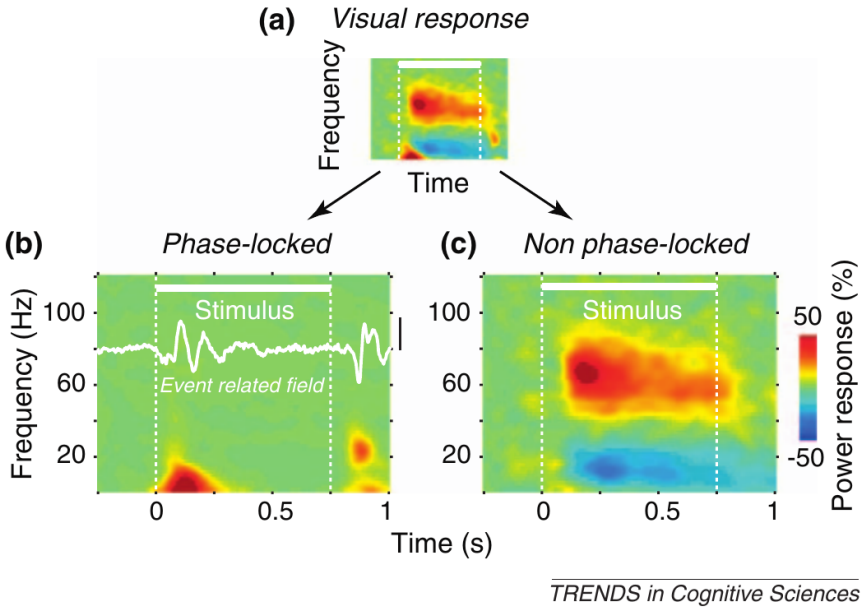


Fig. 1.17. **(a)** Time–frequency representation of a human MEG response to a visual stimulus. **(b)** The phase-locked signal components corresponding to the event-related field calculated by averaging the MEG signal across trials in the time domain. **(c)** The non-phase-locked components isolated by subtracting the time-domain average from each trial before transforming the data into the frequency domain. Source: [8].

They are related to the change of synchrony of the underlying neural populations. In this vein, increases in the spectral power in the time-frequency plane of the response are called *event-related synchronizations* (ERS), and decreases are referred to as *event-related desychronizations* (ERD)[75][111]. Integrative functions such as perceptual inference, top-down attention, and decision-making can be linked with non phase-locked activity components [8].

1.6 1/f power scaling

Frequency spectrum of many signals, including MEG and EEG, typically consists of two components. The first component is a periodic part that appears as spectral peaks [112], and the second component is an aperiodic part usually following *1/f power law*:

$$P \propto 1/f^{\beta} \tag{1.10}$$

where $\beta > 0$ is scaling exponent shaping slope of spectrum.

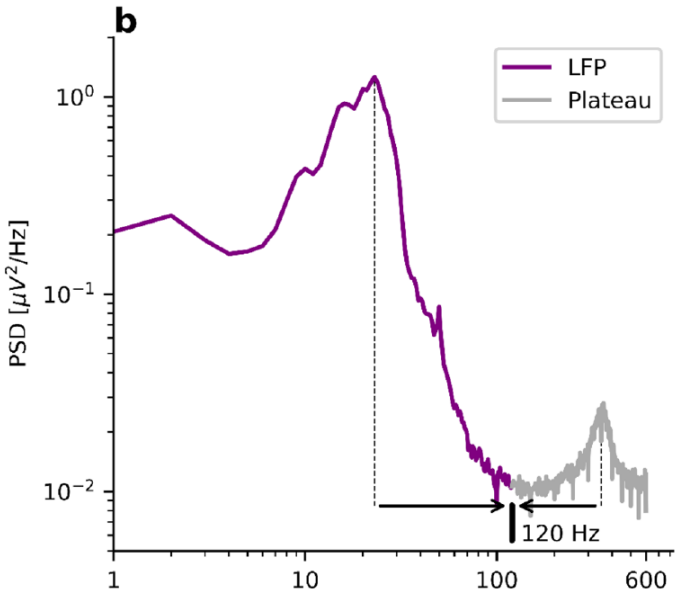


Fig. 1.18. Electrophysiological power spectral density plot (PSD) with visible 1/f power scaling. Local field potential (LFP) recorded from a Parkinsonian patient with an electrode located in the subthalamic nucleus. The spectral plateau disrupts the 1/f power law. Source: [113].

Power scaling is attributed to the integration of synaptic currents underlying obtained spectra [114]. This phenomenon needs to be accounted for during signal analysis to interpret underlying neural processes correctly. Various methods for reducing the impact of 1/f scaling are presented in [113].

1.7 Observing oscillations with magnetoencephalography

Since all electric currents produce magnetic fields, it was reasonable to assume that the human brain also produces the magnetic field correlated to its activity. It does, but the problem is the order of magnitude of these fields. The human brain produces the magnetic field of an average strength of about 10 to 100 femto(!)teslas (100 fT = $1.0 \cdot 10^{-12}$ T = $0.000000000001 \frac{kg}{s^2 \cdot A}$). To put things into perspective, the Earth produces

a magnetic field of magnitude approximately[3] 50 μT $= 5.0 \cdot 10^{-5}$ T $= 0.00005$ T [115], and the strength of refrigerator magnets[4] is about 10 mT $= 1.0 \cdot 10^{-2}$ T $= 0.01$ T. So, to measure the magnetic field that the human brain produces, we need extremely susceptible instruments.

Measuring magnetic fields is simple, in principle. If one place a wire loop into a changing magnetic field, a current will be induced into the loop. Consequently, one can measure a voltage difference (voltage induced by the magnetic flux Φ_B is called the *electromotive force* or *emf*) between the end of the wire forming the loop surface S according to the equation (the *Faraday's Flux Rule*):

$$\text{emf} = -\frac{d\Phi_B}{dt} = -\frac{d}{dt}\int_S \vec{B} \circ d\vec{S}. \tag{1.11}$$

The sensitivity of such device can be increased N-times by replacing the single loop with a coil with N-turns of wire, as well as increasing the area of the coil S. Putting a high-permeability core within the coil will further increase the magnetic flux and thus the induced voltage.

Superconductance. In 1960s, the first measurements in the history were made using such setup, however the sensitivity of MEG recordings improved significantly with the invention of superconduction tunneling by Brian Josephson [116], which awarded him the physics Nobel Prize in 1973. Certain metals like niobium, lead, or mercury switch into a *superconductive* state when being cooled down to extremely low temperatures like 10 K (-263.15 °C). Superconductivity is an example of macroscopic quantum phenomena. However, this state is relatively fragile when superconducting material is exposed to stronger magnetic flux (critical field B_c), current (critical current I_c), or higher temperatures (critical temperature T_c) due to their non-linear interdependence [117][118]. It is worth stabilizing it — induced current inside superconductors can loop perpetually, without any "friction" (it behaves like the resistance is zero in such a loop). One fascinating demonstration is the superconducting gravimeter in Membach, Belgium. For more than two decades, it still measures mass shifts caused by groundwater movements, the weight of storm surge water on the surface, and other phenomena that affect the nearby gravitational field [119]. In 1995, currents were injected into the gravimeter's superconducting coils, causing the measuring device — a niobium sphere, to levitate. More than 25 years later, the currents persist [120].

[3]Geomagnetic elements estimation for Poland: Lat.: 50.380N Long.: 18.335E Altitude: 0.0m, Year:2020 (IGRF) `http://wdc.kugi.kyoto-u.ac.jp/cgi-bin/point-cgi?80,85` (accessed: June 4, 2022); see "Online computation of field components from the IGRF-11" in Finlay et al., 2010

[4]National High Magnetic Field Laboratory, University of Florida, `https://web.archive.org/web/20130321234403/http://www.magnet.fsu.edu/education/tutorials/magnetminute/tesla-transcript.html` (accessed: June 4, 2022)

SQUIDs can produce an output voltage proportional to the measured magnetic flux, providing they are kept near specific operating point by feedback electronic circuits. A SQUID itself has too small area to be useful directly in magnetoencephalography, so it is coupled with a flux transformer (Fig. 1.20; [121]).

SQUIDs and flux transformers. Each magnetoencephalographic sensor takes advantage of superconductance. The key element in the sensor is the *dcSQUID* (direct-current superconducting quantum interference device) — a small 1 mm^2 loop measuring magnetic flux, equipped with two Josephson junctions, which enable superconductance to take place, drastically increase the sensitivity of the loop, thus allowing for precise measurements of the human brain weak magnetic fields (Fig. 1.19).

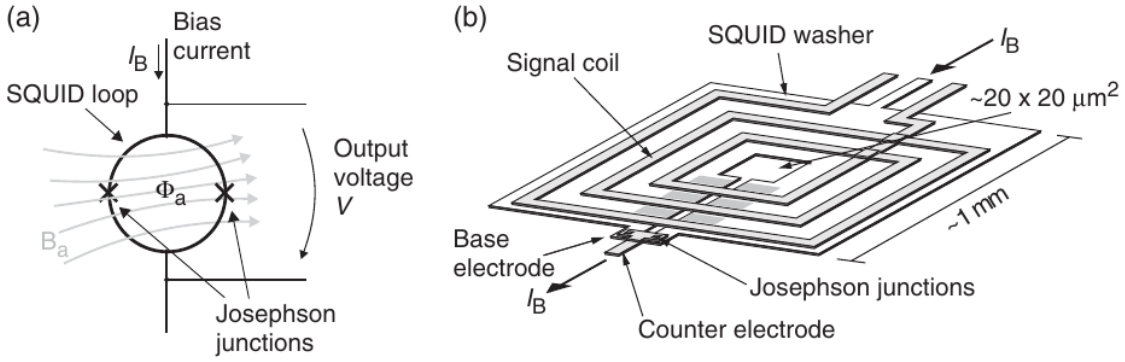


Fig. 1.19. (a) A schematic illustration of the SQUID loop sensing magnetic field B_a as magnetic flux Φ_a, producing output voltage V. **(b)** Thin-film dc-SQUID circuit. Source: [121].

A flux transformer, which has bigger area than a SQUID, "collects" flux from the location of interest, passes it to the SQUID loop, which is connected to the feedback loop circuit. Flux transformers come in many shapes, which determine the spatial sensitivity pattern of the sensor (sensor's *lead field*; Fig. 1.22) and also determine its sensitivity to external magnetic field artifacts. The most common types of flux transformers are (Fig. 1.21):

- *magnetometers* - a simple loop that is most sensitive to the brain magnetic flux but also least robust to artifacts,
- *gradiometers* - a twisted loop that measures the difference between fields entering its two loops. This cancels out external noise fields that become more or less uniform with distance. There are two main types of gradiometers:
  - *planar gradiometer* - the maximum signals are obtained just over the source where the field pattern gradient is the steepest,

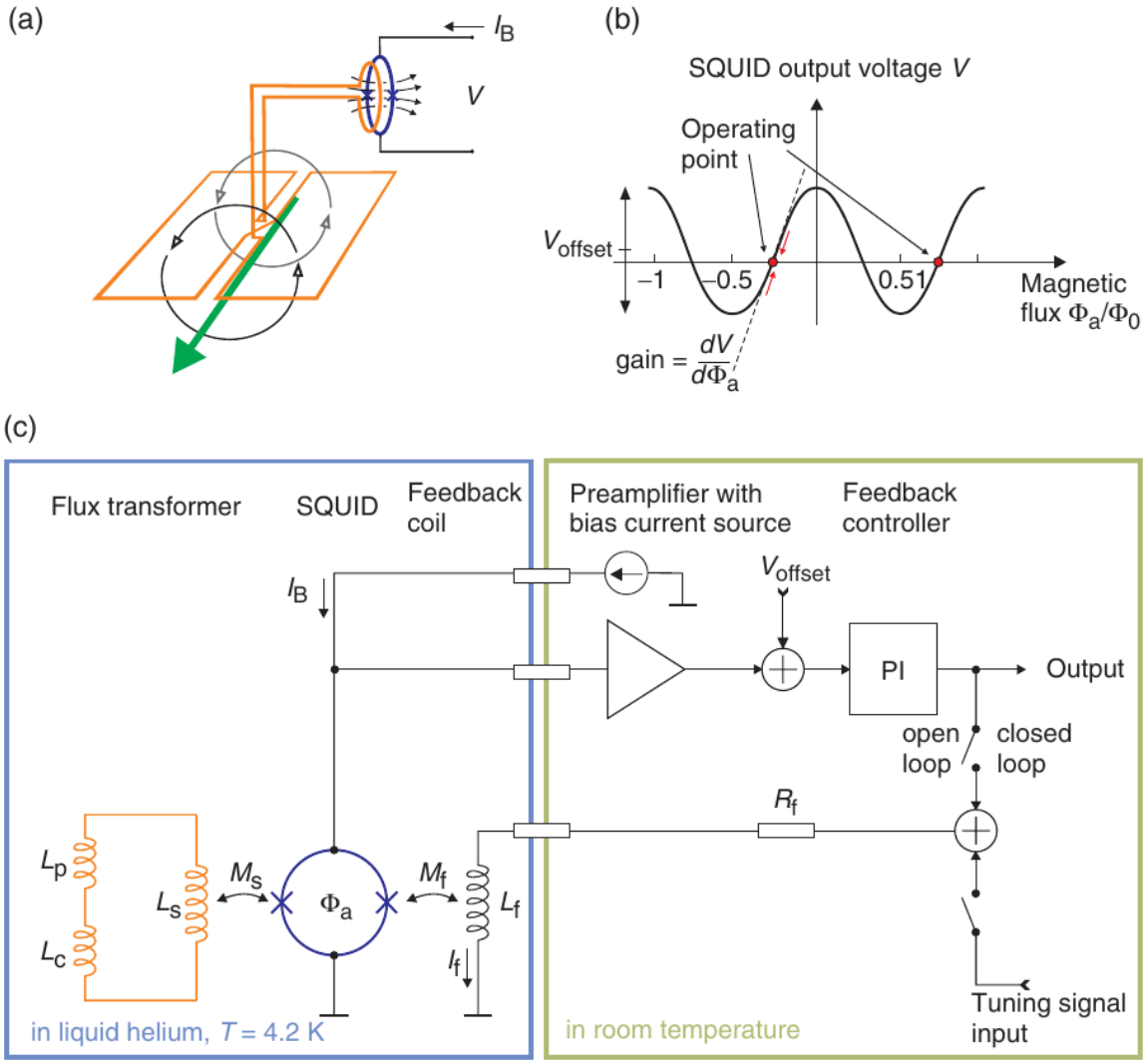


Fig. 1.20. **(a)** Neural current (green arrow) produces a magnetic field, picked up by the flux transformer loop (orange), which is coupled to a SQUID (blue ring). **(b)** SQUID output voltage is periodically dependent on the ratio between SQUID's magnetic flux Φ_a and *flux quantum* $\Phi_0 = 2.07$ Weber. **(c)** A simplified block diagram of the sensor electronics (utilizing direct readout) of one MEG channel. Source: [121].

– *axial gradiometer* - the maximum signals are obtained at the two extrema of the dipolar field pattern.

The closer a flux transformer is to the source of interest, the stronger signal it gets. Proper mapping of brain activity requires multiple measurements in distinct spatial locations.

MEG system components. Recent magnetoencephalographs contain two or three hundred "compound" sensors called *channels*. Each channel is a composition of different flux transformers to optimize its signal-to-noise ratio and spatial sensitivity pattern. For example, The Elekta Neuromag® TRIUX system features 306 channels on 102 planar sensor elements, each comprising a magnetometer measuring B_z (normal field component) and two orthogonal planar gradiometers measuring $\frac{dB_z}{dx}$ and $\frac{dB_z}{dx}$. There is a special nonmagnetic container called *Dewar* (after the inventor James Dewar; Fig. 1.23) wherein sensors reside immersed in the liquid helium ($T = 4.2$ K). Despite good thermal isolation, the helium slowly evaporates. The gaseous helium is collected back to the reliquification system allowing for reuse of the coolant. A typical whole-head MEG system boils 10–20 liters of liquid helium per day. "The helium reservoir of the Dewar is usually 70–90 liters, thus a refill is required 1–3 times a week to keep the system operational"[121]. Adjustment of the angle and position of the Dewar to the subject is possible with the mechanical system called the *gantry*.

Measurement conditions. MEG measurements need to be proofed against external electromagnetic noise. The main external magnetic artifacts arise from moving vehicles, power lines, and radio transmission. Shielded rooms use two metal types: aluminum and the so-called *mu-metal*. Mu-metal used for ferromagnetic shielding is an alloy of nickel and iron with very high magnetic permeability (i.e., environmental magnetic fields tend to spread along mu-metal layers) [122]. Aluminum layers disperse the energy from external magnetic fields provoking induction of eddy currents due to changing magnetic field (Maxwell's second equation - *the Faraday's Law*) in the walls of the room. This is *passive shielding*. Additionally, *active shielding* can be provided in the form of a system, which measures the external magnetic field and produces a counter-field to cancel out unwanted external fields. This approach has its drawbacks: resonances can easily arise, and there might be a lag around 100 ms between an external field and produced counter-field [33]. Another factor that needs to be controlled during MEG data acquisition are mechanical vibrations that may resemble oscillatory brain activity and cause large artifacts. Vibrations may arise from car traffic, trams, trains, or even from an elevator in the building [123][124].

Whereas general principles of good experimentation should be followed, there are

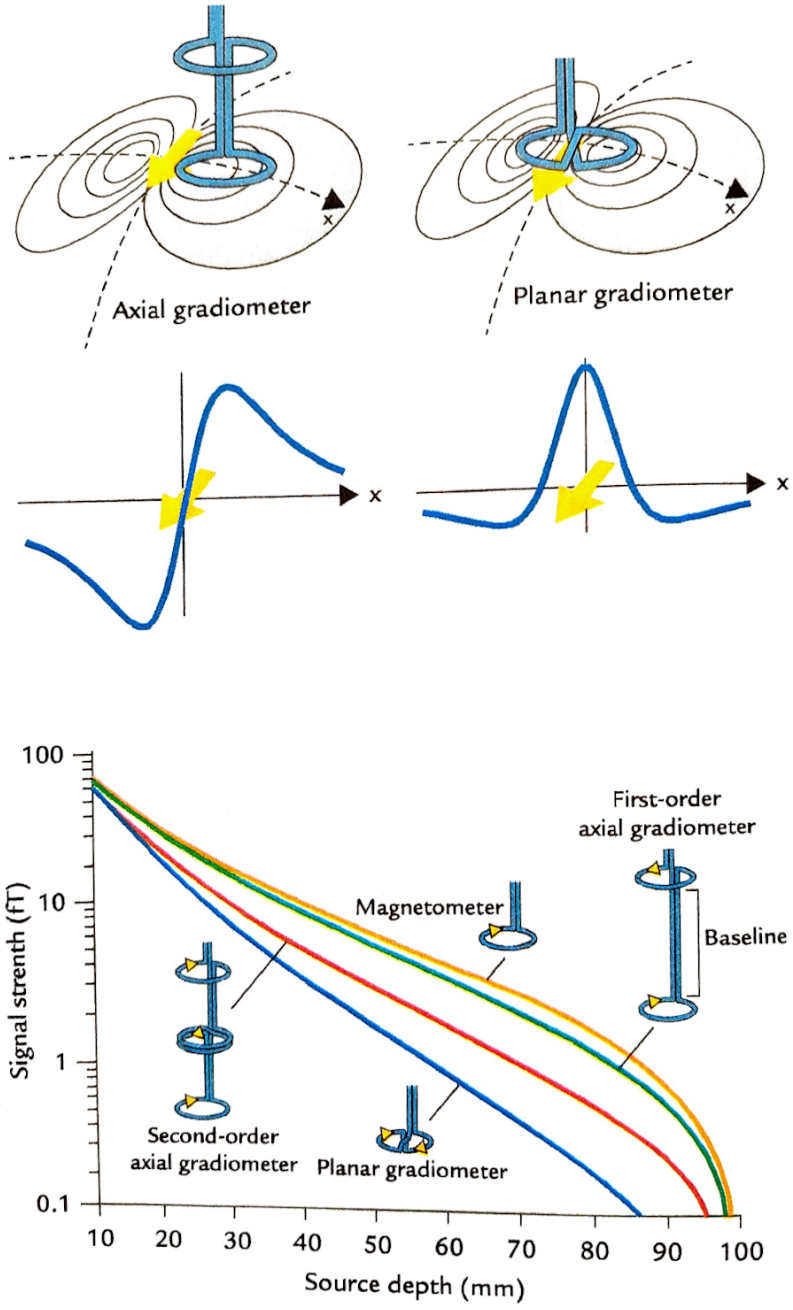


Fig. 1.21. Axial and planar gradiometer sensitivity patterns for the dipolar field pattern produced by a current dipole. The strength of the measured signal (in fT; note the logarithmic scale as a function of source depth in a sphere with 100-mm radius) for the magnetometer will is the largest and for the planar gradiometer is the smallest, for sources at all depths. Source: [33].

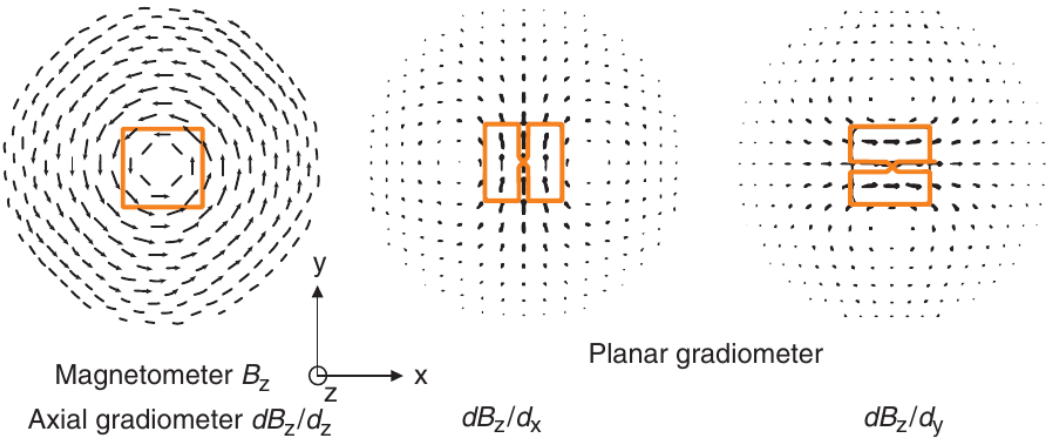


Fig. 1.22. Lead fields of a magnetometer, axial gradiometer, and two orthogonal planar gradiometers with the field components they measure. The approximate radial direction $\hat{r}$ in the sphere is often used instead of $\hat{z}$ when the sensors are arranged into a spherical array. Source: [121].

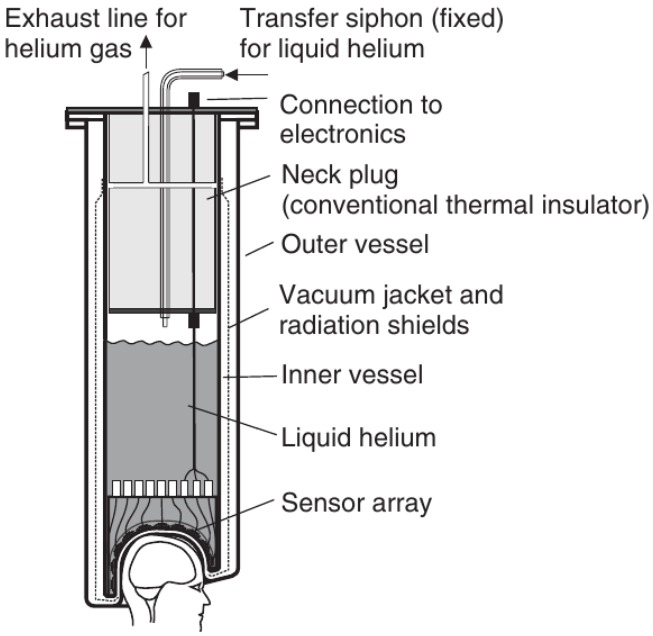


Fig. 1.23. Schematic cross section of a Dewar. Source: [121].

some guidelines specific for MEG acquisition. First, a subjects must not have any magnetic material in/on the clothing (including dental braces and makeup) or the body. If subject participates in the study consisting of both MEG and fMRI recordings, the subject may be magnetized in the MRI scan, so MEG recording should be performed first. Eye artifacts should be controlled by the additional EOG (electro-oculographic) electrodes. Using some head position indicator coils (HPI) is advised to obtain the necessary reference for the sensor positions in space. There is broad literature describing how to optimally set up MEG equipment, prepare subjects properly, and pre-process the acquired data accordingly [123][124], however, the most important aspect is the subject safety - electrical safety must be followed at any time.

Advantages and drawbacks. MEG is totally noninvasive and therefore safe to use in all subjects, including infants. It can be repeated as often as necessary. Acquired signal reflect direct and instantaneous neural activation with an excellent temporal resolution. Some MEG signals are clearly visible without the use of group-level or individual statistics, and clinically valid conclusions can be drawn from single-subject data. Sources of MEG signals can be modeled without reference to the structure of tissues of the head and their relative conductivities. On the contrary, subjects must keep their heads and bodies still during the recording, which is almost impossible in practice, and always some muscle-related activity is confounding the signal. The non-uniqueness of the inverse problem (i.e., knowing the accurate distribution and strength of brain sources provided sensor array measurements) is another major drawback, and ground truth cannot be precisely known. However, interpreting the data relying on plausible anatomical and physiological constraints can still produce useful and accurate enough information. MEG signals are dominated by the most synchronous activation from large neuronal populations (tens of mm^2 of active cortex), so some asynchronous and individually specific activity may not be captured. Intracortical cancellation due to the cortical geometry and cytoarchitecture results in picking only a part of overall brain activity. In MEG, deep and radial sources are less pronounced than superficial and tangential currents, which may cast obtaining information about deeper sources from e.g., the thalamus, amygdala, and hippocampus somewhat challenging. Needless to say that MEG equipment costs much, with the magnetically shielded room being the most expensive part of the whole system.

Future of magnetoencephalography. MEG technology undergoes constant development. Currently, the sensors of multichannel devices are located at least 2 cm above the scalp, and SQUIDs are cooled down to the temperature of 269 °C (4 K) to minimize the evaporation of the liquid helium. This distance makes detection of the fine details of the field patterns almost impossible. Such intensive cooling is

expensive and relatively troublesome. The MEG community is looking forward to a system that would operate at room temperature and sensors placed closer to the human head, ideally to be adjustable to the subject's individual scalp shape. Recently, two technologies look promising - optically pumped magnetometers (OPM; [125]) and fluxgate magnetometers based on yttrium-iron garnet films (YIGM; [126]). Increasing demands of clinical studies stimulate technology development and, to this end, the combined MEG-MRI (magnetic resonance imaging; see Section 1.9) device where ultra-low field (ULF) MRI and MEG are recorded with the same SQUIDs. It allows improved source localization accuracy for purposes such as presurgical evaluation, as both brain structure (with MRI) and function (with MEG) can be mapped without moving the patient and without needing to coregister the multimodal data (i.e., data acquisition based on different physiological phenomena, like primary currents for MEG vs. hydrogen atoms magnetization-relaxation for MRI). Such device is silent during operation, but, compared to conventional MRI, blood-oxygenation-level-based functional MRI (fMRI, see Section 1.9) signals cannot be recorded [127][128].

1.8 Observing oscillations with electroencephalography

During my doctoral research, in one of the my studies I have utilized EEG data [12], so a brief section on electroencephalography is appropriate.

The year 1924, when the very first EEG was recorded, is traditionally considered the advent of electroencephalography. Several noninvasive neuroimaging techniques have been developed over the past century, and EEG is, to this day, still one of the most popular tools. The reader interested in elaborated description is referred to the following publications: [74][33].

Synchronous spiking neural assemblies generate electromagnetic field disturbances visible to both MEG and EEG, howesver, the measurement method differs (Eq. 1.7 and 1.8). Contrary to MEG, EEG records activity from not only tangentially oriented sources but also radial dipoles as well [33].

The information available from electromagnetic measurements of the human brain activity is limited not only due to the inevitable intracortical cancellation (partial alignment and synchrony of neural ensembles) but also due to the volume conduction — a passive spread of currents aberrated by the resistivity of the medium (body tissues) which affects measured electric potentials. Magnetic field is not distorted by medium conductance inhomogeneities [33], rendering MEG a more robust method in this regard.

As in MEG, net current formulation holds for EEG as well (Eq. 1.9), permitting analyzing signals with equivalent current dipoles approach and linear model expressed

by Eq. 1.6, thus enabling source reconstruction and localization. Both similarities and differences between MEG and EEG shape the waves visible in measurements (Fig. 1.14), however, EEG recording is heavily affected by the used electrode montage and chosen reference [129].

A typical modern EEG system consists of: the cap with electrodes, the differential signal amplifier, the analog filter ensemble, the analog-to-digital converter (ADC), the processing unit (usually a computer or microprocessor), digital data storage, and the display. Electrode placement is standardized according to the 10-20 system and its derivatives. Electrode names refer to particular site given electrode covers: Fp (frontopolar), AF (anterio-frontal), F (frontal), FT (fronto-temporal), FC (fronto-central), T (temporal), C (central), TP (temporo-parietal), CP (centro-parietal), P (parietal), O (occipital). Additional electrodes are: GND (ground), REF (reference electrode), Iz (inion). Letters are accompanied by odd and even numbers. Odd numbers refer to the left brain hemisphere, and even numbers refer to the right brain hemisphere (Fig. 1.24). Electrodes should have low-impedance contact with the scalp to prevent spurious results. The differential amplifier provides the proper measurement of the voltage between the potential of a given electrode and the REF electrode. Filters condition the signal before digitization in the ADC and prevent so-called *aliasing* [74].

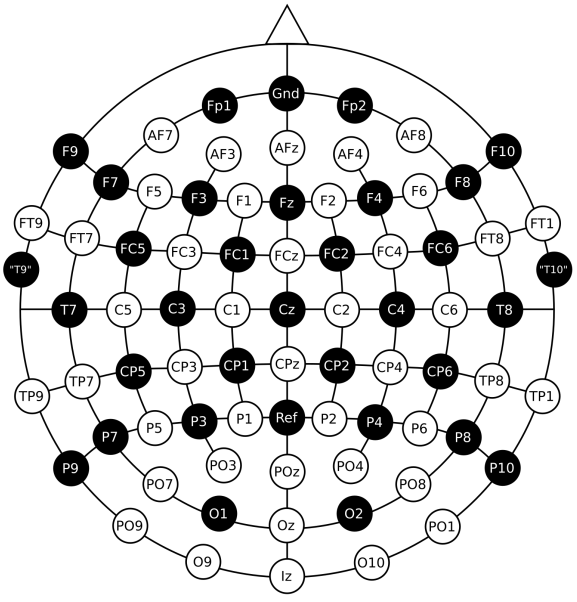


Fig. 1.24. EEG electrodes placement according to the 10-20 system.

Typical signal amplitudes lie within a range from 10 to 100 μV. Usually, EEG signals contain useful information within the 0.1–100 Hz frequency band [130]. Both MEG and EEG offer an excellent temporal resolution and reflect direct and instantaneous

neural activation. With some expertise, substantial insight may be obtained by visual inspection (e.g., epileptic disorders, eye movement, sleep spindles). Both techniques are sensitive to the movement of the head or limbs and need to be recorded in a specially prepared room that shields from external noise. Source localization and reconstruction are difficult due to the ill-posedness of the inverse problem and sources interference confounding activity of interest — an issue especially pronounced in EEG.

1.9 Functional magnetic resonance imaging

Functional magnetic resonance imaging (fMRI) is the prevailing technique in modern analysis of the human brain. Many of the studies reviewed in the next chapter rely on this method, so it is worth giving it some attention.

FMRI allows measuring brain activity by detecting blood flow fluctuations. It combines structural tissue image acquisition enabled by an MRI scanner with the knowledge of metabolic changes in the brain. The technique was invented in the early 1990s by Seji Ogawa's group at Bell Laboratories [131].

Quantum magnetic resonance process is the fundamental principle behind MRI scanner operation (Fig. 1.25). First, a static magnetic field $\vec{B}_0$ of magnitude 1.5, 3, or

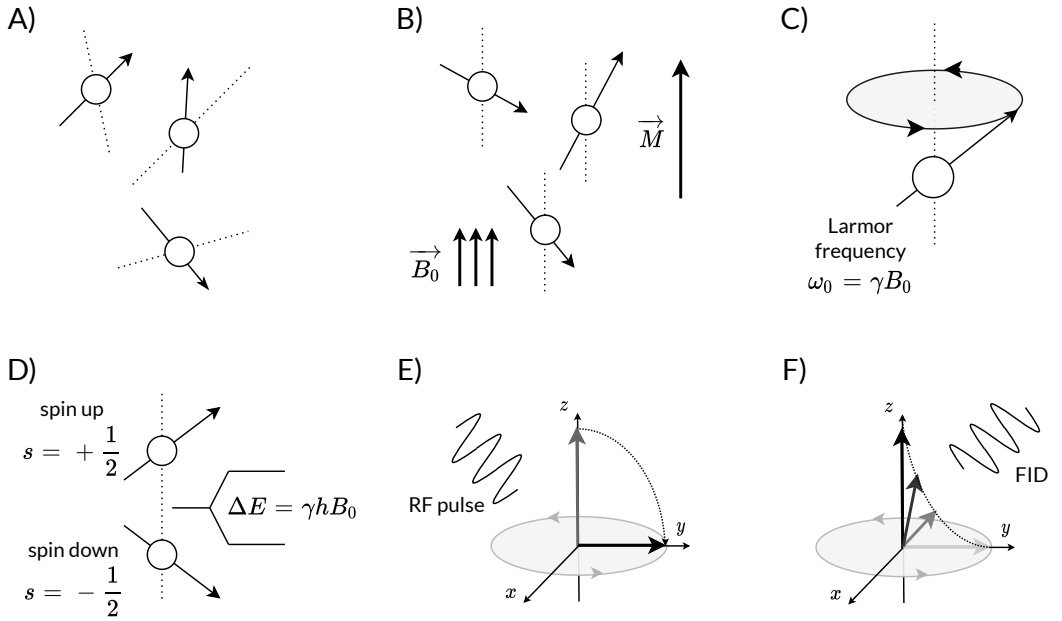


Fig. 1.25. Magnetic resonance imaging physics. **A)** Random nuclear spin orientation. **B)** Applied external magnetic field leads to spin alignment and net magnetization $\vec{M}$. **C)** Larmor precession. **D)** Zeeman effect for hydrogen nucleus. **E)** RF pulse flipping net magnetization vector. **F)** Relaxation process generates free induction decay (FID) signal.

7 Tesla is produced (Fig. 1.25B). A table slides the patient into the field. Hydrogen atoms' nuclei inside the subject's brain, composed of $\sim 75\%$ of water [132], align their spin with the static field's direction. As a result, spins precess around the external field axis with *Larmor frequency*:

$$\omega_0 = \gamma B_0 \tag{1.12}$$

where γ is gyromagnetic ratio for hydrogen nucleus equal to 42.58 $\frac{\text{MHz}}{\text{T}}$, and B_0 is the magnitude of the static field.

In the absence of an external magnetic field, the random orientations of individual spins cancel out (Fig. 1.25A), and $|\vec{M}| = 0$. However, when a static field is present, $|\vec{M}| \neq 0$ because in the equilibrium state, due to the *Zeeman effect* (Fig. 1.25D), populations of high-energy state and low-energy state are uneven [133]. External magnetic field has two components:

$$\vec{M} = \vec{M}_{\parallel} + \vec{M}_{\perp} \tag{1.13}$$

where $\vec{M}_{\parallel}$ is longitudinal magnetization, and $\vec{M}_{\perp}$ is transversal magnetization vector. In the equilibrium, $\vec{M}_{\perp} = 0$ because precession of individual spins is not yet synchronized.

Then, radiofrequency coils emit an electromagnetic wave called *RF pulse* that matches the Larmor frequency of hydrogen nuclei (Fig. 1.25E). This causes resonance. The emitted pulse affects net magnetization vector $\vec{M}$ in two ways. On the one hand, a subpopulation of nuclei changes from the low-energy state to the high-energy state. As more spins are aligned antiparallel to the field, longitudinal magnetization $\vec{M}_{\parallel}$ decreases. On the other hand, EF pulse causes spins to precess in phase, which builds up transversal magnetization $\vec{M}_{\perp}$. Parameters of the RF pulse can be adjusted to affect net magnetization in the desired way.

Next, RF pulses are turned off, and the spontaneous *relaxation* begins, bringing nuclei to the equilibrium state again. During this process, energy is released from the brain tissue in the form of electromagnetic waves, measured by receiver coils as *free induction decay* signals, FID (Fig. 1.25F). Relaxation time of longitudinal magnetization $\vec{M}_{\parallel}$ is denoted as T1, and relaxation time of transversal magnetization $\vec{M}_{\perp}$ is called T2. In real systems, local inhomogeneities of the magnetic field cause spins to dephase more rapidly than T2 suggests, i.e., in shorter relaxation time T2*:

$$\frac{1}{\text{T2}^*} = \frac{1}{\text{T2}} + \gamma \Delta B_0 \tag{1.14}$$

where ΔB_0 is the difference in local field strengths [134]. Different tissues are characterized by specific $T1$ and $T2$ relaxation times. By adjusting the *sequence* of RF pulses,

one can match the contrast of produced images to emphasize desired tissues.

Using MRI imaging, researchers can explore the human brain structure. Brain functional images can be acquired with the fMRI technique, utilizing the so-called *spin-echo (SE) sequence*, which allows measuring a concentration of oxygenated and deoxygenated hemoglobin, having different magnetic properties. Neuronal activity can be then deduced from T2*-weighted images that reflect expected *blood-oxygenation-level-dependent (BOLD)* changes following increased neuronal activity [127].

2 Oscillation Fingerprinting

Dip your finger into an ink-pad and press it against the paper. No two people in the world will produce the same image. Even monozygotic twins are not identical as the similarity of series of environmental pressures they are exposed to diverge in time. Our brains interacting with the external world reconfigure its connectome, hence its rhythmic activity is unique for each of us.

Can human brain oscillations provide unequivocal means to recognize different states of our brain? We know that its function is derivative of its architecture. Though, if there are no two identical people, there are no two identical brains, both in terms of anatomy and function. A growing body of research found under the umbrella of *brain fingerprinting*, aside from the context of biometry [135][136][137] or forensic science [138], promises to understand how the human brain executes cognitive functions by learning how the brain activity is reflected in signals recorded by the most commonly used contemporary neuroimaging methods. More specifically, is it possible to pinpoint the exact pattern in the recorded signal correlated with a specific cognitive process by linking it to the activity of a single brain structure (or network of interacting structures) manipulated through a carefully devised experiment (Fig. 2.1)?

It is hard to name what precisely brain fingerprint is and what is not. This field is not mature enough to provide a cohesive and precise picture; however recently, over ten years accumulating reports aligned to show some exciting possibilities. Different methodologies are focused on specific modalities, specific types of fingerprint, sometimes constitute a mix of previously devised methods, reinforcing each other, slowly building complementary view on to what extent we understand what is reflected in the acquired signal and what appears to be noise — an artifact of our hopefully shrinking ignorance. There may be atlases and databases that humans or computers can use to learn what to look for in the recorded signal to say with confidence that a given region or network is engaged or given brain disease or cognitive process is present. It would be an extremely helpful tool for science and clinical practice if such brain fingerprinting algorithms and databases were devised for mobile and relatively cheap equipment like EEG, without needing a much grander MRI scanner. We are still at the beginning of the way. Currently pioneered methods need to be improved and carefully tested before they can take part in a bigger commercial system [7].

The following paragraphs will be devoted to describing pieces of the whole. Presented approaches need to be treated as propositions, not as complete, furnished robust solutions, at least for now. They were tested in narrow conditions, so any generalizations may be made with care as the field still develops. Then, in the next chapter, I will focus

entirely on the central approach discussed in this thesis — Spectral Fingerprinting. With this understanding, a description of the proposed toolbox and its exemplary application can be placed in the proper context, facilitating better understanding.

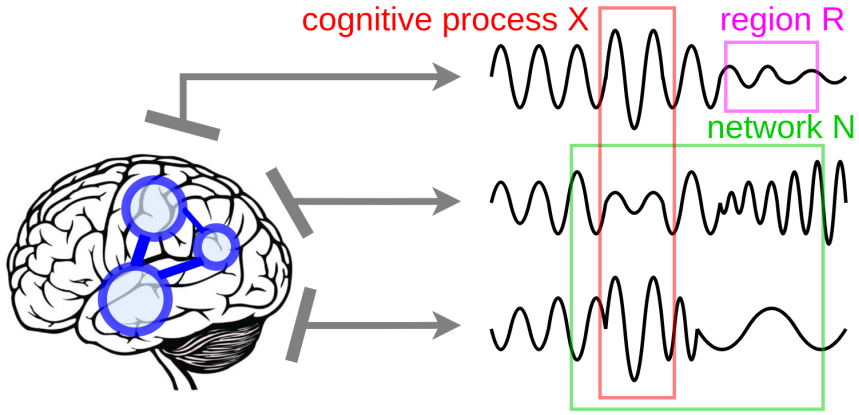


Fig. 2.1. Brain activity organized as a cooperation of modular networks propagates to a sensor array producing readout to be interpreted by the hypothetical brain fingerprinting battery. It can detect time-localized patterns in the measured signal that are hallmarks of evolving cognitive process. Each component of this battery can be mastered in searching fingerprints of a specific type: regional fingerprints, regional cooperation (networks) fingerprints, markers of engagement in the specific cognitive process. Note that these patterns can parallel, interweave, disappear and reappear aperiodically or in a regular repetitive fashion. Joint interpretation provides a synergistic view of the state brain is engaged at the moment.

2.1 Source-reconstructed fingerprints

The most straightforward way of investigating the activity of cortical and subcortical neurons would be to design a transformation that maps a sensor-level time-series into a source-level time-series. There is a whole family of *source localization and reconstruction* tools doing precisely this. Their main goal is to estimate the most probable activity pattern of signal generators, whose location is to be estimated (localization followed by reconstruction) or given a priori (reconstruction only). The main idea is captured by Eq. 1.6, where the sensor signal matrix is assumed to be a linear mixture of the underlying neural oscillators. Different approaches vary by the number of assumptions. For *blind source separation* (BSS) methods, the sole assumption is the independence of the obtained signal components (e.g., Independent Component Analysis, ICA, [139][140]). Additional constraints allowing the passage of signals from the location of interest while attenuating signals from other locations and controlling for correlated confounding activity are characteristic for beamforming family of methods like *linearly constrained minimum variance* filter (LCMV, [141][142]), its frequency domain equivalent *dynamic*

imaging of coherent sources (DICS, [143]), or frameworks offering interference-free estimation like *minimum-variance pseudo-unbiased reduced-rank* filter (MV-PURE, [144][145][146]).

Source analysis techniques have proven to be beneficial. For example, Markus Siegel, in 2008, used the frequency-domain beamforming method (DICS) to discover spectral signatures of human's covert attention (i.e., shifting focus to the target without directly looking at it). They constructed the "lateralization index" marker that can predict the direction of attentional shifts by analyzing magnetoencephalographic time-series of subjects engaged in covert attention orienting task [147]. With the same source reconstruction method and the same modality (MEG), Van Dijk and colleagues localized alpha band (approx. 10 Hz) modulations in the posterior cortex to be correlated with the ability to perceive visual contrasts [148]. Beamforming electroencephalographic data (EEG), in turn, provided encouraging results with clinical diagnosis potential. In 2013, Lydia Elshoff and her collaborators used dynamic imaging of coherent sources to reveal different patterns of regional connectivity underlying epileptic seizures generation and propagation. The proposed pipeline still has some limitations and needs further research before being approved for medical use [149].

2.2 Brain areas acting in concert — resting state networks

However, this work emphasized another aspect outlined by reviews on cortical dynamics - the need for studying regional interactions. Cognitive processes, healthy or impaired, are unlikely to engage a single brain region but occur as the outcome of cooperation of networks coordinated by other networks. As Wolf Singer concluded in his review [150]: *"Recent insights justify considering the brain as a complex, self-organized system with nonlinear dynamics in which principles of distributed, parallel processing coexist with serial operations within highly interconnected networks.* " In 1990s researchers started to encounter more and more repeating coherent activation patterns consistent across subjects, recognized decade later as the *resting-state networks* (RSNs, [151][152]). They were found mainly by analyzing correlation patterns of signals acquired with *functional magnetic resonance imaging* (fMRI; see Section 1.9) from resting subjects, and they comprise seven to seventeen networks [153][154]. In my work, I refer to the division proposed by Yeo and colleagues [153], consisting of seven main resting-state networks found common for $N = 1000$ subjects' fMRI data (Fig. 2.2) :

- Somatomotor Network (SMN),
- Visual Network (VIS),
- Dorsal Attention Network (DAN),

- Cingulo-opercular Network (CON),
- Frontoparietal Network (FPN),
- Limbic Network (LIM),
- Default Mode Network (DMN).

For a long time, the enthusiasm about these networks to be found by electromagnetic brain imaging remained limited. A breakthrough occurred at the end of the first decade of the new millennia, when advances in source imaging methods, together with high-density electroencephalography (hdEEG), led to the development of methods large-scale networks reconstruction by more direct oscillation-based modalities. In 2015, Yuan and collaborators invented a pipeline based on the popular inverse problem solution coined as *low-resolution brain electromagnetic tomography* (LORETA, [155]) to extract RSNs from EEG, validating it with fMRI-based evaluation [156]. This achievement was replicater later by Liu and collaborators publishing two articles about networks reconstruction from EEG [157][65]. Similarly, Marino et al., 2019, discovered the same resting-state patterns using another source analysis method (*minimum norm estimate*, MNE, [158]). Concurrently, in magnetoencephalography, detection of resting-state functional networks became possible, using blind source separation methods and signal envelope correlations [159][160]. This is not surprising as MEG and EEG are two similar acquisition methods relying on the same neural principles.

2.3 Connectivity-based fingerprints

One may ask, how does one know whether obtained localizations and patterns of activity are factual? Is there any way to validate found generators of EEG/MEG waves? Functional magnetic resonance imaging (fMRI) provides the most spatially accurate view of active neural populations for the cost of poor time resolution. It is no coincidence that fMRI is one of the most frequently used techniques in brain fingerprinting, and most often, the first step in designed pipelines is to estimate functional dependencies between brain regions of interest by means of so-called *functional connectivity* (Fig. 2.3).

Functional connectivity is defined as a similarity between neural time-series [161], most commonly measured using Pearson's correlation coefficient, which calculates the linear relationship between pairs of time series. Pearson's correlation may take values from -1 to 1. When the correlation is high (close to 1), one can speak about synchronization, in-phase coupling between signals or high functional connectivity. When the correlation is close to zero — no relationship between signals, and when the correlation is highly negative (close to -1) — an out-of-phase relationship. In terms of functional connectivity, negative connections can be justified from a biological

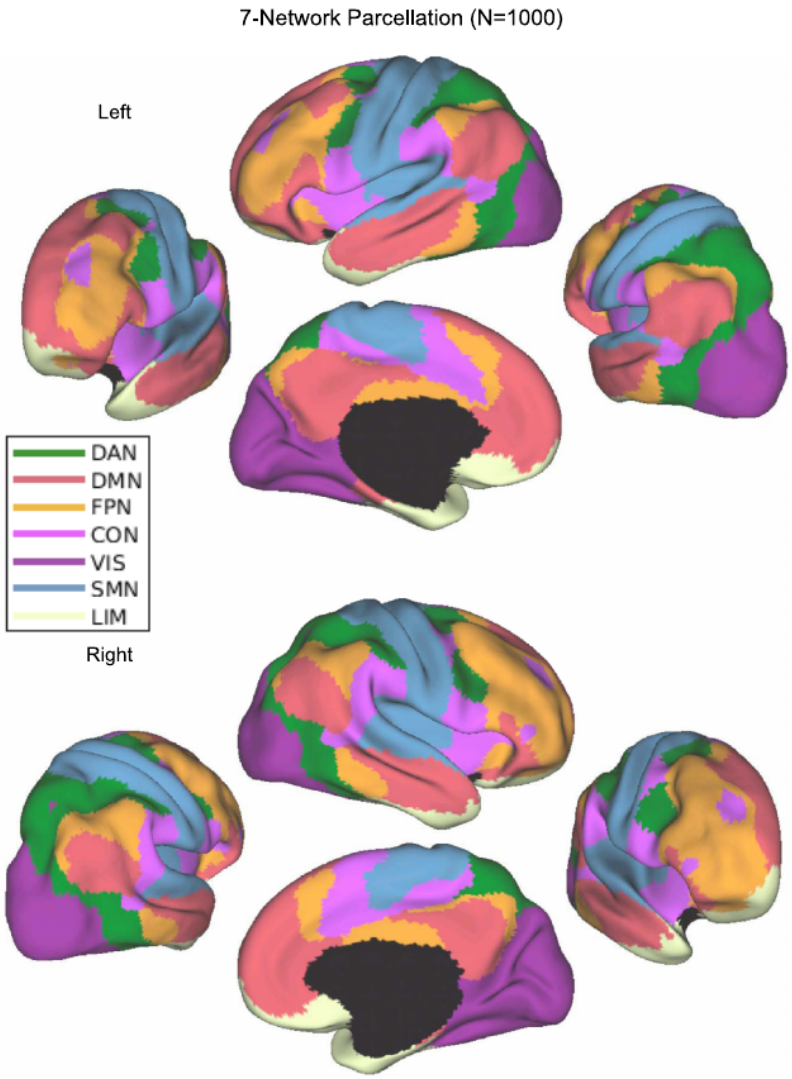


Fig. 2.2. A coarse (7-network) parcellation of the human cerebral cortex based on clustering functional magnetic resonance imaging resting-state data of 1,000 healthy subjects. Abbreviations stand for the following networks: Dorsal Attention Network (DAN), Default Mode Network (DMN), Frontoparietal Network (FPN), Cingulo-opercular Network (CON), Visual Network (VIS), Somatomotor Network (SMN), Limbic Network (LIM). Source: [153].

perspective but sometimes are also considered a side-effect of the data processing procedures and dropped in further analyses [161].

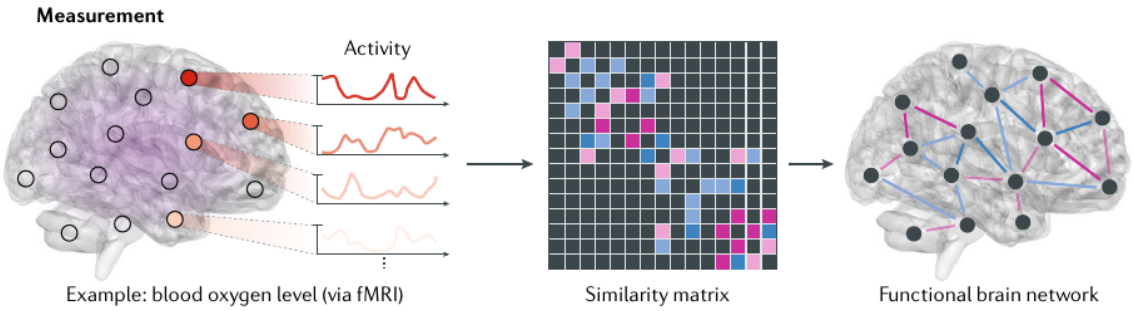


Fig. 2.3. After defining brain areas of interest, one can estimate the similarity between time-series in the form of a correlation matrix called the *functional connectivity* matrix. It can be used in turn to model brain activity as a graph. Source: [162].

Luckily for biometrics, it turned out that individual differences between human functional connectivity are unique enough to serve as a "personal" fingerprint. The success rate of connectivity-based identification of individual subjects reached over 90% [136][137]. Another interesting study focused on finding functional connectome fingerprints of mental diseases. The examination of functional connectivity matrices from over 1,000 individuals (one healthy group and other groups suffering from different mental diseases: unipolar depression, bipolar disorder, psychosis, schizophrenia) revealed consistent features characteristic for illness [163][7].

The idea of fingerprinting particular brain regions was employed to construct data-driven anatomical-functional parcellation of the human cortex. As reported in Nature 2016, renowned and frequently cited work from Glasser et al., proposed a data-driven method of obtaining the robust multi-modal fingerprint of each cortical area, providing a common framework for future research [164]. One year later, a fingerprint of the information transfer between regions and resting-state networks appeared, extending already impressive achievements made possible with functional connectivity [165]. Interestingly, some processing pipelines based on source reconstruction algorithms provide a way to estimate quantity similar to functional connectivity but using more accessible, mobile, and cheaper hardware. Hipp and his collaborators recovered the functional connectivity matrix by beamforming the MEG signal [166]. Similar achievement is possible using the frequency-domain equivalent of the same beamformer [167].

2.4 Graph-theoretical fingerprints

Experts in the field were always conscious about the importance of studying not only regions themselves but also regional short-range and long-range dependencies [8][168].

Graph-theoretical approaches appeal as a natural way of conceptualizing neuronal interactions. In this formalism, chosen regions of interest are represented by graph nodes, and graph edges model interactions between regions. Weights of the edges are often quantified by calculating correlation or synchrony measures between pairs of nodes (their time-series).

Statistically significant differences in graph parameters that model brain states from contrasting thoughtfully designed experimental conditions can reveal fingerprints of underlying cognitive processes. For example, an interesting study from 2015 by Bola and Sabel employed graph-theoretical analysis of source-reconstructed activity recorded with electroencephalograph during an oddball visual task (a subject needed to press a button when the target stimulus is displayed on the screen, and ignore non-target stimuli; Fig. 2.4). Results revealed significant changes in certain graph parameters correlated to the occurrence of the attention shift [169]. Similar to other approaches, it

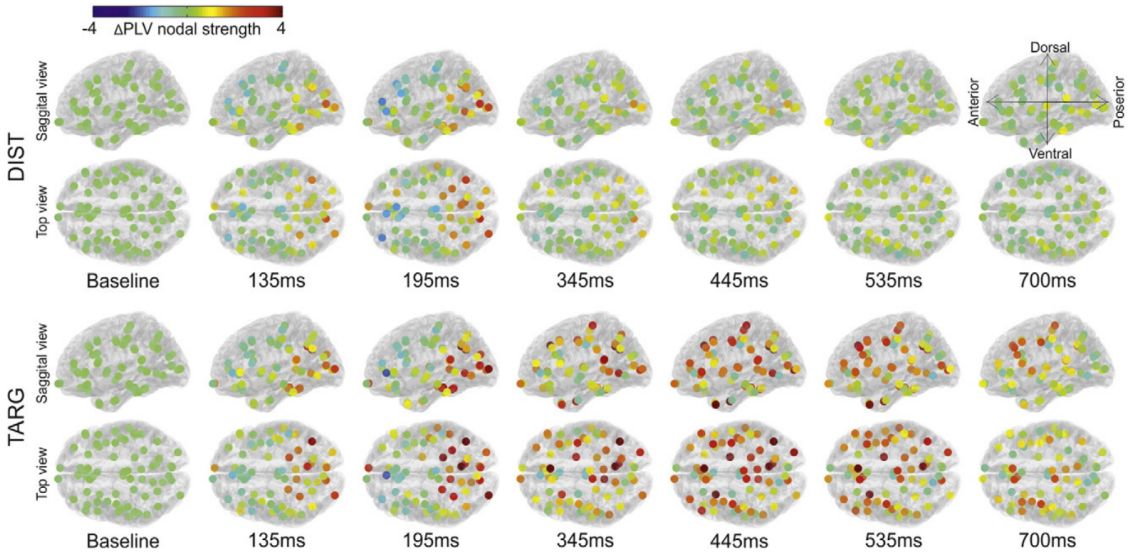


Fig. 2.4. Progression of the changes in synchrony across the cortex between two experimental conditions: response to the distractive stimuli (DIST) and to the target stimuli (TARG). Color depicts the relative change in synchrony measured by Phase Locking Value (PLV, [170]). Attending to the target stimuli produced a massive wave of increasing synchronicity between graph nodes, flowing from the posterior to the anterior parts of the cortex. Source: [169].

is possible to discriminate between subjects using graphs constructed from anatomical and functional patterns of connectivity measured by MRI [171], which may be attractive in biometric applications.

Interesting discoveries were also made with mixed methodologies taking advantage of functional connectivity estimation followed by graph-theoretical modeling. Following this strategy, Fenna Krienen and her collaborators discriminated between fourteen different task states and viewed them as task-specific fluctuations related to the central

core pattern of activation that happens to act as the attractor of the underlying dynamical system. Such an approach can inspire discovering robust fingerprints of stable brain states [172]. In a similar vein, in 2018, Ciric et al., presented a framework for tracking dynamical changes in typical connectivity between nodes of the Default Mode Network that allows for rendering functional changes in a particular network as a sequence of well-defined states (Fig. 2.5; [173]).

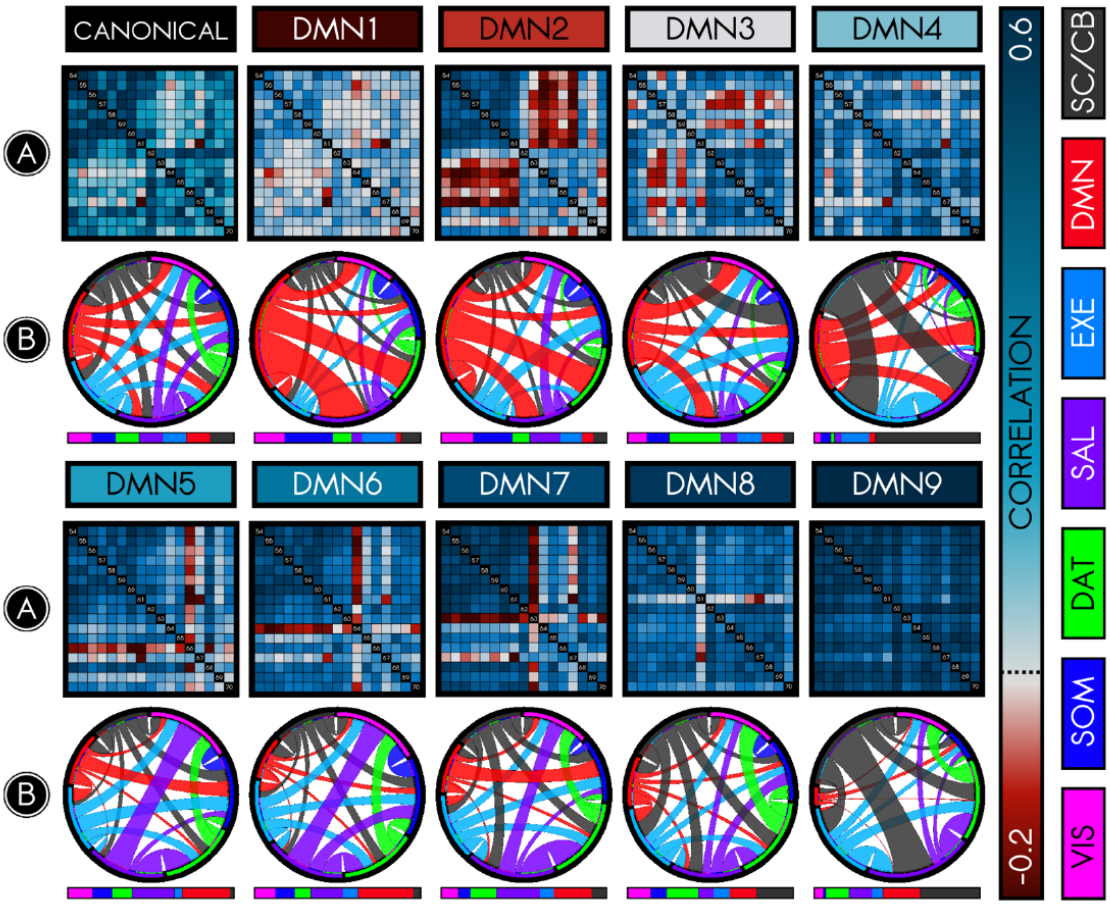


Fig. 2.5. Connectivity dynamics of the Default Mode Network (DMN). Connectivity pattern of the canonical (time-averaged) DMN compared to nine observed states. **(A)** The connectivity among seventeen nodes in the DMN. **(B)** Relative connectivity between one given state of the DMN (from a total of 10, including canonical) and the rest of the networks: Subcortical/Cerebellum Network (SC/CB), Executive Control Network (EXE), Cingulo-Opercular/Salience Network (SAL), Dorsal Attention Network (DAT), Somatomotor Network (SOM), Visual Network (VIS). Source: [173].

2.5 Spectral approaches

Spiking of neural populations produces oscillatory patterns visible as brain rhythms and event-related synchronization/desynchronization (see Section 1.5). Thus, considering spectral descriptions to construct useful brain fingerprints seems natural. Previously

mentioned studies, where dynamic imaging of coherent sources (DICS) was employed, are good examples of following that strategy. It turned out that this way it is possible to detect attention shifts [147], perception of visual contrasts [148], epileptic seizure generators [149], and mapping the functional connectivity in general [167] in a relatively cheap and mobile setting.

These works count as an essential contribution to the body of knowledge about brain fingerprinting, however first attempts to define spectral fingerprints (not in the biometric sense) were undertaken by Donner and Siegel in 2011, where brain activity was divided into two classes of possible patterns: short-range local gamma-oscillations and long-range interaction based on the beta rhythms [8]. Siegel further synthesized these notions to define a more general framework, encompassing more than only gamma and beta rhythms, where spectral fingerprints are coherent patterns of oscillations mediating between local activity underlying cognitive variables encoding different (e.g., sensory) features of human experience, and networks that interact with the same frequency are the basic building blocks of neural processing called canonical computations (Fig. 2.6, [168]).

Inspired by that idea, Keitel and Gross in 2016 conceived a method of combining magnetoencephalographic multichannel signals, representing the human brain oscillatory activity, and anatomical structure of the brain into sets of spectra (Fig. 2.7) that were intended to serve as meaningful regional signatures, allowing a robust recognition of given brain areas through their characteristic spectral patterns. Spectral fingerprints defined this way turned out to be stable biomarkers that are sufficiently specific to permit the successful identification of brain regions. Moreover, spectral profiles' peaks, which correspond to the natural frequencies of regions of interest (ROIs), are consistently modulated by specific tasks, neurological or mental disorders, and can be generalized across groups of participants [9][11].

Our research group was encouraged by those achievements and decided to investigate further the possibilities offered by the Spectral Fingerprinting framework, which eventually led to the release of the *Toolbox for Frequency-based Fingerprinting* that is discussed in this thesis [10][12].

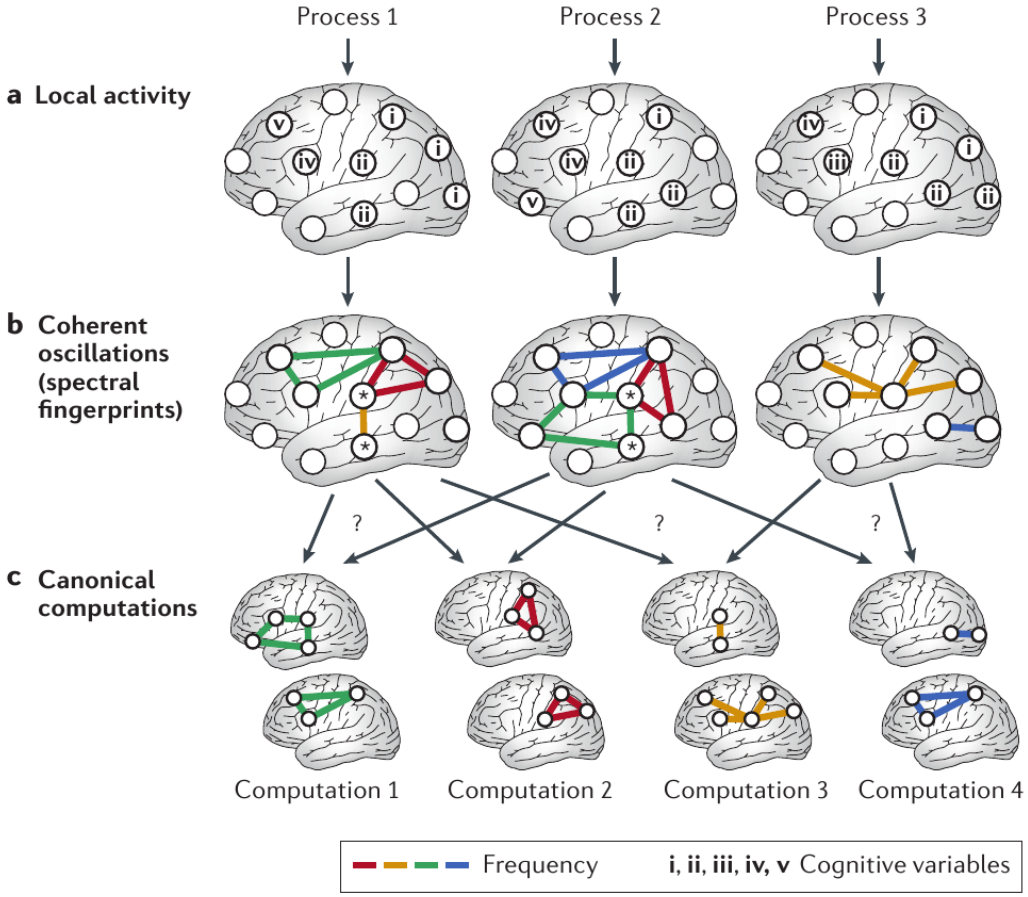


Fig. 2.6. Large-scale spectral fingerprints of cognitive process framework proposed by Siegel et al., 2012. Local activity accompanying encoding of different features of cognition (cognitive variables) is coordinated with frequency-specific coherent connection patterns called *spectral fingerprints*. Frequency-specific interactions identify corresponding canonical computations that underlie cognitive processes. Source: [168].

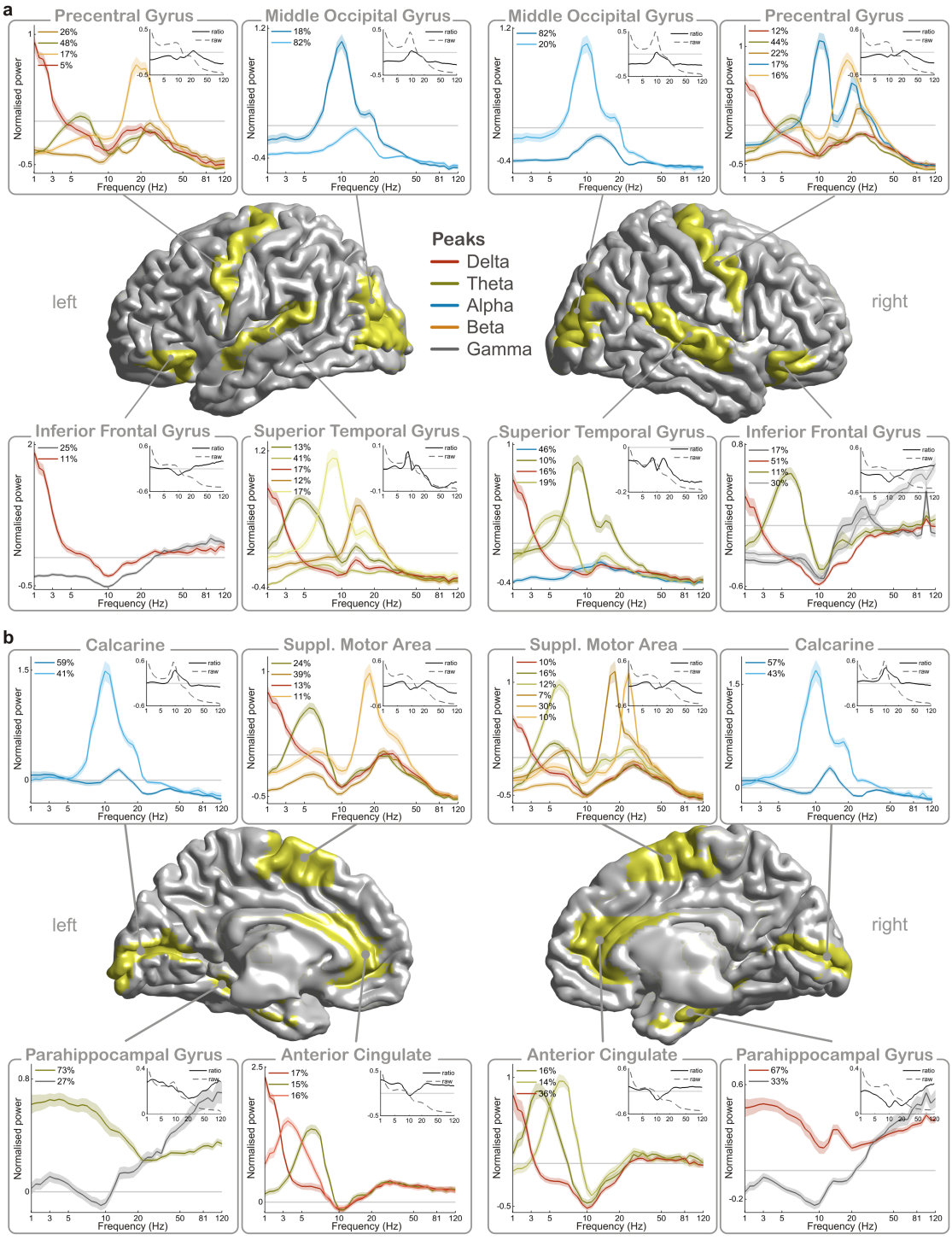


Fig. 2.7. Resting-state spectral fingerprints discovered by Keitel and Gross [9] for different anatomical areas in the 1–120 Hz frequency interval. The lines are color-coded for the respective frequency bands according to peak frequency. Legends show the corresponding duration of each spectral mode (i.e., the percentage of trials in which each spectrum was present on average during recording). Y-axis depicts the power normalized in relation to the average spectrum of the whole brain. Shaded regions depict the standard error of the mean across participants. Inlets show average power spectra for respective areas without normalization (dotted lines) and with ratio normalization (continuous lines). Source: [9].

3 Spectral Fingerprinting

In this chapter, a particular brain fingerprinting strategy will be presented — Spectral Fingerprinting (Fig. 3.1; [9]). It is the central method in my doctoral research. The background, aims, and results of the seminal work will be presented. Moreover, a follow-up report from Lubinus and collaborators [11] has shown usefulness of Spectral Fingerprinting in the context of cognitive deficits. My first publication is another work that confirms the method's reproducibility in line with the results of Lubinus [12]. The material in this chapter will provide a proper context for the next chapter where the Toolbox for Frequency-based Fingerprinting will be presented [10].

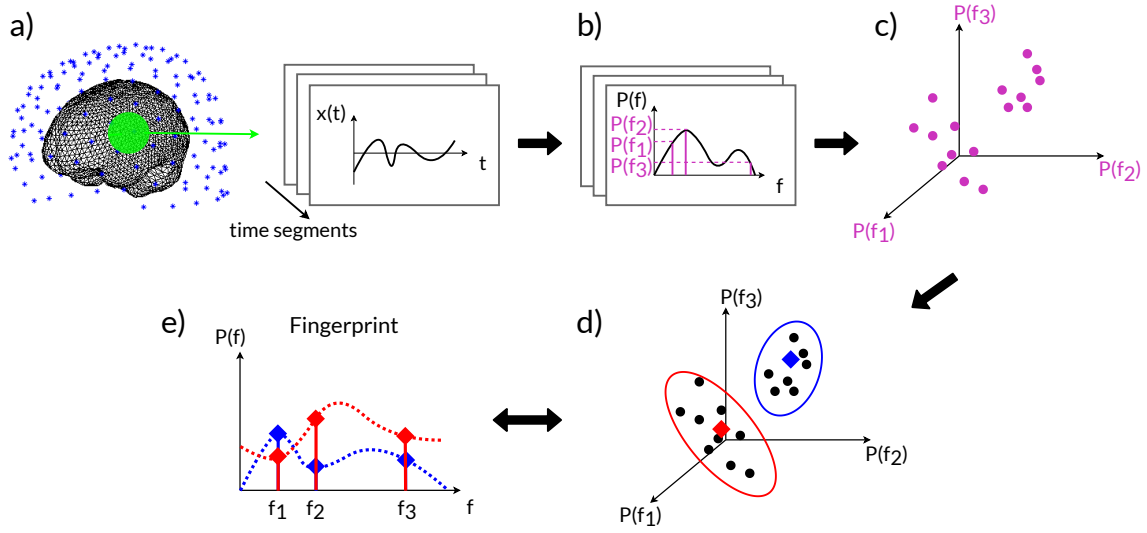


Fig. 3.1. Illustration of the Spectral Fingerprinting algorithm [9]. **a)** Mean-field, electrical activity of the brain is recorded via an array of sensors (blue dots). A beamformer [174] solves the so-called *inverse problem*, thus enabling reconstruction of the source activity in selected voxels constituting chosen brain regions of interest (green area). Source-level time series are cut into segments of equal length. **b)** Power within each segment is estimated for a given set of frequencies of interest (here, three frequencies, f_1, f_2, f_3, are shown for readability) and then averaged across locations inside the region of interest. **c)** Power spectrum of each segment is represented as a point in a n-dimensional frequency space, where the power of the selected frequencies provides coordinates of the point. **d)** Segments are clustered together. Each centroid is equivalent to one **e)** *spectral mode* of a given brain region and depicted as an interpolant curve spanned between the frequencies of interest in the resulting fingerprint (interpolation type is arbitrary, as it serves visualization purpose only).

3.1 Background

The human brain has non-uniform anatomy, both locally and globally. Locally, the alignment of neural tissues (cytoarchitecture) differs across brain areas, and globally each region has a specific structural connectivity pattern [175].

Anatomy shapes the functional properties of the brain [168]. Individual area function and interaction between areas depend on their internal structure.

Various cognitive demands and the ongoing regulation of physiology force areas to cooperate [176][177][178]. It is reflected in electrophysiological measurements. The correlated activity mostly accompanies regions that share a similar function. One can discover networks of areas acting in concert by searching for spectral power correlations in MEG/EEG signals [159][179][166][180][181][182][183]. Synchrony measures are often used for that purpose as well [170][169][184][185]. Several studies discovered hallmarks of networks activity in resting-state recordings [2][186][187][188]. Interestingly, the dynamics of resting-state activity mirror the functional architecture of cortical networks [150].

Keitel and Gross stated in their article: "*while studying the global network level has dramatically improved our understanding of functional resting states, we still know surprisingly little at the level of individual brain areas*" [9]. Research on oscillatory preferences of single brain areas [73] and previous attempts to characterize patterns of frequencies across the cortex [180] convinced the authors to look for regional "signatures" of resting-state activity — spectral fingerprints.

The study's first aim was to show the local functional specificity of brain areas. Anatomically defined parcels [189] were chosen from the many possible criteria to divide the brain into regions. The method described below indeed allowed to capture the unique identity of different areas, as was proven by the results of the identification procedure. Moreover, found representation generalized across participants. Inter-subject variability was shown to follow some common patterns, in other words.

The second aim was to alleviate common methodological problems of previously proposed approaches. Spectral Fingerprinting was not based on averaging power spectra inside frequency bins. Clustering of non-overlapping segments (single-trials) of activity was performed instead. This way, the non-stationary nature of brain oscillations was taken into account. Moreover, beamforming technique was used to find source activity underlying recorded sensor signal. Many studies have not accounted for the presence of the $1/f$ trend present in the human brain power spectrum (see Section 1.6), limiting the explanatory role of frequencies other than alpha. Here, the spectral normalization procedure diminishes the $1/f$ tendency and makes spectra comparable across participants.

Results from Keitel and Gross also show that individual characteristics of brain areas conform to the brain network framework mentioned above. Similarities between regional fingerprints form patterns resembling large-scale networks.

The next paragraphs will guide through the study's methodology and results. A

more technically inclined audience can find details of the used methods in Appendix A. Overview of the reported results will be accompanied by the description of two other published works using the Spectral Fingerprinting approach, including the initial article related to the method reproduction [12]. In Chapter 4, my ultimate contribution in the form of Toolbox for Frequency Fingerprinting will be presented [10].

3.2 Methodology

Keitel and Gross invented the Spectral Fingerprinting pipeline (Fig. 3.2; Algorithm 7 in Appendix A). It models individual brain region activity as a set of spectral curves. Each member of this set is called *spectral mode*, or *mode* for short. A set of modes acts as a region's "signature."

Data collection during the experiment was straightforward. A total of $N = 22$ healthy adults were invited to a magnetoencephalographic acquisition session. Participants were instructed to sit upright and fixate on a cross projected centrally on a screen. Approximately 7 minutes of 248-channel MEG resting-state signal was collected. Data were cleaned from artifacts and visually validated. Additionally, images of individual brain structure were needed. Structural magnetic resonance images (MRI; see Section 1.9) were collected. MRI and MEG coordinate systems were matched (so-called *co-registration*) to enable projecting sensor signal onto a source space, where brain structure was defined. Each single source location (2757 locations in total) was represented by a single cell of a three-dimensional 10 mm^3 grid. Sources were grouped into 115 non-overlapping brain areas using the Automatic Anatomical Labeling atlas (AAL; [189]). LCMV beamformer [142] projected sensor time-series onto source-space reconstructed time-series. Reconstructed signals were divided into 1-second long segments (Fig. 3.1a) and further transformed into Fourier spectral power spectra. This resulted in 466 ± 20 "snapshots" per participant of his/her brain state. The frequency axis consisted of 42 logarithmically spaced discrete values spanned across 1-120 Hz band (Fig. 3.1b). Regional spectral power segments were obtained by averaging spectra across the region's voxels. For each *region of interest*, spectral segments were clustered twice (Fig. 3.1c, d). The first clustering was done separately for each participant. The second time, clustering collectively for all participants resulted in spectral profiles representing brain dynamics states generalizable across participants (Fig. 3.1e; Fig. 2.7). These sets of profiles were called *spectral fingerprints*. Similarity across brain areas was analyzed using agglomerative hierarchical clustering (Algorithm 14). As a result, a binary tree was formed (Fig. 3.5). Identification accuracy, which quantified regional specificity, was calculated by dividing individual cluster data into several training-test

subsets (bootstrap aggregating, [190]; Algorithm 8). The fit between test subsets (individual-level clusters) and training subsets (group-level clusters) was quantified using negative log-likelihood estimation.

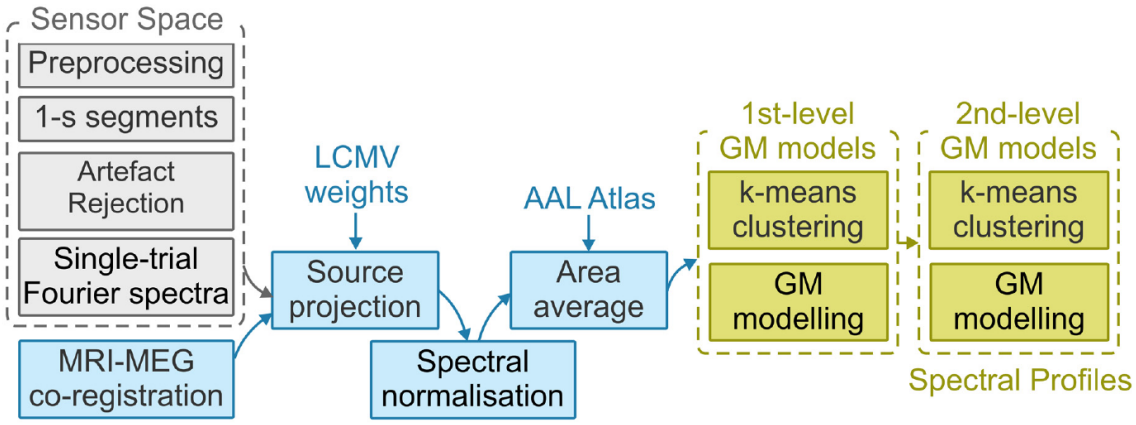


Fig. 3.2. Spectral Fingerprinting analysis pipeline. MEG signal underwent typical preprocessing steps, including artifact rejection. Then was segmented into 1-second segments (commonly called *trials*), which were transformed into power spectra. Source projection using the LCMV approach [142] was possible due to the MRI-MEG co-registration procedure, i.e., matching MRI and MEG coordinate system. Source-projected spectra were normalized and averaged across locations inside each brain region. Areas were defined in accordance with Automated Anatomical Labeling (AAL) atlas [189]. Clustering was performed to reduce dimensionality and group brain activity into common spectral patterns. GM modeling refers to the computation of the Gaussian Mixture Models initialized with the k-means centroids. Individual-level clustering resulted in 1st-level GM models. Subsequent clustering on group-level produced so-called 2nd-level GM models, where produced clusters are called *spectral profiles* or *spectral modes*. Source: [9]

3.3 Results

The human brain is engaged in different modes of activity. Computed group-level clusters (spectral fingerprints) consist of more than one mode with at least one characteristic peak. It is the case for all but one of brain areas (right middle frontal gyrus was characterized with a single mode).

Spectral Fingerprinting allows distinguishing between different modes of activity occupying the same frequency band. The method clusters the activity inside a given frequency band instead of averaging it. For example, in the visual cortex, two different alpha clusters were found (see Fig. 2.7 top row for middle occipital gyrus). The existence of different generators of alpha rhythm was confirmed and discussed in [79]. Further studies are needed to link state changes of the brain to switching between the spectral modes.

The specificity of spectral power peaks allows for area recognition. Fig. 3.3 reports the results of the regional identification procedure. For each brain area, identification

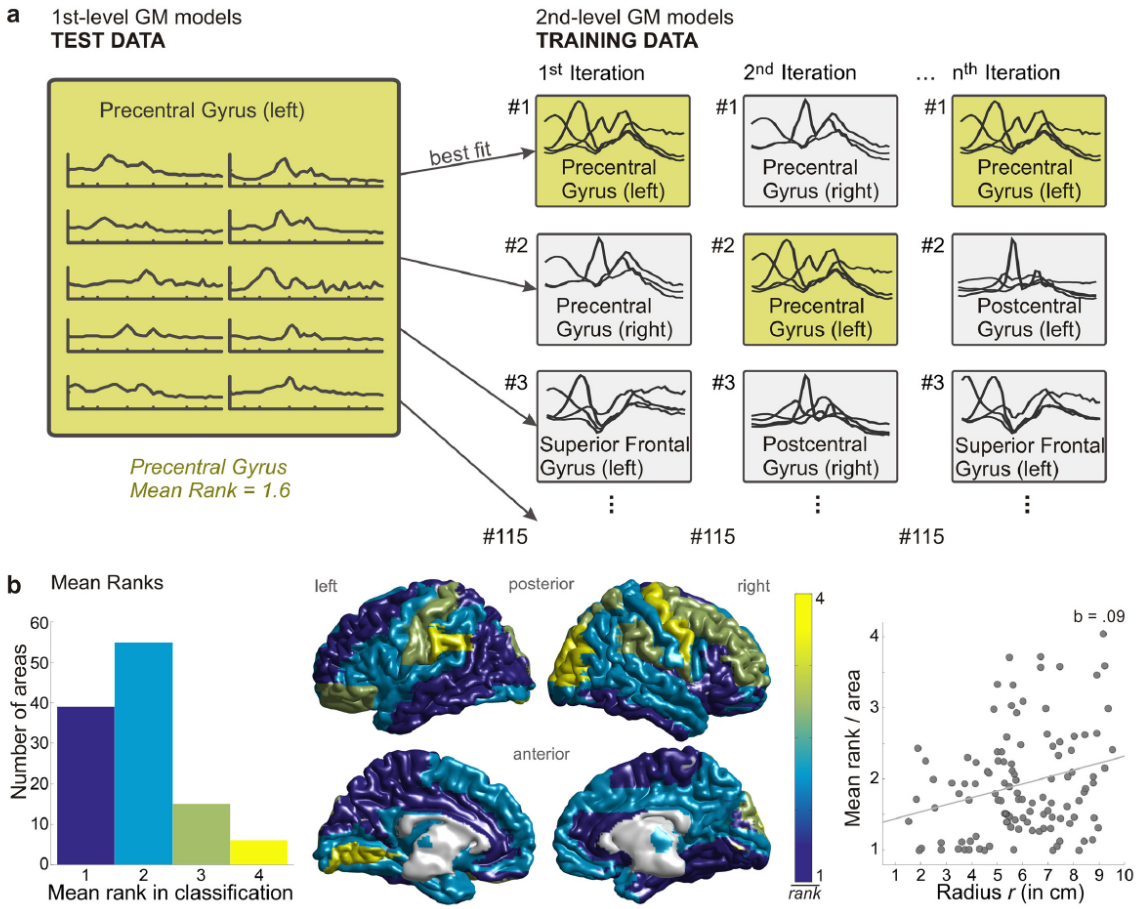


Fig. 3.3. Spectral Fingerprinting identification procedure and results. **a)** Illustration of classification procedure (Algorithm 8 in Appendix A). Spectral fingerprints (group-level) were created from half of the sample acting as a training subset. From the remaining sample, area-specific subject-level data were then tested against each trained fingerprint (all possible regional pairs were tested). The fit of each test set was determined through negative log-likelihood and then ranked. The procedure was repeated several times with different training-test data subset assignment. Ranks were averaged across iterations. **b)** Distribution of regional mean ranks in identification. (left) Histogram of mean ranks across 115 brain areas. Bin width is one. (middle) Topography of mean ranks (color-coded from 1, blue, to 4, yellow; bin width is one). (right) Linear regression suggests the dependency of mean rank per region and the distance from the center of the brain. The mean rank increases the further an area is away from the center of the brain. Regions closer to the center of the brain are easier to classify than more superficial areas. Source: [9].

accuracy was quantified in terms of probability (negative log-likelihood, NLL). NLL values were further ranked. The authors defined the *mean rank* measure (Algorithm 9). It is a non-negative real value, constrained between 1 and the number of regions considered. The number of regions is denoted as A (here $A = 115$). It tells how hard it is to identify a particular area. A mean rank of 1 indicates that the correct region model was the most likely to fit the test area; a rank of 2 indicates that the correct region model was the second most likely, and so on. The expected mean rank for identification purely by chance equals $A/2 + 0.5$. Keitel and Gross achieved a mean rank average across areas equal to 1.8. It means that many regions were correctly matched, and some were confused with only one different area (see the histogram in Fig. 3.3b). It often was the homologous region (i.e., region located symmetrically on the other hemisphere). Mean rank with accepting homologue areas as correct match improved to 1.4.

The authors confirmed that clustering is necessary for robust identification. It was done by setting the number of clusters to $k = 1$, thus essentially simulating a single power spectrum per area. The identification yielded a mean rank of 2.3 (worse than 1.8 for clustered data). In conclusion, a single spectral profile can still be quite efficient as a regional signature; however, characterizing brain areas as a set of spectral modes yielded better results. This indicates that the resting-state oscillatory activity of individual brain areas can act as a regional fingerprint that permits sufficient high-accuracy identification.

Spectral fingerprints are modulated in response to changing stimuli. The authors compared participants' data collected in resting and auditory stimulation conditions to test this claim. The auditory experiment differed in only one aspect, compared to the pure resting-state condition — participants listened passively to the 7-min recording of a real-life story ("Pie-man," told by Jim O'Grady at "the Moth" storytelling event, New York). Analysis showed that auditory stimulation distinctly alters spectral modes in primary auditory cortices in line with the existing knowledge about speech perception mechanisms (Fig. 3.4).

Peak frequencies in regional fingerprints agree with the intrinsic frequency preferences of neurons supported theoretically [191] and evidenced in other studies [73][41]. For example, the power of the dominant spectral mode (i.e., spectral profile with the largest duration) in the middle occipital gyrus peaks at 10.5 Hz, which agrees with the natural frequency found in Brodmann Area 19 (11 Hz). A spectral peak at 18.5 Hz in the superior parietal gyrus corresponds to the natural frequency found in Brodmann Area 7 (20 Hz). Power peak in paracentral lobule at 27.5 Hz relates to the natural frequency found in Brodmann Area 6 (31 Hz).

The results also suggest that it is easier to identify areas deeper under the cortex.

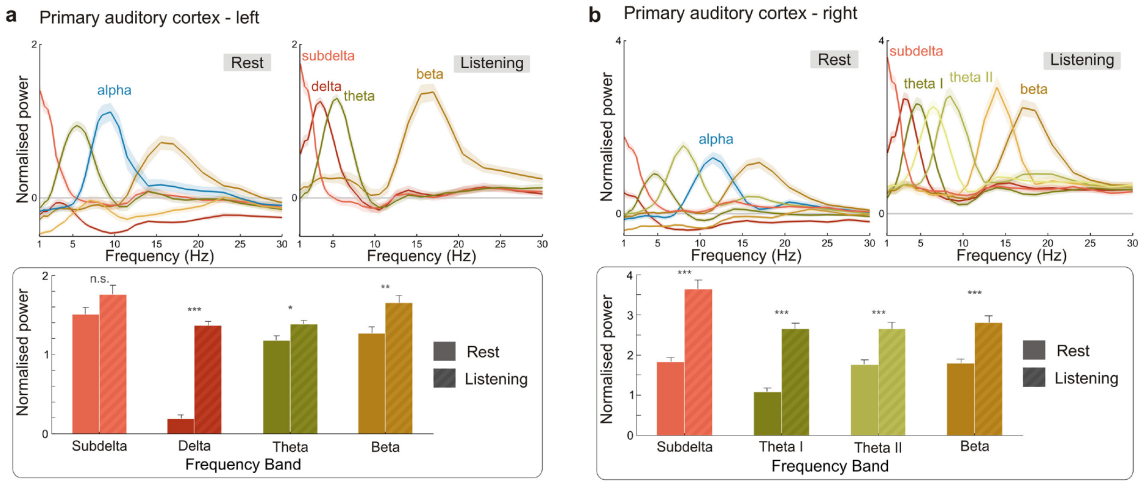


Fig. 3.4. Spectral fingerprint modulation in response to auditory stimulation. Upper panels show group-level spectral modes for primary auditory cortices. The amplitudes of spectra with comparable peak frequencies in both conditions were subjected to an independent sample t-test across participants (lower panels). Bars illustrate amplitude means across individual spectra. Error bars illustrate standard error of the mean across participants. Asterisks denote significance between rest and listening: * $p < .05$, ** $p < .01$, *** $p < .001$, n.s. not significant. Source: [9].

A weak linear relationship between identification accuracy and distance from the brain center was also found ($R^2 = 0.07, F(1, 113) = 8.15, p_{\text{corrected}} = 0.04$, p-value corrected for multiple testing using sequential Bonferroni procedure; see the scatter plot in Fig. 3.3b), It seems counterintuitive, as beamforming in general offers better estimation of source activity the closer estimated location is to the sensor array [43]. Our unpublished exploratory data analyses did not confirm the direction of the linear relationship stated by Keitel and Gross. Unfortunately, no further exploration was conducted, and none of the works devoted to Spectral Fingerprinting comments on this observation. I refrain for now from making a conclusive statement about it. Further studies devoted to Spectral Fingerprints will hopefully clarify this matter.

Areas with similar spectral profiles resemble large networks. Hierarchical agglomerative clustering linked subsets of bilateral frontal, central, and two sets of occipital areas. Distant homologue areas in the left and right hemispheres were connected due to the similarity of their spectral profiles, not due to their spatial proximity. Formed unions corresponded well to large-scale anatomical parcellation of the human cortex (Fig. 3.5). This is striking because the clustering algorithm has no information about the spatial location of individual areas, it is purely functional-based. Yet, linked clusters resemble sensorimotor, frontal, and visual networks [159].

Similar results were reported on other datasets by different authors. They used the same [11][12] or similar fingerprinting methodology [91].

To summarize, findings from the report of Keitel and Gross are in line with previous

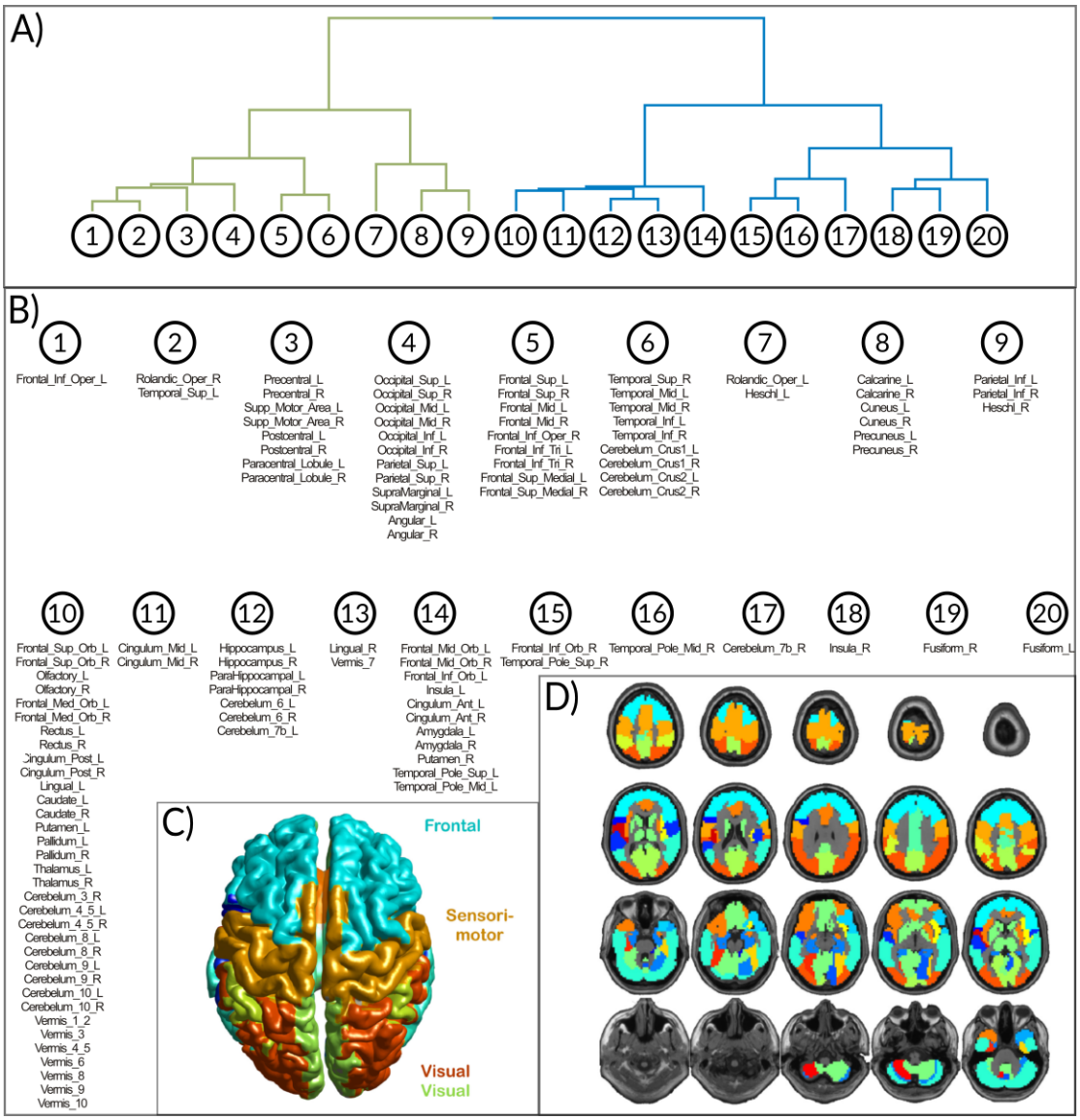


Fig. 3.5. Area networks according to spectral fingerprints similarity analysis (Algorithm 14 in Appendix A). **A)** Binary hierarchical cluster tree resulting from the similarity analysis. The dendrogram was cut at a maximum of 20 clusters. Truncated tree leaves are labeled with names (presented in panel **B)**) of the corresponding 115 areas from the AAL brain atlas[189]. **C)** Topography of similar areas. **D)** Axial slices with color-coded similarity. Adapted from: [9].

physiological research on natural frequencies in the brain. They showed that it is possible to study brain areas specialization by constructing and testing local activity modeled as spectral fingerprints. Oscillatory patterns turned out to generalize across participants. Spectral peak frequencies are shifted in response to the changing stimuli (rest vs. auditory stimulus). Moreover, local specialization was confirmed and seemed to be the substrate of global large-scale networks that underlie cognitive processes.

3.4 Replication study by Lubinus et al., 2021

In 2021, Christina Lubinus and collaborators [11] replicated the results of Keitel and Gross and discovered that spectral fingerprints could be selectively altered by congenital blindness. They took the data previously collected in a larger project [192][193]. Participants were divided into three groups (congenitally blind, CB, $N = 26$; sighted blindfolded, S-BF, $N = 24$; sighted with opened eyes, S-EO, $N = 23$). Groups were matched in age, gender and education. There were $N = 73$ subjects in total. Approximately four minutes of cleaned resting-state MEG signal was divided into 0.8-second segments (mean = 340.3 trials, sd = 34.7). Data underwent source reconstruction, frequency analysis, and clustering according to the Spectral Fingerprinting pipeline [9]. Anatomical brain parcellation (AAL, [189]) defined regions of interest. Area specificity was tested, following the same classification schema as in [9]. Participants within each of the three groups were randomly split into test and training groups. Individual fingerprints from the test subsample were compared with the trained models using negative log-likelihood. The automatic cross-group classification was performed to check which regions are mostly affected by congenital blindness. Training data was drawn from the S-BF group, and test data was drawn from the CB group.

Area identification performed on prepared data yielded an average mean rank equal to 2.7. It means that most regions were confused with only one or two other ROI on average (Fig. 3.6). Again, classification ranks referred to the probability of a region being identified: a mean rank of 1 that a region was correctly assigned on every classification iteration; a mean rank of 2 means that a correct region had the second highest probability among all areas, across iterations. Identification purely by chance would yield a mean rank $115/2 + 0.5 = 58$.

To test which areas differed in their spectral properties between S-BF and CB groups, authors performed a cross-group classification (Fig. 3.7). Models from blindfolded people were to "recognize" test regions data from congenitally blind people. This yielded significantly worse identification (mean rank of 5.3 vs. within-group mean ranks of 2.6 for S-BF and 2.5 for CB subjects). Oscillatory patterns for congenitally blind are

heavily altered to deteriorate regional identification using healthy people data.

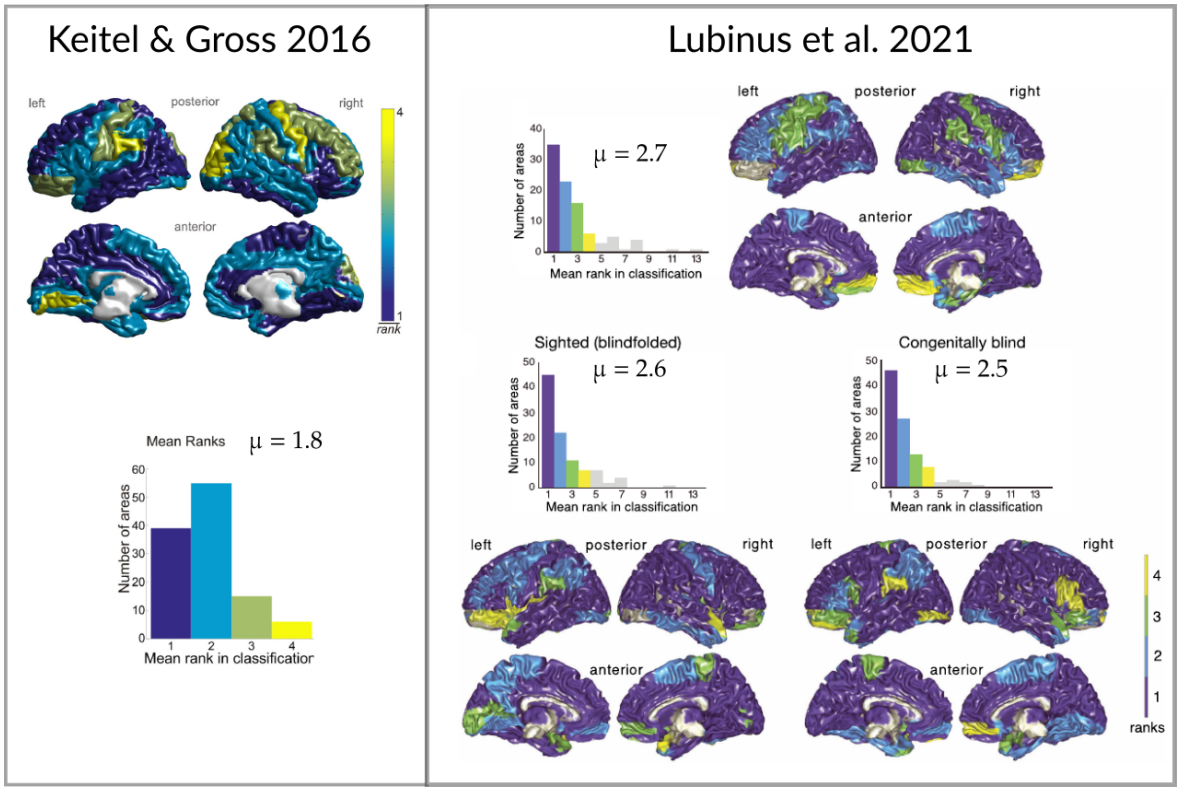


Fig. 3.6. Comparison between two different studies performing identification of brain areas from their spectral fingerprints. Bin width is 1 for all presented histograms. The averages of mean rank across all areas are denoted by μ. Adapted from: [9] and [11].

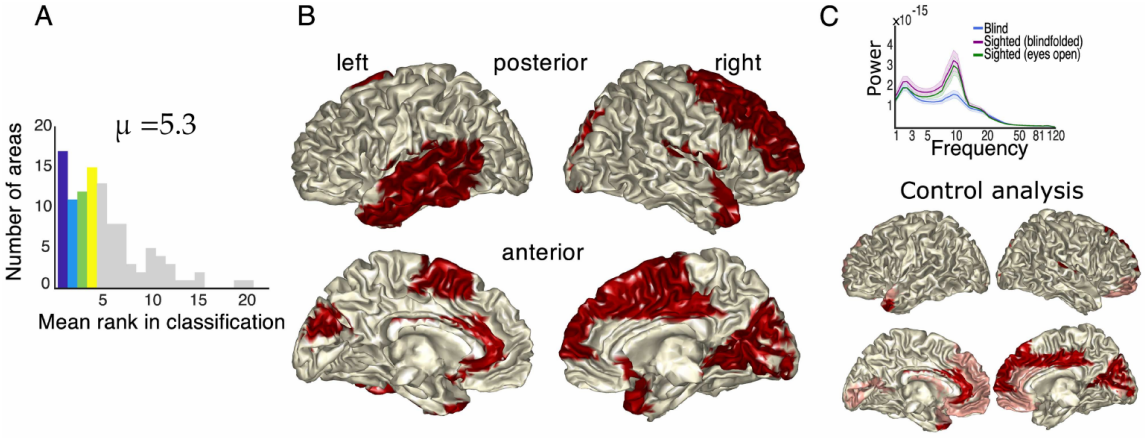


Fig. 3.7. Cross-group comparison by Lubinus et al., 2021 [11]. Training data was drawn from the blindfolded people and test data from the congenitally blind people. **(A)** Histogram of identification ranks (bin width is 1). The average mean rank across all areas is denoted by μ. **(B)** Regions with significantly worse rank value in comparison to the within-group identification. **(C)** Comparison between spectral fingerprints in different experimental groups for an arbitrarily chosen region. Shade error bars depict the standard error of the mean across participants. The lower panel shows the result of the control analysis performed on non-normalized data. Analysis revealed significant findings in various areas overlapping with the areas observed in the main analysis (highlighted in dark red). Adapted from: [11].

In 2018, before the replication study by Lubinus, I announced initial results regarding the reproduction of fingerprints on the original MEG data and replication on two independent EEG datasets [12]. Discussion of this publication is shown in the next section.

3.5 Reproduction and replication study by Komorowski et al., 2018

Four years before the work of Lubinus et al., was published — in 2017, the team I was part of became interested in Spectral Fingerprinting and wanted to check how well results from Keitel and Gross could be reproduced. The authors provided the beta version code that allowed other researchers to repeat some part of their analysis. We decided to reconstruct the code covering the missing parts. This reconstructed code became later the basis for the author's toolbox implementation [10] that we discuss in the next chapter.

In 2018, we published the replication paper [12] with the hope of repeating the findings of [9]. Additionally, our goal was to check (for the first time ever) brain regions' identification accuracy level for high-density EEG recordings instead of MEG. This endeavor was of great theoretical and practical interest because obtaining MEG spectral

fingerprints requires costly equipment compared to any electroencephalographic setup.

I have computed spectral fingerprints for three independent datasets and compared them with results obtained by Keitel and Gross on their original dataset. The original MEG dataset (MEG-O) consisted of data from 22 subjects recorded in resting-state condition, with the analyzed signal length of 466 ± 20 s. A subset that was open to the public, labeled MEG-R, was used in our attempt to reproduce the results from the original article. Additionally, with our reconstructed code, two separate EEG datasets recorded in the resting-state conditions were analyzed: first EEG dataset (EEG-A) was downloaded from the Age-ility Project open repository [194]. Subjects were asked to sit and rest with their eyes closed. The length of the signal was 118 ± 7 s for each of the 22 subjects participating in this study; second EEG dataset (EEG-B) was collected in our laboratory and included N = 12 subjects recorded in the same experimental conditions as the MEG dataset from Glasgow (resting state, eyes open). The analyzed signal length was 460 ± 32 s.

I have visually examined spectral profiles in particular ROIs (one-by-one) and found that both MEG-O and MEG-R were to a great extent coherent. However, this was not the case for EEG-based fingerprints (Fig. 3.8). The area identification procedure revealed a massive difference of regional specificity (measured with mean rank) between MEG-based and EEG-based fingerprints (Fig. 3.9). On average, the EEG-based fingerprint of the correct area was the tenth most likely to fit the test area. Probably, this was caused by the fact that the EEG technique does not resolve brain dynamics in higher frequencies as well as MEG [195], and thus information needed to better distinguish spectral profiles was lost.

In conclusion, this chapter introduced the seminal approach to Spectral Fingerprinting [9], which is the central method of my doctoral research. It allows to capture individual characteristics of brain areas facilitating their identification. I have also overviewed the related work of Lubinus et al., 2021 [11] that showcases the usability of the method in detecting spectral alteration of the human brain regional activity due to congenital blindness. The degree of attempted reproduction of MEG fingerprints and their specificity was similar to one presented by Lubinus and collaborators. Causes of the difference between original results [9] and their reproduction remain unknown. However, I hope that published open-source implementation of the promising Spectral Fingerprinting approach in the form of the ToFFi Toolbox [10] would help solve this mystery soon.

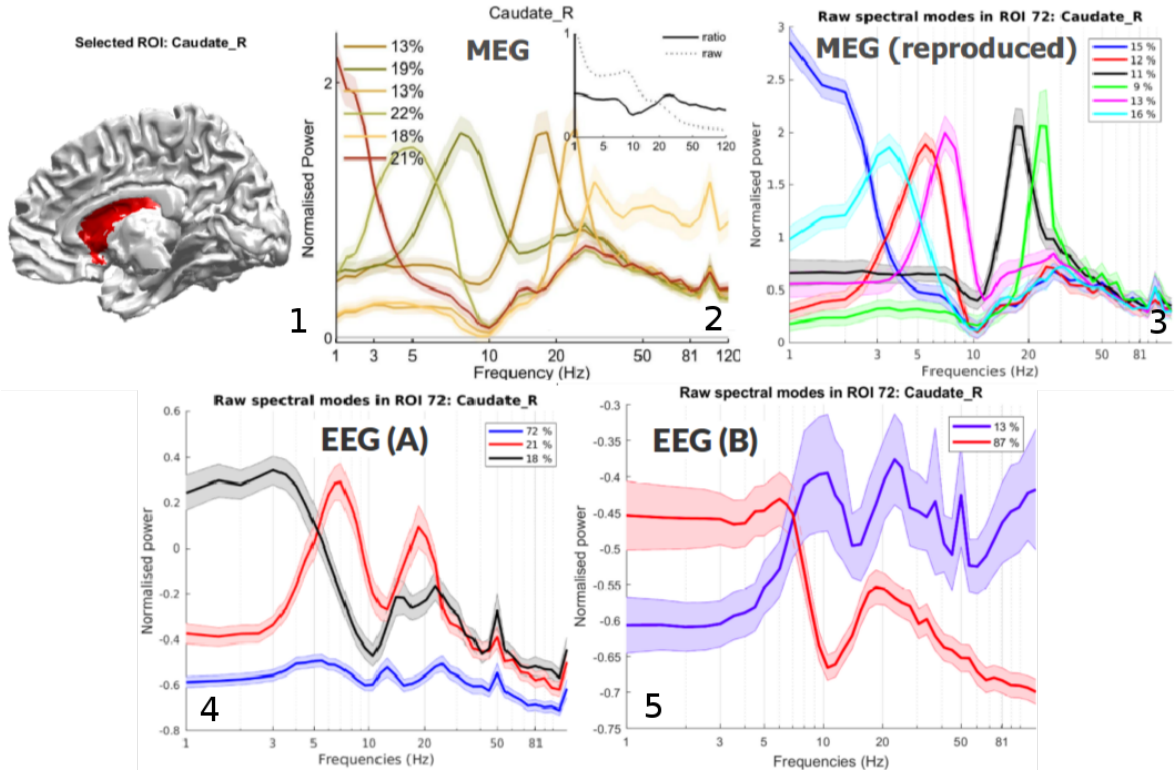


Fig. 3.8. Spectral fingerprint of the caudate nucleus. Pictures from left: **(1)** Caudate nucleus (right part) highlighted according to its coordinates from the AAL [189] brain atlas. **(2)** Clustered power spectra of the caudate nucleus (normalized power, i.e., spectral power in comparison to the whole brain). Legends show the corresponding duration of each pattern (i.e., the percentage of trials in which each spectrum was present on average during recording). Shaded error bars illustrate the standard error of the mean across participants. Inlets show average power spectra for respective areas without normalization (dotted lines) and with ratio normalization (continuous lines) [9]. Analogous spectral fingerprint was obtained from reproduction **(3)**, using EEG data from the Age-ility project **(4)**, and using data recorded in our laboratory **(5)**. Source: [12].

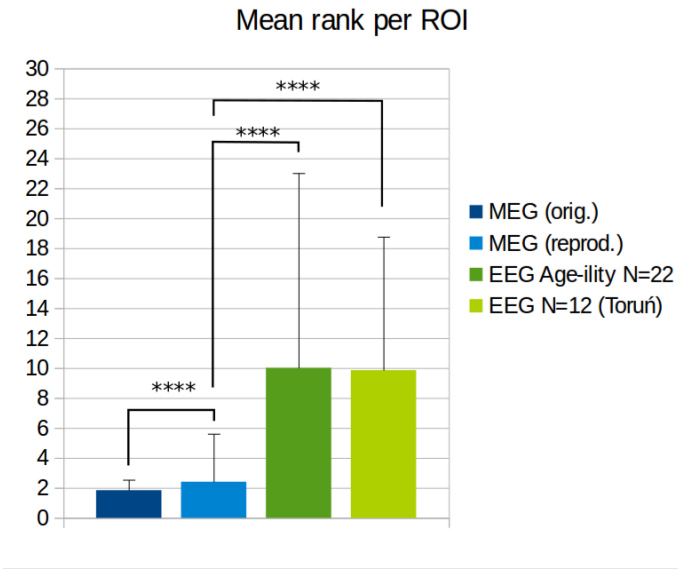


Fig. 3.9. Comparison of spectral fingerprints identification accuracy. There is a significant difference between EEG datasets and MEG datasets in terms of mean rank value: (Mann-Whitney: U=8194, $p<10^{-4}$) for MEG-R vs. EEG-A; (Mann-Whitney: U=7844.5, $p<10^{-4}$) for MEG-R vs. EEG-B. This means that building SFs using MEG entails more specific ROIs activity representations and thus easier classification of ROIs activity. The difference between EEG datasets in terms of the mean rank is insignificant (Mann-Whitney: U=13294, $p = 0.67$). There was a significant difference between MEG datasets (Wilcoxon signed rank test: U=1667.5, $p<10^{-4}$). One-sided standard deviation bars were added. Negative parts of the error bars were neglected for clarity. Comparisons between EEG datasets with MEG-O dataset were omitted for clarity. Source: [12].

4 Implementation of the ToFFi Toolbox

4.1 Motivation

We are now ready to follow the introduction of our novel implementation of the Spectral Fingerprinting method [9] as a highly-configurable MATLAB toolbox [10]. In Chapter 3, we learned how Spectral Fingerprinting emerged and what has been done on the topic so far. The labor toward understanding patterns of human brain oscillations (Chapters 1 and 2) has finally converged into this very practical part of the thesis.

Let us start with the vital question: why would anybody need another toolbox? There are indeed many open software packages available to analyze neural data. The Fieldtrip Toolbox[5] [196] was designed to analyze both sensor and source level EEG/MEG/iEEG/NIRS data. EEGLAB[6] [197] helps process continuous and event-related electrophysiological data by implementing many analytic methods (ICA, time/frequency analysis, artifact rejection, event-related statistics, microstates analysis) and several useful routines for visualization. To simulate brain dynamics, perform connectivity analyses, and solve forward/inverse problems, the supFunSim[7] toolbox [146] and the Virtual Brain[8] system [198] are among suitable choices.

However, there was no open software for analyzing spectral fingerprints. One can debate that the original authors of the method provided the beta version script for computing fingerprints (Appendix B), however, it documents only part of the pipeline presented in Fig. 3.2 and cannot be directly used to analyze other than original data. Corresponding with one of the authors (dr Anne Keitel) and obtained materials helped me recreate the missing code. Moreover, during my research, I needed a versatile tool to explore fingerprints in different settings effectively. Delivering the **Toolbox for Frequency-based Fingerprinting** (ToFFi) was an attempt to fill those gaps.

In this chapter, modules of the ToFFi Toolbox and their implementation will be presented, concluded with a brief description of the typical workflow. Some thought will be given to the time complexity of the presented algorithms. Implementation of the toolbox supports both concurrent computations and reproducibility of the results. These two issues will also be discussed. To illustrate toolbox capabilities, this chapter will end with a demonstration. The next chapter (Chapter 5) is an example of how ToFFi can be applied in the context of real-life computational neuroscience study.

All of the material included in this chapter is the original contribution of the thesis

[5] `https://www.fieldtriptoolbox.org/` (accessed: June 4, 2022)
[6] `https://eeglab.org/` (accessed: June 4, 2022)
[7] `https://github.com/nikadon/supFunSim` (accessed: June 4, 2022)
[8] `https://www.thevirtualbrain.org/` (accessed: June 4, 2022)

author in the field of neuroinformatics, except the references or material explicitly stated to be crafted by somebody else. It advances software development in the field and will hopefully lead to some exciting discoveries.

The open-source code of the presented software can be accessed and downloaded here: `https://github.com/micholeodon/ToFFi_Toolbox`. Consult Appendix C for further directions. Mathematical equations and algorithm pseudocodes can be found in Appendix A.

4.2 Capabilities and upgrades

ToFFi Toolbox upgrades the original Spectral Fingerprinting pipeline [9] by being capable of:

1. Analyzing multichannel brain activity recordings such as EEG or MEG,
2. Modeling human brain activity as individual-level spectral fingerprints,
3. Modeling human brain activity as group-level spectral fingerprints,
4. Analyzing spectral fingerprints similarity,
5. Estimating fingerprint-based area identification accuracy,
6. Performing parallel computations,
7. Adjusting many processing parameters,
8. Controlling results reproducibility,
9. Visualizing the results.

There are two main paradigms of electrophysiological recordings: continuous and event-related. They include both resting-state brain activity as well as task-related fluctuations from experiments that approximate real-world situations. Originally, Spectral Fingerprinting was used to analyze spontaneous brain oscillations, however, technically there are no known limitations to using the ToFFi Toolbox to analyze the contrast between rest and task activity or model fingerprints of repeated responses taking advantage of the ERP paradigm (see paragraph on evoked potentials in Section 1.5).

In terms of computing fingerprints (both on individual and group level), the original procedure proposed in [9] was recreated, and new features regarding the clustering were added (e.g., optimality criterion for the number of spectral modes during construction of individual fingerprints).

In the beta-script provided by the authors of the method (Appendix B), there was no code regarding fingerprint similarity analysis and area identification accuracy estimation. During the initial reproduction and replication of results (reported in [12]),

I have followed the original identification procedure based on bootstrap aggregating (so-called *bagging*). Later, I substituted it for a more straightforward method based on cross-validation. In comparison with cross-validation, bagging reduces the estimates' variance at the cost of increased bias [199]. Bagging was proposed in [9] as a beneficial strategy for alleviating the problem of the low number of subjects ($N = 22$) which could lead to a high variance of resulting estimates. However, with time, the problem of too small sample size started to disappear. First, during my research, I relied mostly on the Human Connectome Project MEG Dataset, which contains $N = 89$ subjects [200]. Secondly, the number of publicly open databases was increasing, often containing data of hundreds of subjects. Thirdly, cross-validation was easier to control, explain, and interpret, which is important, also for scientific communication. All these arguments concluded eventually in bagging being replaced with cross-validation. This step has also inspired me to implement a new toolbox component — the analysis of the individual-level fingerprints, based on cross-validation as well (Algorithm 12).

Another upgrade concerned the problem of long running time of Spectral Fingerprinting algorithms. To solve it, I have adapted ToFFi Toolbox to benefit from concurrent computing methods. Most computationally-intensive parts of the Spectral Fingerprinting pipeline are now possible to be run in parallel on many CPU cores or even on more than one machine simultaneously.

In the beta-script, adjusting the parameters was risky, as the code was not prepared for that. To change some of the parameters, there was a need for a deep understanding of the code and some skill in programming, which unfortunately limited the pool of potential users. Currently, thanks to the reimplementation, the user of the toolbox gained control over many parameters of the processing via the centralized configuration interface represented by a single text file. This is a far more accessible approach.

Features that were previously absent include: the possibility of usage of different input datasets, deciding on the sequence order of computation stages, parallel computations, and using different brain atlases. This last point is also the fulfillment of one of the future work suggestions stated by Keitel and Gross in their article [9].

As some algorithms of the Spectral Fingerprinting approach demand simulating randomness (e.g., clustering routines), the challenge of maintaining result reproducible emerged. Significant effort was put into granting the user convenient control over MATLAB pseudo-random number generators. With extensive documentation and open source code, any interested investigator can repeat analyses and recreate results obtained by other toolbox users.

Finally, one can produce plots and graphs immediately after computations are done, thanks to the implemented visualization routines, making comparisons between the

data obtained by different researchers easier and more standardized.

4.3 External software

Most of the procedures included in the ToFFi Toolbox are coded in MATLAB. The following toolboxes were used: Signal Processing Toolbox, Statistics and Machine Learning Toolbox, MATLAB Parallel Computing Toolbox, and Fieldtrip Toolbox. Managing the data and automation of repeating routines were achieved using Bash scripting language. Interaction with a supercomputer cluster was facilitated using SLURM workload manager. Below, one can find a brief description of mentioned software.

MATLAB is the name of both programming language and commercial computer application, used to perform scientific and engineering numerical calculations with the capability to present results as graphs or animations. The name is the abbreviation of "MATrix LABoratory", as it was originally designed to perform calculations on matrices. It was designed in the late 1970s by Cleve Moler [201]. As of 2022, MATLAB is a mature, actively developed language used by more than 5 millions users worldwide[9].

Its typical application areas are: linear and nonlinear algebra, machine learning, signal processing, image processing, economics and finance, control theory, robotics, and biology. Extending MATLAB functionality is possible by the growing collection of so-called *toolboxes* — sets of libraries for solving specialized problems in specific areas (e.g., signal analysis, electronics, molecular dynamics). MATLAB is:

- a multi-paradigm language (imperative, procedural, functional; supports object-oriented programming),
- an interpreted language,
- a weakly-typed language (with type inference),
- an array programming language (similar to Julia or R).

MATLAB is popular in neuroscience. There are several books devoted to use of MATLAB in brain research [202][195][203][204], and plenty of open-source and commercial toolboxes available to aid researchers in the field (for examples refer to: [197][196][205][206][207][146][10]). With its high-level easy-to-read syntax, scientists need not be concerned too much about managing memory, variable initialization, pointers, and other details relevant in lower-level programming languages like C, which renders MATLAB attractive for delivering solution prototypes and data analysis pipelines. For

[9] `https://www.mathworks.com/content/dam/mathworks/handout/2022-company-factsheet-8-5x11-8282v22.pdf` (accessed: June 4, 2022)

time-optimized solutions, MATLAB code can be converted into C code and compiled into a standalone executable using MATLAB Coder©.
ToFFi Toolbox takes advantage of built-in functions from the following toolboxes:

- MATLAB Signal Processing Toolbox,
- MATLAB Statistics and Machine Learning Toolbox,
- MATLAB Parallel Computing Toolbox,
- Fieldtrip Toolbox.

Recommended software versions to use ToFFi Toolbox:

- MATLAB - R2021a or newer,
- Fieldtrip - revision 20210816 or newer.

Additionally, `vline.m` and `hline.m` functions[10] by Brandon Kuczenski are used for plotting, and `HZmvntest.m` function[11] by Antonio Trujillo-Ortiz for multivariate normality testing.

In subsequent sections, one can find how brain activity data was stored and organized in computed memory. For easier reading, it is appropriate to briefly mention the types of MATLAB variables. By default, MATLAB stores all numeric variables as 16-digit double-precision floating-point values complying with IEEE Standard 754 [208][209]. Data precision can be changed using implemented variable-point arithmetic. Additional data types store text, integer or single-precision values, or a combination of related data in a single variable. Most of the variables were multidimensional arrays, structure arrays, or cell arrays.

Multidimensional arrays have at least one dimension along which each container stores data elements of the same type. Here are examples of multidimensional arrays — a number:

```
>> 7    % a number (i.e., 1x1 array)
ans =
7
```

a vector:

```
>> [1, -2, 6.7, 0]    % a vector (1x4 array)
ans =
1.0000   -2.0000    6.7000         0
```

a 2D-matrix:

[10] `https://www.mathworks.com/matlabcentral/fileexchange/1039-hline-and-vline` (accessed: June 4, 2022)

[11] `https://www.mathworks.com/matlabcentral/fileexchange/17931-hzmvntest` (accessed: June 4, 2022)

```
>> [2, 3, 1; -7 3 1]    % a 2D-matrix (2x3 array)
ans =
2     3     1
-7     3     1
```

a 3D-matrix (2D-matrix need to be constructed first, and then extended):

```
>> M = [1 3 4; 2 1 1]   % 2D-matrix ...
M =
1     3     4
2     1     1
>> M(:,:,2) = [7 7 7; 1 1 1] % extended into 3D-matrix (2x3x2 array)
M(:,:,1) =
1     3     4
2     1     1
M(:,:,2) =
7     7     7
1     1     1
```

A *structure array* is a data type that groups related data using containers called *fields*. Each field can contain any type of data. Data in a structure can be accessed using dot notation of the form `structName.fieldName`. Here is an example of an array containing two structures:

```
>> brainArea(1)
ans =
struct with fields:
name: 'Precentral Gyrus'
ID: 1
location: [-2 11.2000 0]
>> brainArea(2)
ans =
struct with fields:
name: 'Cerebellum'
ID: 2
location: [3 0.1000 -12]
```

A *cell array* is a data type with indexed data containers called cells. Each cell can contain any type of data. Here is an example of a 2x3 cell array:

```
>> {'a', [1 0; 0 1], -9.1; 2, 'hello', brainArea}
ans =
2×3 cell array
{'a'}    {2×2 double}    {[ -9.1000]}
{[2]}    {'hello'   }    {1×2 struct}
```

Fieldtrip toolbox was used in several parts of the code following the original implementation of Spectral Fingerprinting by Keitel and Gross (Appendix B). It is

an open-source software package that helps in analysis of MEG, EEG, MRI, fMRI, and other electrophysiological data. It contains a set of consistent and user-friendly high-level functions that allow neuroscientists to analyze experimental data. It offers time and frequency analysis algorithms, source reconstruction solutions, connectivity analysis, and statistical procedures [196]. Fieldtrip's capabilities are constantly being extended by the community.

In ToFFi, tasks aided by the Fieldtrip toolbox include: signal preprocessing, spectral analysis, and brain atlases preparation.

Bash ("***B****ourne* ***A****gain* ***Sh****ell*") is one of several Unix shells available and a command language written by Brian Fox in 1988 [210] that typically runs in a text window (a user interface for an underlying command-line interpreter) where the user types commands that are translated into operating system instructions to cause desired actions. These actions regard searching for and manipulating files[12] — their content and attributes.

In ToFFi, Bash scripts were given to the user to automate some of the maintenance tasks encountered frequently: configuration, data backing up, restoration, and removal.

SLURM[13] is an open-source, scalable cluster management program and job scheduling system for large and small Linux clusters. It takes responsibility for allocating computer nodes to users for a specified time window in order to use nodes' computational resources. It provides an interface for starting, executing, and monitoring parallel jobs. Finally, it manages a queue of pending work to optimize the usage of resources.

In ToFFi, SLURM commands were included in coded scripts responsible for launching separate parts of Spectral Fingerprinting concurrently on a supercomputer (parallel computations were carried out at the Centre of Informatics Tricity Academic Supercomputer & Network, CI-TASK; Gdańsk, Poland).

4.4 ToFFi Toolbox Architecture

I tried to write readable and well-documented code, complying with good practices[14].

As it was impossible to include all the toolbox code explicitly in this thesis, a link to the code repository and its documentation can be found in Appendix C.

When writing code, one of many good practices is *modularity* — the division of a computer program into small, possibly independent, self-contained pieces with precisely

[12] From Unix's perspective, every data stream is technically a "file": documents, directories, network connections, processes are *files*.

[13] `https://slurm.schedmd.com/` (accessed: June 4, 2022)

[14] Richard Johnson (2022). MATLAB Style Guidelines 2.0 (`https://www.mathworks.com/matlabcentral/fileexchange/46056-matlab-style-guidelines-2-0`), MATLAB Central File Exchange. Retrieved March 30, 2022.

defined responsibilities. ToFFi is a modular piece of software. We can think about two versions of it: its logical architecture and its practical architecture (Fig. 4.1).

First, it was noted that Spectral Fingerprinting pipeline is a chain of processing steps that perform a transformation of some input data set - the program change input data representation passing data from the time domain into the frequency domain. It naturally follows that there needs to be some preparation stage before the transformation is possible. This concluded with knowing that the first part will be a component responsible for this data preparation. Then the second component could be the transformation part itself. Finally, knowing that Keitel and Gross performed two tests on constructed representations (i.e., test for similarity and test for identification), the third and last theoretical component was needed. Taking all of these observations into account, I defined three major components in the toolbox's logical architecture: *Data Preparation*, *Spectral Fingerprinting*, and *Analysis* (Fig. 4.1A).

Secondly, in practice, such a system needed three additional parts. Considering the complexity of the toolbox, it would be convenient to have a central piece for its configuration (first additional part) traversing many files to configure. A researcher should have the possibility to visualize spectral fingerprints and their measures (second additional part). There is also data managing involved along the way: moving, deleting, and backing up (third additional part). In summary, *Configuration*, *Presentation*, and *Maintenance* parts needed to be implemented as well (Fig. 4.1B). These parts make it easier to use the toolbox because a user can think of a toolbox as a set of five things (instead of hundreds of individual scripts). Automation of repeating routines encourages frequent use and also prevents errors that otherwise could happen when doing those tasks manually. In total, there are five main components of the toolbox:

I. Data Preparation,
II. Spectral Fingerprinting,
III. Analysis,
IV. Presentation,
V. Maintenance.

Presented division embodies the "divide and conquer" principle [211], which strength lies in dividing a problem into smaller manageable sub-problems, solving them independently, and compounding results together into a final solution. Similarly, smaller pieces of the software were coded separately and then connected together to form a large system.

Implementation heavily impacted the repository directory structure shown below. Comments on the right help relate directories to Figure 4.1B.

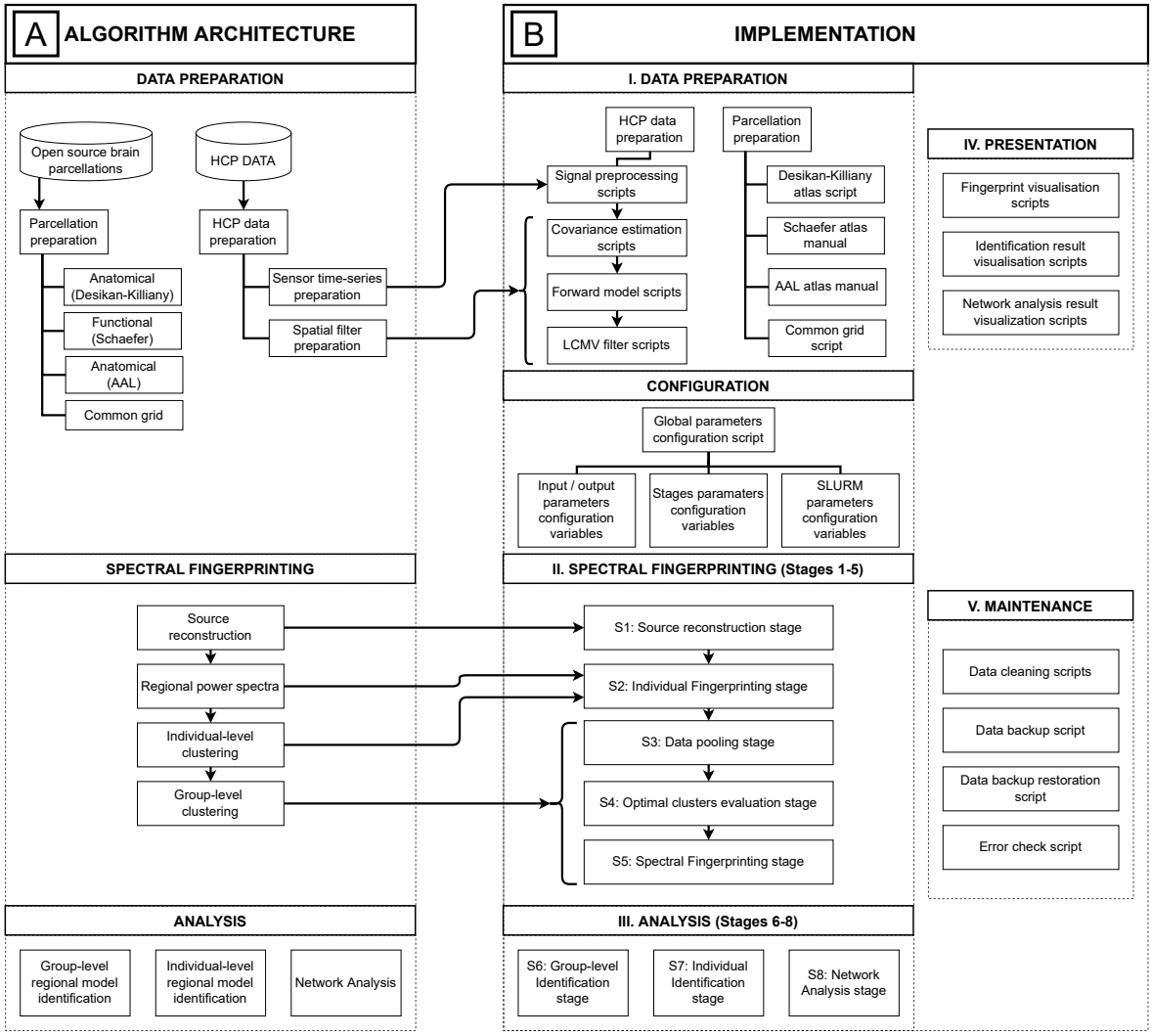


Fig. 4.1. ToFFi toolbox architecture.

```
├── backup_output.sh                  <--- (V - Maintenance module)
├── CFG_illustrative.sh               <--- configuration scripts to be run under
│                                          Linux in order to launch
│                                          the illustrative example
├── CFG_illustrative_serial.sh        <--- ditto, but w/o using multiple CPUs
├── CFG_illustrative_tryton.sh        <--- ditto, but for SLURM-managed clusters
├── check_errors.sh                   <--- (V - Maintenance module)
├── commonData                        <--- contains atlases, source models and
│                                          other data relevant for all the stages
├── CONFIGURE.m                       <--- configuration script
├── DATA_PREPARATION                  <--- contains routines to prepare downloaded
│   │                                      HCP data for the Spectral Fingerprinting
│   │                                      (I - Data Preparation module)
│   ├── HCP_DATA_PREP
│   │   ├── 01_PREPROCESSING
│   │   ├── 02_COVARIANCE_GEN
│   │   ├── 03_LEADFIELD_GEN
│   │   ├── 04_LCMV_FILTER_GEN
│   │   ├── 05_RENAMING_TOOL
│   │   ├── CommonInit.m
│   │   ├── instruction.org
│   │   └── output
│   └── PARCELLATIONS_PREP            <--- contains instuctions/scripts to prepare
│       │                                  brain atlases in order to segment source
│       │                                  volume into ROI
│       │                                  (I - Data Preparation module)
│       ├── AAL116_preparation.org
│       ├── common_grid_prep.m
│       ├── Common_grids_preparation.org
│       ├── DK_individual_atlases
│       └── Schaefer100_preparation.org
├── docs
│   ├── FUNCTIONS_REFERENCE.html      <--- list of functions and their code
│   └── ToFFi_Manual.pdf              <--- toolbox manual
├── ext_tools                         <--- contains external tools and scripts
├── globalFunctionsScripts            <--- contains scripts and functions used in
│                                          more than one stage
├── PRESENTATION                      <--- contains scripts and function used to
│   │                                      visualize computed results
│   │                                      (IV - Presentation module)
│   ├── common_functions
│   ├── fingerprints
```

(verte ...)

(... continued from the previous page)

```
│   ├── identification
│   ├── network_analysis
├── restore_from_backup.sh            <--- (V - Maintenance module)
├── rm_logs_autosaves.sh              <--- (V - Maintenance module)
├── rm_output_data.sh                 <--- (V - Maintenance module)
├── RUN_SELECTED.m                    <--- script used to launch STAGES 1-8 under
│                                          single computer equipped with one
│                                          or more CPU cores
│                                          (V - Maintenance module)
├── RUN_SELECTED_SLURM.sh             <--- script used to launch STAGES 1-8 under
│                                          multiple computers equipped with one
│                                          or more CPU cores (Linux + SLURM only)
│                                          (V - Maintenance module)
├── STAGE_1                           <--- scripts, functions and output of the
│   │                                      source analysis stage
│   │                                      (II - Spectral Fingerprinting module)
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_1.sl
├── STAGE_1_Source_Analysis.info
├── STAGE_2                           <--- scripts, functions and output of the
│   │                                      Individual Fingerprinting stage
│   │                                      (II - Spectral Fingerprinting module)
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_2_Individual_SF.info
├── STAGE_2INT.sl
├── STAGE_2.sl
├── STAGE_3                           <--- scripts, functions and output of the
│   │                                      cluster pooling stage
│   │                                      (II - Spectral Fingerprinting module)
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_3_Pool.info
├── STAGE_3.sl
├── STAGE_4                           <--- scripts, functions and output of the
│   │                                      clustering evaluation stage
│   │                                      (II - Spectral Fingerprinting module)
```

(verte ...)

(... continued from the previous page)

```
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_4_Eval_Clusters.info
├── STAGE_4INT.sl
├── STAGE_4.sl
├── STAGE_5                          <--- scripts, functions and output of the
│   │                                     group fingerprinting stage
│   │                                     (II - Spectral Fingerprinting module)
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_5_Group_SF.info
├── STAGE_5.sl
├── STAGE_6                          <--- scripts, functions and output of the
│   │                                     group-level brain regions identification
│   │                                     stage (III - Analysis module)
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_6_Classification.info
├── STAGE_6INT.sl
├── STAGE_6.sl
├── STAGE_7                          <--- scripts, functions and output of the
│   │                                     individual-level brain regions
│   │                                     identification stage
│   │                                     (III - Analysis module)
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_7_Individual_Classification.info
├── STAGE_7.sl
├── STAGE_8                          <--- scripts, functions and output of the
│   │                                     network analysis stage
│   │                                     (III - Analysis module)
│   ├── functions
│   ├── output
│   └── scripts
├── STAGE_8_Network_Analysis.info
├── STAGE_8.sl
└── templates
```

One can notice that Data Preparation (I) is reflected by `DATA_PREPARATION` directory. Configuration component is expressed by a single `.sh` file with the name starting with "`CFG_`" for Linux systems or `CONFIGURE.m` file when operating on Windows.

Spectral Fingerprinting (II) is divided into five stages which are mirrored by five directories: `STAGE_1`, `STAGE_2`, `STAGE_3`, `STAGE_4`, `STAGE_5`. Files with ".info" extensions serve only as navigation help and a reminder of what each stage is responsible for. Similarly, Analysis (III) part code can be found inside `STAGE_6`, `STAGE_7`, `STAGE_8` directories. Both Spectral Fingerprinting (II) and Analysis were named "STAGES" to emphasize several common characteristics. First, I wanted to make a clear distinction between the core procedures directly transforming the data to explore brain function and side routines that are more concerned with file bookkeeping, organizing, and preparation. Secondly, they are all part of Spectral Fingerprinting original idea proposed by Keitel and Gross in their paper [9]. Thirdly, stages can be started in parallel and run in a different order (examine Fig. 4.10 to learn which stages can be run independently and which depend on previous stages).

Visualization routines from component (IV) can be accessed inside `PRESENTATION` directory.

There is no single directory for Maintenance (V). This part is just a collection of handy scripts intended to automate some management tasks and help user to keep repository clean: `backup_output.sh`, `check_errors.sh`, `restore_from_backup.sh`, `rm_logs_autosaves.sh`, `rm_output_data.sh`, `RUN_SELECTED.m`, `RUN_SELECTED_SLURM.sh`.

Next, each of the modules will be described in more detail.

4.4.1 Module I: Data Preparation

The Data Preparation module (I) is responsible for arranging sensor time-series, spatial filters, and brain parcellation data for processing to be acceptable input of the module II routines. Diagram in Fig. 4.2 shows data structures' flow across different subroutines. Names of MATLAB workspace variables are written in parenthesis and are referenced in upcoming text using a `monospace` font. File names are shown beneath for the data written onto the hard drive.

At first, data preparation routines begin with a multichannel signal (`data` variable). Signal preprocessing scripts (Fig. 4.1B, I. DATA PREPARATION) start with a selected time interval of interest divided into subsequent short-time windows (e.g., 1 second segments). Detrending is performed so the linear components present in the segments do not bias power spectra towards lower frequencies. These steps are contained in

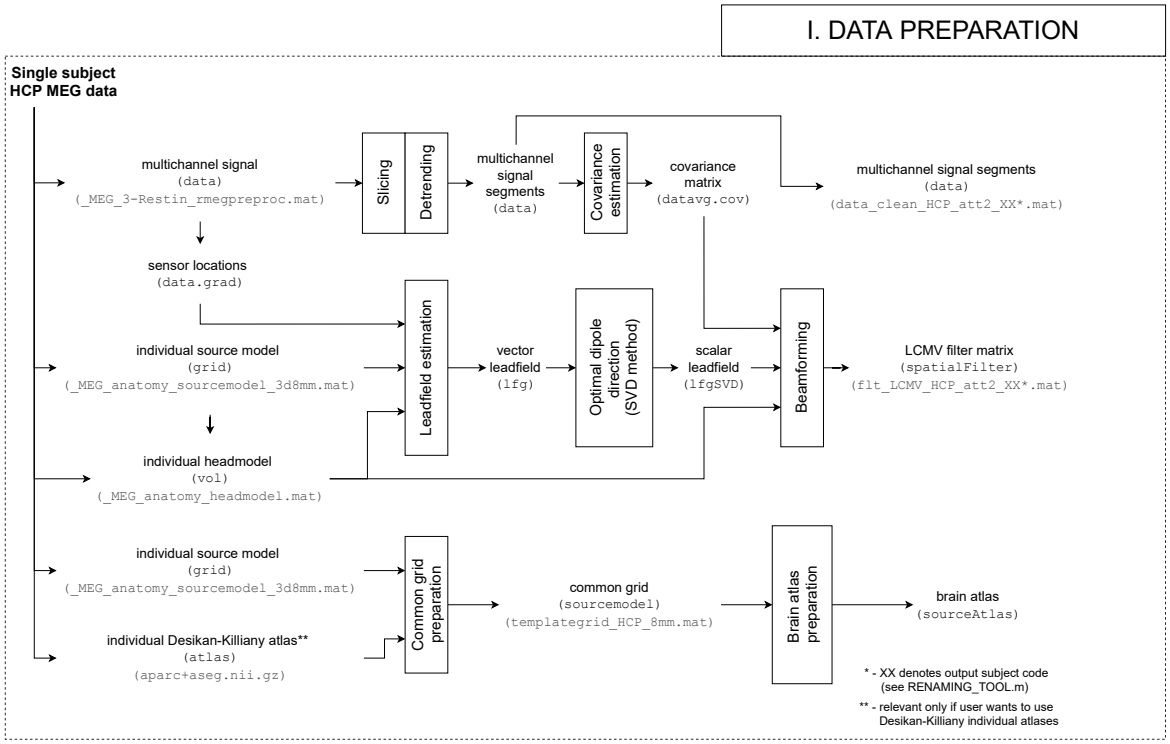


Fig. 4.2. ToFFi: Data Preparation module. Diagram of how the data are processed by the DATA PREPARATION component. Each depicted piece of data is endowed with its name, MATLAB workspace variable name (middle parentheses), and a file name (bottom parentheses) if it is read or written to the hard drive during processing.

DATA_PREPARATION/HCP_DATA_PREP/01_PREPROCESSING/PREPROCESSING.m file.

Covariance estimation scripts compute the average data covariance matrix (R^s in Step 2 of Algorithm 3; datavg.cov variable) needed to construct a spatial filter for beamforming (Eq. A.3). This procedure is coded inside DATA_PREPARATION/HCP_DATA_PREP/02_COVARIANCE_GEN/COVARIANCE_GEN.m file.

Other ingredients needed for beamforming are: sensor locations in space (data.grad variable), individual source models (3-D grids where the individual brain source space is defined; grid variable), and individual head models (describing individual brain geometry as a single shell in 3-D space; vol variable).

Knowing the sensors' location and head model, estimation of the *vector lead field* (lfg variable) is possible. Lead field defines the propagation of the unit activity in X-Y-Z coordinates inside a single grid element onto an activity spread across all the sensors (see Eq. A.2). Considering only directions of current dipoles having the strongest power (η_m defined throughout Steps 4 – 6 of Algorithm 2), a vector lead field is transformed into a *scalar lead field* ($h_{m,\mathrm{EVD}}$ in Step 6 of Algorithm 2; lfgSVD variable) using the method proposed in section 4.3 of the book by Sekihara and Nagarajan [174]. All steps related to lead fields construction can be found in DATA_PREPARATION/HCP_DATA_PREP/03_LEADFIELD_GEN/LEADFIELD_GEN.m file.

Combining scalar lead field and covariance matrix according to the Eq. A.3 produces the LCMV filter matrix (spatialFilter variable) acts like an inverse of the lead field matrix: it estimates how unit activity at the given sensor is translated into activity inside elements of the 3-D grid (source space). Generation of the spatial filters is implemented inside DATA_PREPARATION/HCP_DATA_PREP/04_LCMV_FILTER_GEN/LCMV_FILTER_GEN.m file.

Data Preparation (I) also contains routines for preparing a source grid common for all the participants (last row in Fig. 4.2; sourcemodel variable). Group brain parcellation (sourceAtlas variable) should be defined in the same space as those common grids. In the case of using an individual brain atlas (e.g., Desikan-Killiany atlases are provided with the toolbox), additional step is performed to project individually defined brain parcels into space common for all the participants to enable group-level fingerprinting. Common grid and brain atlas preparation are described in the text and code files inside DATA_PREPARATION/PARCELLATION_PREP directory.

One last script belonging to Data Preparation (I) is written inside ToFFi_Toolbox/ToFFi_Toolbox-20211013/DATA_PREPARATION/HCP_DATA_PREP/05_RENAMING_TOOL/RENAMING_TOOL.m. It can recode the names of the subjects' data files to fit demanded convention.

For this Data Preparation module, there are three crucial output data structures

(right side of Fig. 4.2). First, it is multichannel signal segments. It is represented by the variable called `data`, which took the form of a structure with the following fields:

```
data =
struct with fields:
fsample: 508.6275
hdr: [1×1 struct]
grad: [1×1 struct]
sampleinfo: [294×2 double]
trial: {1×294 cell}
time: {1×294 cell}
label: {241×1 cell}
cfg: [1×1 struct]
```

Each cell of the `trial` field contains a matrix containing 1-second long signal samples, where each row represents readings from one of the sensors. Another important field is `grad`. It contains sensor positions in 3-D space. This information was crucial for beamforming performed in STAGE 1 of the second (II) module (see Fig. 4.1B, "S1: Source reconstruction stage"). Dimensions depend on the length of the acquired signal and the number of channels. It should contain signal samples cleaned from artifacts.

Another important structure needed as an input to Spectral Fingerprinting (II) is the filter matrix (Eq. A.3; `spatialFilter` variable). It is a matrix with the number of rows equals the number of brain sources to be reconstructed (3-D grid voxels marked as those representing brain tissue; other voxels belong to the skin, skull, or be located outside the head) and the number of columns equals the number of sensors (Fig. 4.3).

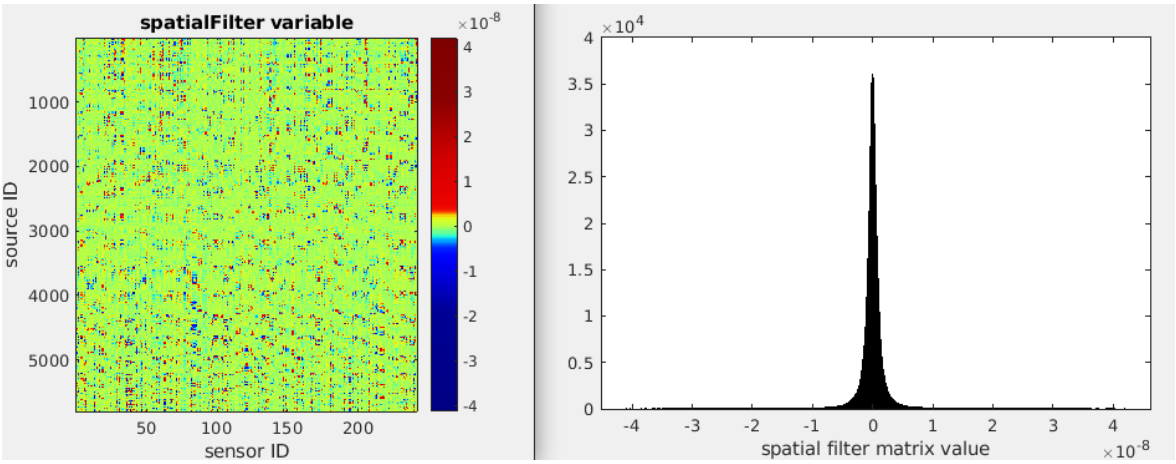


Fig. 4.3. The left panel shows an example of a spatial filter matrix for single subject. The right panel depicts the histogram of the matrix values. Each row of the matrix represents a single brain location. Entries in each row show how unit activity of neural ensembles at that location (of total 5798 locations shown here) is reinforced/attenuated across all sensors (here 241 MEG sensors).

The last output variable of this first toolbox component is the brain atlas

(`sourceAtlas` variable). The most important role of it groups source locations (voxels) considered above in spatial filter into non-overlapping regions of interest. It is a structure containing a common atlas for all subjects with the following fields:

```
sourceAtlas =
struct with fields:
dim: [20 25 22]
transform: [4×4 double]
unit: 'cm'
tissue: [20×25×22 double]
tissuelabel: {1×100 cell}
cfg: [1×1 struct]
```

`dim` is double 1-D array of positive integers containing the size of the 3D source space. Matrix called `transform` contains affine transformation matrix for mapping the voxel coordinates to head coordinate system. `tissue` is a double 3-D array with integer values from 0 to N (the value 0 means "unknown"). Values are related to the ROI labels stored in `tissuelabel` - cell of strings containing the name of each ROI.

Details about each data structure shown in Fig. 4.2 can be found in the toolbox documentation (section 6.2 therein; link to the documentation provided in Appendix C).

As it is impossible to provide a universal interface for all possible multichannel data sets, this part of the software should be rewritten by the user if there is a need to use a custom data set. The rewritten code must fulfill the specfication of the input data structures of Spectral Fingerprinting routine (II). For demonstration purposes, I have used the Human Connectome Project MEG dataset[15], which is already cleaned from artifacts [200][212][213].

Relating described steps to the pipeline presented in [9] (Fig. 3.2), Data Preparation (I) should always cover the following diagram blocks: "Preprocessing", "1-s segments", "Artefact Rejection", "MRI-MEG co-registration". It is responsibility of the user to provide data preprocessing appropriate to the application in mind to produce output fitting to the input of STAGE 1. Luckily, the Human Connectome Project MEG dataset met most of the criteria for the research I had planned.

4.4.2 Module II: Spectral Fingerprinting

Spectral Fingerprinting module (II) transforms the MEG/EEG multichannel array of signals prepared in module (I) into spatially localized power-based representations

[15]Consult WU-Minn HCP 1200 Subjects Data Release Reference Manual and its appendices for details: `https://www.humanconnectome.org/study/hcp-young-adult/document/1200-subjects-data-release` (accessed: June 4, 2022)

called *spectral fingerprints*. Fourier Transform, source reconstruction (beamforming), and Gaussian Mixture Modeling algorithms are used to compute spectral fingerprints both at the individual and the group level. It includes the first five of total eight stages (S1-8 in Fig. 4.1). Diagram in Fig. 4.4 shows data structures' flow across different subroutines.

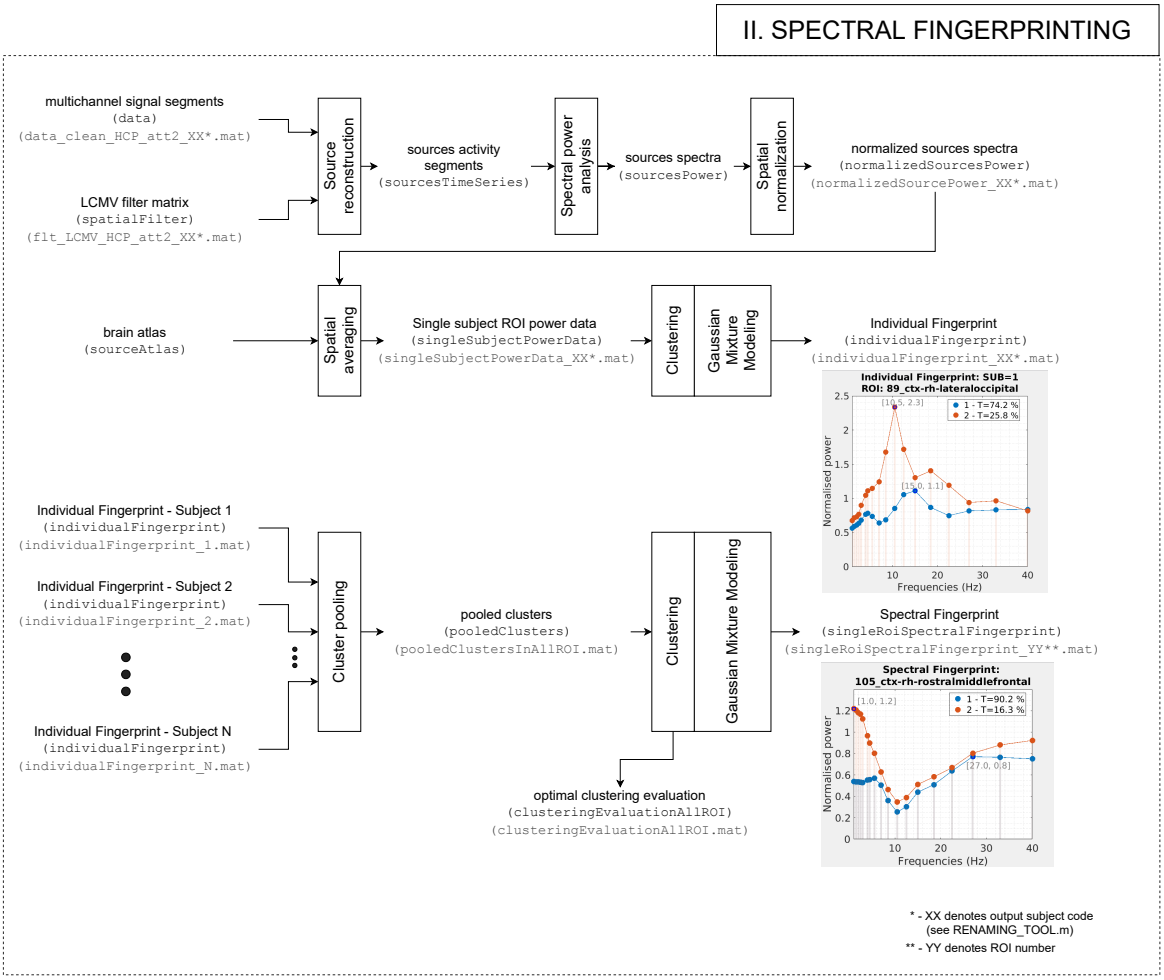


Fig. 4.4. ToFFi: Spectral Fingerprinting module. Diagram of how the data are processed by the SPECTRAL FINGERPRINTING component. Each depicted piece of data is endowed with its name, MATLAB workspace variable name (middle parentheses), and a file name (bottom parentheses) if it is read or written to the hard drive during processing.

STAGE 1 includes "Source projection" and "Spectral normalization" blocks presented in the pipeline diagram (Fig. 3.2), and equivalently "Source reconstruction", "Spectral power analysis", "Spatial normalization" blocks in Fig. 4.4.

For each participant, multichannel signal segments and LCMV filter matrix are combined according to Eq. A.5 to estimate brain activity in selected brain locations from sensors' readings. Segments of sensor activity become segments of source activity contained within `sourcesTimeSeries` variable:

```
sourceTimeSeries =
struct with fields:
trial: {1×294 cell}
time: {1×294 cell}
fsample: 508.6275
label: {5978×1 cell}
```

Cell array called `trial` contains 2D-arrays with reconstructed source activity, each 2-D array representing a single time segment. The number of rows equals the number of sources and number of columns equals the number of time segments. Each source has its working name stored in `label` cell array.

After spectral power analysis according to the procedure presented in Algorithm 1, `sourceTimeSeries` variable is transformed into `sourcesPower` variable. This step is an example of taking advantage of a function already implemented in the Fieldtrip Toolbox [196]. Listing 1 shows the code for the function `fourierAnalysis_Method1.m` responsible for moving from the time domain into the frequency domain.

Listing 1: Code of `fourierAnalysis_Method1.m` function.

```
function [powSpectra, outputFrequencies] = fourierAnalysis_Method1(cfg, timeSeries)

disp('Spectrum calculation on each trial separately ...')

% Source spectral activity calculation
cfgFreq              = [];
cfgFreq.method       = 'mtmfft';
cfgFreq.output       = 'pow';
cfgFreq.taper        = 'dpss';
cfgFreq.tapsmofrq    = 2;
cfgFreq.pad          = 2;
cfgFreq.keeptrials   = 'yes';
cfgFreq.foi          = cfg.frequencies;
powSpectra           = ft_freqanalysis(cfgFreq, timeSeries);
outputFrequencies    = powSpectra.freq;
```

It takes as an input the `sourceTimeSeries` variable (denoted `timeSeries` inside the function's scope) and some auxiliary configuration structure `cfg`, returning two outputs. `'mtmfft'` stands for '**m**ulti-**t**aper **m**ethod for **F**ast **F**ourier **T**ransform', which is described in Algorithm 1 . `'pow'` is the information for Fieldtrip subroutines to output power values instead of complex Fourier coefficients. `'dpss'` is for '**d**iscrete **p**rolate **s**pheroidal **s**equences' which is one of the types of windows used in spectral methods [214]. The estimation with tapers is more robust to noise, increases the signal-to-noise ratio of high-frequency activity, and is advised when analyzing single-trial (segmented) data [195]. `cfgFreq.tapsmofrq = 2` means that spectra will be smoothed around frequencies of interest within $\pm$ 2 Hz frequency interval i.e., inside 4 Hz smoothing box. This parameter affects Fieldtrip's automatic computation of the number of tapers to

be used. `cfgFreq.pad = 2` is the length in seconds to which the data can be padded out. In this case, the resolution is set to $1/\mathbf{2} = 0.5\,\text{Hz}$. The zero-padding technique increases the resolution of the output spectra without increasing the amount of information in the data (i.e., it does *not* increase the precision of the Fourier transform) [195][215]. `cfgFreq.foi` should be provided with the list of frequencies for which spectral fingerprints should be defined. Line 14 in Listing 1 is a call to the Fieldtrip's routine responsible for performing Fourier transformation. The second output of `fourierAnalysis_Method1.m` is `outputFrequencies` vector. It contains the final list of frequencies to be analyzed (which may not be exactly the same as the user has provided in `cfgFreq.foi` due to the reasons described in Notes to Algorithm 1) First output is the `powerSpectra` structure (named `sourcesPower` in Fig. 4.4) of the following form:

```
sourcesPower =
struct with fields:
powerSpectrum: [294×5798×42 double]
freqAxis: [1×42 double]
dimord: 'rpt_chan_freq'
label: {5798×1 cell}
```

`powerSpectrum` is a 3D array that contains values of squared Fourier coefficients for each time segment (first dimension of the matrix), source (second dimension)m and frequency (third dimension). `freqAxis` contain a vector of frequencies expressed in Hz. As was noted above, it may not be the same list user provided during the configuration step (Section 4.4.6). Its size should match the size of the third dimension of `powerSpectrum` field. This field forms a discretized frequency axis and is present among many data structures used in this toolbox. `dimord` is information added automatically by Fieldtrip for its own sake to know the order of dimensions in `powerSpectrum` matrix. Names stored in `label` cell array are not undergoing any change.

Lastly, in STAGE 1, computed power spectra undergo spatial normalization to reduce the influence of power-law behavior that complicates the identification of frequencies other than alpha (see Section 1.6). In the ToFFi Toolbox, two types of normalization are offered. "ROI-wise" is the type where power spectra amplitudes for each voxel are divided by the average spectrum amplitudes (for each frequency separately) across all the voxels belonging to a particular region of interest (ROI). However, default normalization is the division by the average spectrum of the whole brain (see Step 6 in Algorithm 3), which was also chosen by Keitel and Gross [9]. User can also refrain from normalization — toolbox configuration allows for that. After the normalization step, the output `normalizedSourcesPower` structure remains the same as `sourcesPower`

variable. Only the `powerSpectrum` field has different values inside when compared to the input `sourcesPower` variable.

I have introduced some changes relative to the original procedure documented as a beta version script cited in Appendix B. First, I decided to delete the multiplication of the Fourier spectra by 10^{14} present in line 128. The original reason for this multiplication was to prevent losing precision and causing the numerical underflow when working with very weak MEG signals. However, MATLAB offer a dense enough representation of floating-point numbers around zero (output of the command 'eps(0)', which is the distance from 0 to the next larger number, equals approximately $5 \cdot 10^{-324}$), so numerical computations involving tiny amplitudes of the MEG data loaded into MATLAB (10^{-14}) will not cause numerical precision problems. Secondly, in the beta-script (Listing 10), line 135 presents the computation of the spectral power, however, there is some ambiguous summation of Fourier coefficients multiplied with spatial coefficients. However, summation of Fourier complex coefficients of spectra obtained using different tapers is not equivalent to averaging power spectra over tapers, which should be done in the multi-tapering method [216]. Fieldtrip's default routine presented in the Listing 1 computes spectral power correctly.

STAGE 2 consists of "Area average" and "1st-level GM models" blocks presented in the pipeline diagram (Fig. 3.2), and equivalently "Spatial averaging", "Clustering", "Gaussian Mixture Modeling" blocks in the middle row in Fig. 4.4.

First, the brain regions definition is loaded into the workspace. Then voxels of the 3D-brain grid for which normalized power spectrum segments were computed are grouped into a given region of interest (ROI). Spatial averaging inside each ROI is performed according to Step 7 of Algorithm 3. After this step, a reduction occurs: from possibly thousands of voxels to only about a hundred of regions, depending on the atlas of choice. The outlier detection step, which is not directly depicted in Fig. 4.4, results in the rejection of segments that contain extreme values, according to the constraints described in Step 8 in Algorithm 3. Finally, spatial averages are stored inside a cell array `singleSubjectPowerData`. It contains as many cells as many regions are. For example, a cell regarding the fifth brain area in the atlas contains the following fields:

```
>> singleSubjectPowerData{5}
ans =
struct with fields:
subjectID: 'Sub_1'
iSub: 1
iROI: 5
tissuelabels: {1×100 cell}
goodROI: [1×100 double]
nRoiAtlas: 100
normalizedPowerSpectrum_Trials: [288×42 double]
freqAxis: [1×42 double]
nTrials: 288
rejectedTrialsIndices: [6×1 double]
nTrialsBeforeRejection: 294
```

The first three fields help to identify the subject and ROI. `tissuelabels` field stores the names of each brain region in the atlas. `goodROI` array is just a copy of a list of regions the user has chosen when configuring the toolbox. `normalizedPowerSpectrum_Trials` is a matrix containing normalized power spectra segments. Recall that the words "trial" and "segment" are used interchangeably. The number of rows is equal to the number of segments that lasted after rejecting outliers, and the number of columns equals the number of frequencies of interest (size of the frequency axis stored inside `freqAxis` field). Note that the indexing of segments has changed after rejecting some of them. There are two fields that help to recover old numeration if needed: `nTrialsBeforeRejection` is the number of segments before rejection and `rejectedTrialsIndices` is a vector of segment indices complying with segment numeration before rejection. The core step in STAGE 2 is the construction of individual fingerprints that happens in Clustering and Gaussian Mixture Modeling blocks (Fig. 4.4), described separately in detail in Algorithms 4 and 6, involved in Steps 4 and 5 of Algorithm 7. As a result, individual fingerprint (IF) is obtained for each subject separately:

```
individualFingerprint =
struct with fields:
subjectID: 'Sub_1'
iSub: 1
tissuelabels: {1×100 cell}
nRoiAtlas: 100
goodROI: [1×100 double]
spectralModes: {1×100 cell}
trialsSpectralModeMembership: {1×100 cell}
gmClassInstance: {1×100 cell}
didGmmConverge: [1 1 ... 1]
clusterDuration: {1×100 cell}
emptyClusters: {1×100 cell}
numberOfClustersInThisRoi: [1 1 2 1 ... 1 2 2]
optimalNumClustEvaluation: {1×100 cell}
nTrials: 285
freqAxis: [1×42 double]
```

Some of its fields have the same meaning as fields of `singleSubjectPowerData` structure, namely: `subjectID`, `iSub`, `tissuelabels`, `nRoiAtlas`, `goodROI`, `nTrials`, `freqAxis`. Every field that is a "`1×100 cell`" contains data related to one of the brain areas defined in the atlas (here, the atlas consists of 100 areas). `spectralModes` is a cell array in which each cell stores a double-precision 2D-array. The number of its rows is equal to the number of clusters (fixed or optimally determined using the selected optimization criterion, depending on the configuration), and the number of columns equals the number of frequencies of interest (size of the frequency axis stored inside `freqAxis` field). Each row contains a single Gaussian component center. In Spectral Fingerprinting jargon, each component is called *spectral mode*. The number of spectral modes for each region is stored in `numberOfClustersInThisRoi` field. Fig. 4.5 is an example. Here is the content of the `spectralModes{70}` calculated for some arbitrarily chosen subject no 6:

```
>> IF.spectralModes{70}
ans =
0.5889    0.5729    0.5605    0.5797    0.5575
(first row continued) 0.5786    0.6117  0.7671    0.7020    0.4865
(first row continued) 0.3664    0.4682  1.2246    1.1005    1.2526
(first row continued) 1.6353    0.8365  0.7378
(second row) 0.7733    0.7780    0.7700    0.7598    0.7490
(second row continued)    0.7150    0.6908    0.6520    0.5362    0.3367
(second row continued)    0.2578    0.3948    0.5491    0.6153    0.6745
(second row continued)    0.7467    0.7584    0.6762
```

The first row corresponds to the blue curve and the second row to the red one. `freqAxis`

field contains the following values of frequencies in Hz:

```
freqAxis: [1, 1.5, 2, 2.5, 3, 4, 4.5, 5.5, 7, 8.5, 10.5, 12.5, 15, 18.5,
           22.5, 27, 33, 40]
```

`trialsSpectralModeMembership` contains a column vector, where each entry corresponds to one segment, and its value tells to which spectral mode this segment belongs. `gmClassInstance` contains results of using `fitgmdist`[16] built-in MATLAB function applied to the data, separately for each area. `didGmmConverge` is a vector of flags (one flag per area) indicating whether EM algorithm for fitting Gaussian Mixtures has converged. Regions and subjects for which the algorithm not converged are logged inside `NOTCONVERGED.csv` files for later inspection if needed. The number of points in each cluster is proportional to its duration in relation to the duration of the whole MEG/EEG recording. These durations are computed according to Step 8 in Algorithm 7, expressed as percentages, and stored in `clusterDuration` cell array. For example, the amount of time presented as percentages in Fig. 4.5 is stored in the `clusterDuration` field:

```
IF.clusterDuration{70}
ans =
6.1818   93.8182
```

`emptyClusters` cell array contains cluster indices that have no points assigned to it. Such a situation can happen if the input data is degenerated or some of the resulting spectral modes are identical (by the wrong choice of parameters). In such situation, one of the "twins" have no points assigned to it because they were "stolen" by its "sibling". `optimalNumClustEvaluation` contains all the results of calculating optimal number of clusters in each area and stores structure in each of its cells. The structure contains three fields. `optNumClustersAllIter` and `criterionValues` are vectors with the number of entries equal to the number of repeats (also called *iterations*) of optimal number cluster estimation (Algorithm 5). Iterations were mutually independent. `optNumClustersAllIter` stores the number of clusters determined in each iteration as the best solution, and `criterionValues` stores the value of the optimization criterion for the best solution (e.g., Silhouette criterion value). Mode of `optNumClustersAllIter` values is chosen as the final number of spectral modes to be extracted from normalized power segments.

I have introduced changes relative to the original procedure documented as a beta version script (Appendix B). In beta version script provided by the authors of [9], only AAL [189] atlas with 116 anatomically defined regions was available. In ToFFi Toolbox, there are two additional brain atlases available: Schaefer parcellation with 100 regions defined considering cognitive functions supported by the given area [217], and anatomical

[16] `https://www.mathworks.com/help/stats/fitgmdist.html` (accessed: June 4, 2022)

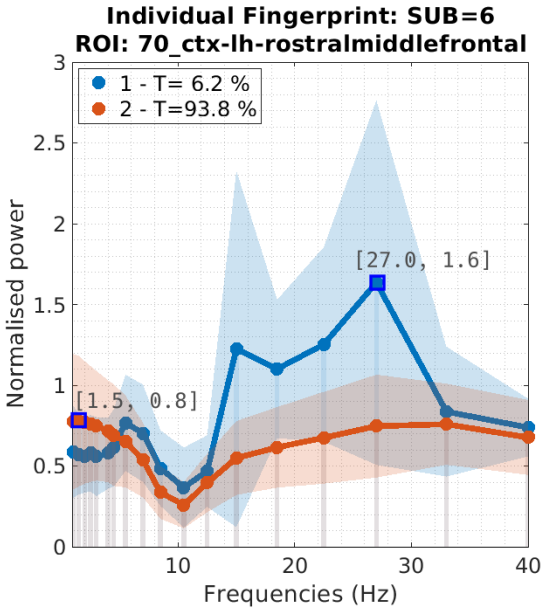


Fig. 4.5. Resting-state individual fingerprint for examplary region from Desikan–Killiany atlas in the 1–40 Hz frequency interval. Legends show the corresponding duration of each spectral mode (i.e., the percentage of trials in which each spectrum was present on average during recording). The frequency axis was configured to be logarithmic in order to optimize the lower frequencies resolution. Y-axis depicts the power normalized in relation to the average spectrum of the whole brain. Shaded regions depict the standard deviation (1σ) of the corresponding spectral mode. For i-th of total F frequencies of interest, standard deviation was estimated as $\sqrt{\Sigma_{i,i}}$, where $\Sigma_{i,i}$ is the i-th diagonal entry of the covariance matrix of the Gaussian Mixture Model component corresponding to the given spectral mode.

Desikan-Killiany parcellation [218] with 113 areas, often used in neuroscientific research. Originally, Keitel and Gross allowed the number of clusters to be a fixed value, same for all areas. In ToFFi, the alternative setting is possible, which allows for obtaining the optimal number of clusters for each area separately. Moreover, in line 165 of Listing 10, the outlier rejection is one-tailed, i.e., $z > 2.5$. It needed to be changed into a two-tailed criterion, i.e., $|z| > 2.5$, so effectively, not only extremely large power is considered abnormal but extremely small as well.

STAGE 3 is equivalent to the "Cluster pooling" block in 4.4, which was not depicted in the original pipeline diagram (Fig. 3.2). It is responsible only for gathering IFs computed in STAGE 2 and reorganizing them into a single structure convenient for further operations (see Step 10 in Algorithm 7). Resulting `pooledClusters` is an array of MATLAB structures, each representing one of the areas and having the following fields (here for exemplary area nr 5):

```
>> pooledClusters(5)
ans =
struct with fields:
numCentroidsPerSubject: [2 2 2 ... 2 2]
centroidsAllSubjects: [169×42 double]
centroidsSubjectIndices: [169×1 double]
centroidsDuration: {1×89 cell}
iROI: 5
nRoiAtlas: 100
goodROI: [1×100 double]
freqAxis: [1×42 double]
```

Meaning of fields `iROI`, `nRoiAtlas`, `goodROI`, `freqAxis` remains the same. `numCentroidsPerSubject` is a vector with the same number of entries as the number of subjects selected during toolbox configuration. Each entry is the number of clusters given subject contributed to the total number of clusters of brain area indexed with `iROI` number. `centroidsAllSubjects` stores in each row every spectral mode from every subject (number of columns corresponds to the size of the frequency axis). `centroidsSubjectIndices` stores subject ID for each spectral mode. `centroidsDuration` contains a percentage of time relative to the whole MEG/EEG recording each cluster was present, grouped into MATLAB cells — one cell per subject.

STAGE 4 is responsible for calculating the optimal number of clusters in spectral fingerprints to be constructed in STAGE 5 from the pooled clusters introduced by STAGE 3. STAGE 4 routines are not directly presented in the original pipeline (Fig. 3.2), but they are part of the "Clustering" block in the lowest row in Fig. 4.4. These

calculations rely on MATLAB's `evalclusters`[17] function. Choice of possible criteria, metric functions, and clustering algorithms are limited to ones possible with this function. The output of this stage is an array of structures called `clusteringEvaluationAllROI`. Each structure is related to one brain region, and it has the following fields (here for exemplary area nr 5):

```
>> clusteringEvaluationAllROI(5)
ans =
struct with fields:
optimalNumClusters: 2
optNumClustersAllIter: [1×600 double]
criterionValues: [1×600 double]
iROI: 5
goodRoi: [1×100 double]
```

`optNumClustersAllIter` and `criterionValues` are vectors with the number of entries equal to the number of iterations of optimal cluster number evaluation (Algorithm 5). Iterations are mutually independent. Note that this time toolbox groups points (i.e., spectral modes, not segments) belonging to all the subjects together in comparison to the evaluation made in STAGE 2. `optNumClustersAllIter` stores number of clusters determined in each iteration as the best solution and `criterionValues` stores values of the optimization criterion for these solutions (e.g., Silhouette criterion values). Mode of `optNumClustersAllIter` values is chosen as the final number of spectral modes to be extracted from segments stored in `optimalNumClusters` field.

For this stage, the change in the original beta-script code was introduced. Instead of taking the solution with the maximum criterion value across iterations, as it is coded in the beta-script, ToFFi chooses the most frequent solution. It is justified to prevent accidentally choosing some degenerated solution that resulted from odd random k-means initiation (k-means for cluster evaluation performed in Steps 3b and 11b of Algorithm 7) of the evaluation algorithm in one of its iterations. During the toolbox development, it was validated that before introducing described change in the code, data distribution often suggested fewer clusters that it was determined using the maximum criterion value. Choosing the most frequent solution resulted in adequate data clustering.

STAGE 5 covers "2nd-level GM models" block presented in the pipeline diagram (Fig. 3.2), and equivalently "Clustering", "Gaussian Mixture Modeling" blocks in the lowest row in Fig. 4.4.

Having determined optimal number of spectral modes to be constructed from data pooled in STAGE 3, the toolbox is ready to group clusters from individual subjects into

[17] `https://www.mathworks.com/help/stats/evalclusters.html` (accessed: June 4, 2022)

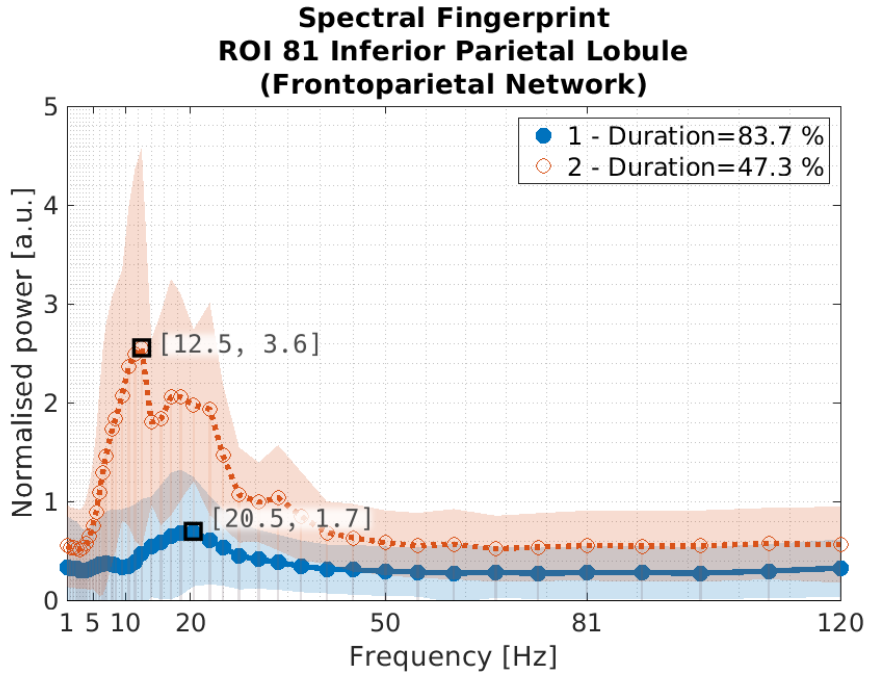


Fig. 4.6. A spectral fingerprint of the inferior parietal lobule. For this particular region, it consists of two spectral modes. Each mode is one of the centroids found by the clustering algorithm and shows normalized power for $F = 42$ frequencies of interest. Shaded regions depict the standard deviation (1σ) estimated from the covariance matrix of the Gaussian Mixture Model component corresponding to the given spectral mode. The first mode peaks at 12.5 Hz, and the second mode peaks at 20.5 Hz. The frequency axis resolution is set to logarithmic to optimize spectral analysis resolution of lower frequencies. Duration is shown as a percentage of time segments in which each spectral mode was present on average during recording.

another set of spectral modes, this time describing the group-level patterns of brain activity. Similarly, as in STAGE 2, k-means clustering is followed by Gaussian Mixture Modeling (Steps 12 and 13 in Algorithm 7). A set of resulting clusters describing single area activity as modes of normalized power spectra is called spectral fingerprint (SF). Each region has its own separate SF, which is a set of spectral modes characteristic of that area. SF is represented in the following and final output MATLAB structure produced in Spectral Fingerprinting (II) module (here is an example for area nr 5):

```
>> singleRoiSpectralFingerprint
singleRoiSpectralFingerprint =
struct with fields:
iROI: 5
nRoiAtlas: 100
goodROI: [1×100 double]
freqAxis: [1×42 double]
lvl1_numCentroidsPerSubject: [2 2 2 ... 2 2]
lvl1_centroidsFromAllSubjects: [169×42 double]
lvl1_centroidsSubjectIndices: [169×1 double]
lvl1_centroidsDuration: {1×89 cell}
lvl2_kMeans_pointsClusterIndices: [169×1 double]
lvl2_kMeans_centroids: [2×42 double]
lvl2_kMeans_withinClusterDistance: [2×1 double]
lvl2_gmm_centroids: [2×42 double]
lvl2_gmm_pointsMembership: [169×1 double]
lvl2_gmm_gmClassInstance: [1×1 gmdistribution]
lvl2_gmm_didGmmConverge: 1
lvl2_gmm_centroidDuration: [33.0176 85.0028]
lvl2_gmm_subjectsContributed: {[43×1 double]  [88×1 double]}
lvl2_gmm_nSubjectsPerCentroid: [43 88]
isOnlyPopularClustersShown: 0
```

Several fields repeat from `individualFingerprint` structure presented above ("lvl1_" and "lvl2_" prefixes differentiate between data from individual fingerprints and group-level fingerprints), namely: `lvl1_numCentroidsPerSubject`, `lvl1_centroidsFromAllSubjects`, `lvl1_centroidsSubjectIndices`, `lvl1_centroidsDuration`. Apparent redundancy was done for convenience in processing the computed fingerprints, as some of the features can be computed or visualized from both individual and group level data. This may be a suboptimal solution that could be improved, however, when the toolbox was developing, to meet the deadlines, more focus was on delivering working software, with optimization to be done later. As this software is publicly available on GitHub, it invites contributions from the community.

`lvl2_kMeans_pointsClusterIndices` is a column vector containing a cluster label, one per each of the data points grouped during the first part of the "Clustering" block presented in Fig. 4.4 last row (k-means algorithm by default). It tells which of the spectral modes from IFs belong to which group-level spectral mode inside a given SF. Similarly, after Gaussian Mixture Modeling `lvl2_gmm_pointsMembership` field is constructed, which describes the final assignment between modes from pooled IFs and SF modes. `lvl2_kMeans_centroids` and `lvl2_gmm_centroids` contains group-level spectral modes. Again, a number of rows corresponds to the number of clusters and the number of columns to the number of frequencies (the length of `freqAxis` field). Similarly to the `individualFingerprint` structure, `lvl2_gmm_gmClassInstance` contains results of using `fitgmdist` built-in MATLAB function applied to the data. `lvl2_gmm_didGmmConverge` is a flag indicating whether EM algorithm (Algorithm 6) used for fitting Gaussian Mixtures converged. Again, `NOTCONVERGED.csv` file, if present, help to identify any problem with convergence. The durations of spectral modes are calculated according to Step 17 of Algorithm 7 and stored inside `lvl2_gmm_centroidDuration` vector.

Keitel and Gross, in their article, emphasized that final group-level fingerprints should contain only modes that are composed of clusters belonging to the majority of subjects. Two vectors: `lvl2_gmm_subjectsContributed` and `lvl2_gmm_nSubjectsPerCentroid` document which subjects and how many contributed to each spectral mode. Each entry of these fields corresponds to a separate spectral mode. This way, further analyses or visualization routines can process either all of the modes or only those common to the majority of the participants. Lastly, `isOnlyPopularClustersShown` is a safe flag: if its value equals 0, it means that `singleRoiSpectralFingerprint` stores data related to all the modes, and when its value equals 1, this signifies that data stored in this structure contains only the most "popular" modes. Without `isOnlyPopularClustersShown`, one could not be sure if stored spectral fingerprints pertain to all of the subjects or only the majority.

To illustrate some of the important fields, Fig. 4.6 shows spectral fingerprint with two modes and corresponding `lvl2_gmm_centroids`, `freqAxis`, `lvl2_gmm_centroidDuration` fields present themselves as follows:

```
>> singleRoiSpectralFingerprint.lvl2_gmm_centroids
ans =
Columns 1 through 23
0.3331    0.3256    0.3213    ...  0.5883    0.6489    0.6836
0.5483    0.5327    0.5290    ...  1.8353    2.0561    2.0543
Columns 24 through 42
0.6925    0.6063    0.5308    ...  0.2724    0.2935    0.3240
1.9725    1.9319    1.4656    ...  0.5476    0.5724    0.5602
>> singleRoiSpectralFingerprint.freqAxis
ans =
Columns 1 through 23
1.0000    1.5000    2.0000    ...   15.5000   17.0000   18.5000
Columns 24 through 42
20.5000   23.0000   25.0000   ...   98.5000  109.0000  120.0000
>> singleRoiSpectralFingerprint.lvl2_gmm_centroidDuration
ans =
83.6851   47.3079
```

First row in `lvl2_gmm_centroids` field and the first entry in `lvl2_gmm_centroidDuration` fields correspond to the blue curve and second row to the red curve in Fig. 4.6.

4.4.3 Module III: Analysis

The Analysis module (III) consists of additional routines that can perform analysis on particular output files from module (II). Currently, the following analyses were implemented: group-level brain regions identification (STAGE 6), individual-level brain regions identification (STAGE 7), and regional clustering (network analysis; STAGE 8). New stages can be added in the future if needed.

Diagram in Fig. 4.7 shows data flow across different subroutines of the current implementation component.

STAGE 6 aims to measure how characteristic are regional spectral fingerprints (obtained from STAGE 5) and how well one can identify them. This can tell whether measured brain activity is specific enough to guess which fingerprint belongs to which brain region.

It was part of the original pipeline presented by Keitel and Gross [9]. Its diagram is shown in the upper row in Fig. 4.7. In ToFFi, this stage was implemented differently than in the original publication. Both original and current versions of the procedure are presented in Algorithm 8 and 10, respectively.

Group-level identification is based on the Cross-Validation procedure (Algorithm 10). Cross-Validation is a statistical method of evaluating a learning algorithm by dividing data into two subsets: training (one used to learn or train models) and validation (used

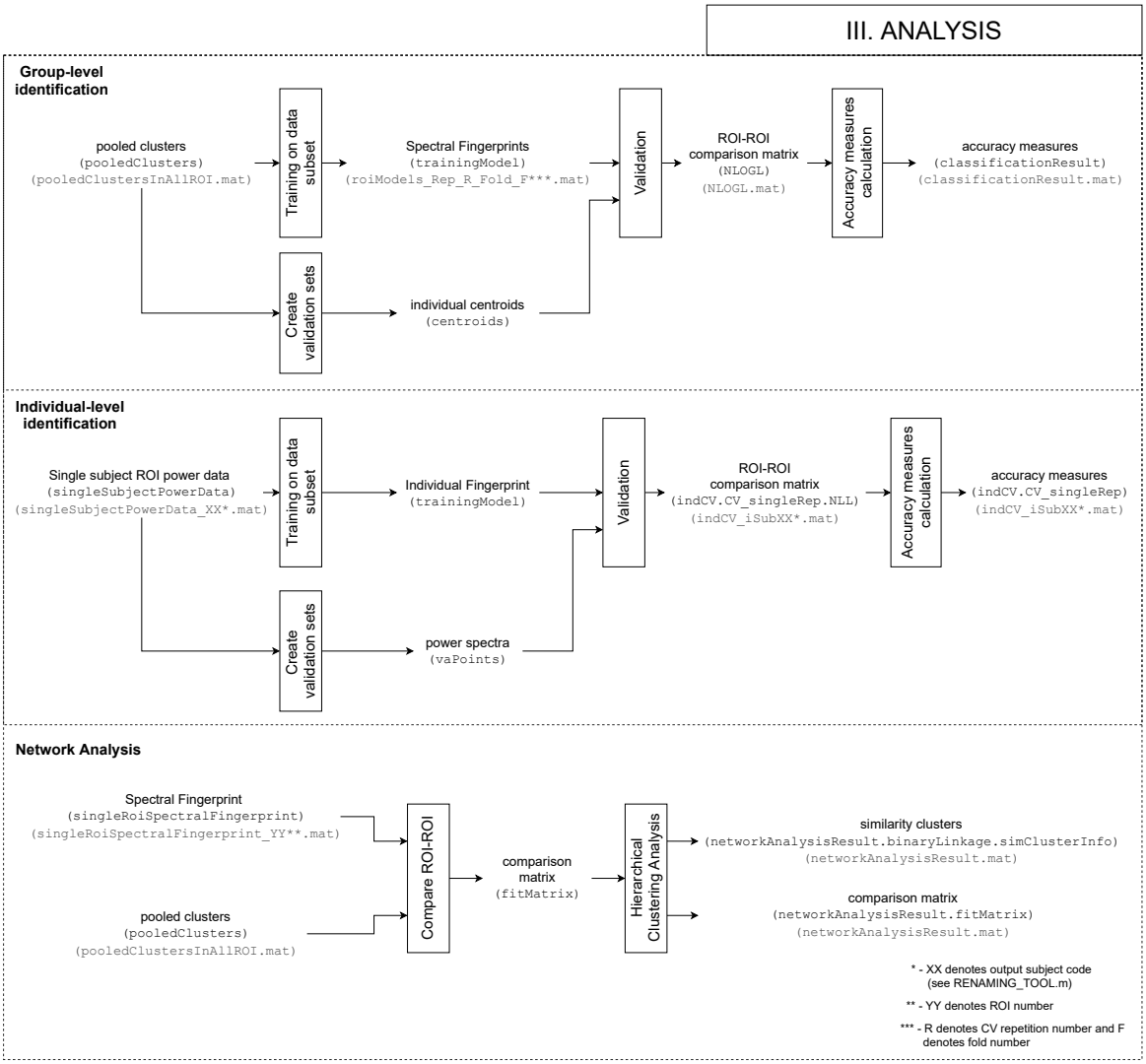


Fig. 4.7. Diagram of how the data are processed by the ANALYSIS component. Each depicted piece of data is endowed with its name, MATLAB workspace variable name (middle parentheses), and a file name (bottom parentheses) if it is read or written to the hard drive during processing.

to validate the model). The training and validation sets must cross over in successive rounds such that each data point has a chance of being validated against. In ToFFi, subjects are randomly assigned to folds. Each data fold contains individual fingerprints' clusters (extracted from `pooledClusters` variable in Fig. 4.7 obtained from STAGE 3). Then spectral fingerprints are constructed from the clusters pooled in the training folds - for each ROI separately. These fingerprints are contained within `trainingModels` MATLAB array of structures (one structure per brain area):

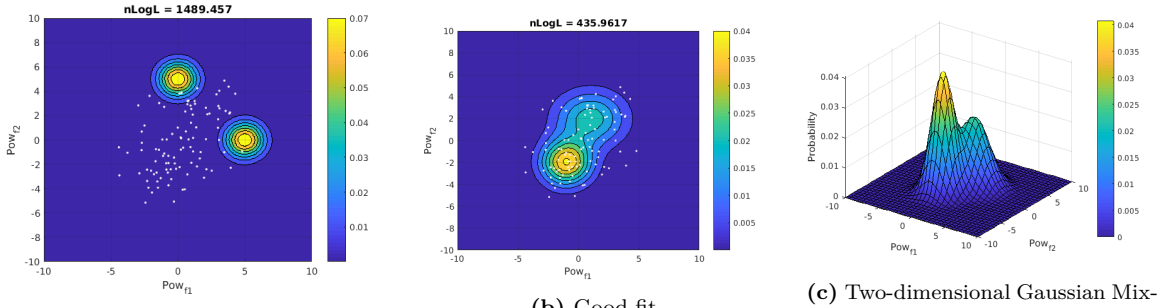
```
trainingModel =
1×100 struct array with fields:
iROI
nRoiAtlas
goodROI
centroidsFromTrSub
centroidsTrSubIndices
kMeans_pointsClusterIndices
kMeans_centroids
kMeans_withinClusterDistance
gmm_centroids
gmm_pointsMembership
gmm_gmClassInstance
gmm_didGmmConverge
centroidDuration
nSubPerCentroid
subjectsPerCentroid
nonPopularModeFlags
```

Note that (unsurprisingly) it shares many of the same fields as in `singleRoiSpectralFingerprint` variable described earlier.

These spectral fingerprints serve as regional training models. Spectral modes from validation folds (centroids variable in Fig. 4.7) are tested against trained models using the formula expressed by Eq. A.27, giving rise to a pairwise-regional comparison matrix containing negative log-likelihood values acting as a fitness measure stored `NLOGL` variable. `NLOGL` has a more peculiar design. For example, assume that the user computed group-level identification with 5 independent repetitions of 10-fold cross-validation for data from 20 subjects to study 8 brain areas. Then, examination of the negative log-likelihood matrix for the second validation subject in the third repetition fifth fold may look like this:

```
>> NLOGL
NLOGL =
struct with fields:
Reps: [1×5 struct]
>> NLOGL.Reps(3)
ans =
struct with fields:
iRep: 3
Folds: [1×10 struct]
>> NLOGL.Reps(3).Folds(5)
ans =
struct with fields:
nLogL: [8×8x2 double]
>> NLOGL.Reps(3).Folds(5).nLogL(:,:,2)
ans =
1.0e+03 *
-0.0380   -0.0394   -0.0241    0.0224   -0.0287   -0.0228   -0.0187   -0.0323
-0.0370   -0.0382   -0.0244    0.0245   -0.0270   -0.0232   -0.0178   -0.0316
0.9521    1.3288    0.1437    0.1155    1.2425    0.0624    0.2077    1.4153
1.5597    1.6437    0.6672   -0.0023    0.4764    0.3463    0.1662    0.3810
0.0726    0.0780    0.0293    0.0006   -0.0073    0.0055   -0.0052   -0.0159
0.3594    0.4985    0.0709    0.0282    0.3500    0.0141    0.0592    0.4392
0.9313    0.9712    0.4194   -0.0052    0.2458    0.2034    0.0978    0.2002
0.0561    0.0581    0.0228    0.0030   -0.0116    0.0018   -0.0053   -0.0206
>>
```

The lower the `nLogL` values, the better fit between the trained model and validation data (Fig. 4.8).



(a) Poor fit.

(b) Good fit.

(c) Two-dimensional Gaussian Mixture Model.

Fig. 4.8. Fit value interpretation. Validation example for two-dimensional data. Validation data (white points) and trained regional model (contour lines) should match. Better fit is reflected in a lower value of the nLogL measure.

Finally, average regional accuracy/rank is computed (Algorithm 11). For the next iteration, another fold of data is selected as a validation fold, and other folds are used for training. The training and validation phase repeats, then the iteration is concluded

with the average accuracy/rank computation. After k iterations, a single repetition of cross-validation is completed. One can set more than one repetition of cross-validation to be launched, and the averaged results across repetitions will be obtained as well. Each repetition yields different results because subjects are assigned to the folds randomly.

To recall, *mean rank* is based on the idea from [9]. It is a positive real value, constrained between 1 and the number of areas taken into consideration (for example, if one wants to perform identification among 10 ROIs, mean rank will be constrained between values 1 and 10). It tells how hard it is to identify a particular ROI. A mean rank of 1 indicates that the model of the target area was the most likely to fit the tested area; a rank of 2 indicates it was the second most likely, and so on. I have additionally proposed the *accuracy* measure as a more intuitive and also more conservative measure than mean rank. It is a non-negative real value between 0 and 1 (equivalent to percent), independent of the number of ROIs taken into consideration. It tells how often a particular ROI is correctly recognized among the others. An accuracy value of 1 means it fits the target area 100% times. A value of 0 means there were no correct guesses at all, and the spectral fingerprint of test ROI is totally confused with fingerprints of other regions.

After all repetitions, identification results are summarized inside the `classificationResult` variable:

```
>> classificationResult
classificationResult =
struct with fields:
nReps: 5
nFolds: 10
CV_reps: [1×5 struct]
accPerRoiMeanReps: [1×113 double]
acc_std_PerRoiStdReps: [1×113 double]
meanRankPerRoiMeanReps: [1×113 double]
meanRank_std_PerRoiStdReps: [1×113 double]
```

Identification accuracy values averaged across all the repetitions are stored inside `accPerRoiMeanReps` vector. Note that in the current example, it contains 113 elements (number of areas in the given brain atlas), but if only eight regions are of interest, then only eight entries will have meaningful values (the rest of the entries are zeroed). The standard deviation of the mean accuracy across repetitions is stored in `acc_std_PerRoiStdReps` vector. Analogously, the average mean rank and its standard deviation across repetitions are stored in `meanRankPerRoiMeanReps` and `meanRank_std_PerRoiStdReps` variables, respectively. Results prior to averaging across repetitions can be accessed at `CV_reps` field and subsequent fields with self-explanatory

names.

STAGE 7 is depicted in the middle row in Fig. 4.7. This stage aims to measure how specific are the individual regional fingerprints (obtained from STAGE 2), how well one can tell them apart. This can tell whether measured brain activity is characteristic enough so it can be guessed which fingerprint belongs to which brain region. he difference between STAGE 6 and this stage is that in STAGE 7 the toolbox conducts identification at the level of a single subject instead of doing it at the group-level.

STAGE 7 was not initially proposed in the Keitel and Gross work [9], presumably because of the required computational load. Thanks to the concurrency implemented in ToFFi, computation time was decreased reasonably to implement identification on an individual level.

Normalized power spectra segments from a single subject (`singleSubjectPowerData` variable from STAGE 2) from multiple regions are pooled together and then distributed randomly across the folds. Then individual fingerprints (`trainingModel` variable in Fig. 4.7) are constructed from the segments pooled in the training folds for each ROI separately. These individual fingerprints will serve as regional training models. Segments from validation folds (`vaPoints` variable) are tested against trained models using the formula Eq. 3c.

For the next iteration, another fold of data is selected as a validation fold, and others serve as training folds. After all, folds having a chance being validated, a single repetition of cross-validation is completed. Average regional accuracy/rank is then computed (Algorithm 13). One can set more than one repetition of cross-validation to be launched, and the averaged results across repetitions will be obtained as well. Each repetition yields different results because power segments are assigned to the folds randomly.

For this stage, output data is organized a bit differently than it is in STAGE 6. As each subject undergoes his own cross-validation repeated many times, `indCV_iSubXX.mat` files are saved, where XX denotes subject ID. Inside each file, there is a structure called `indCV` with the following fields:

```
>> indCV
indCV =
struct with fields:
CV_avg: [1×1 struct]
CV_singleRep: [1×10 struct]
```

Inside `CV_avg` field, results averaged across repetitions can be found:

```
>> indCV.CV_avg
ans =
struct with fields:
acc_avg_acrossReps: [1×113 double]
acc_std_acrossReps: [1×113 double]
mr_avg_acrossReps: [1×113 double]
mr_std_acrossReps: [1×113 double]
```

Accuracy measures are stored in a similar fashion as it is in STAGE 6. Mean accuracy and its standard deviation across repetitions are stored in `acc_avg_acrossReps` and `acc_std_acrossReps`, respectively. Mean rank average and mean rank standard deviation are stored in `mr_avg_acrossReps` and `mr_std_acrossReps`, respectively. Results prior to averaging across repetitions can be accessed at `indCV.CV_singleRep` field and subsequent fields with self-explanatory names. ROI-to-ROI comparison matrix containing fitness values quantified by negative log-likelihood matrix can be accessed as follows. First, the user needs to load the individual identification result of one of the subjects:

```
>> load('indCV_iSub3.mat')
```

Then, to access results from, for example, the third repetition and second fold, the user needs to type in MATLAB's Command Window:

```
>> indCV.CV_singleRep(3).NLL.Folds(2)
ans =
struct with fields:
nLogL: [8×8 double]
>> indCV.CV_singleRep(3).NLL.Folds(2).nLogL
ans =
1.0e+04 *
-0.0794   -0.0765    0.0289    0.0520   -0.0343    0.0203    0.0690   -0.0352
-0.0868   -0.0952    0.0331    0.0550   -0.0329    0.0246    0.0699   -0.0394
0.7550    1.9513    0.0231    0.0925    0.1654    0.0299    0.0787    0.4118
1.3819    2.9156    0.1460    0.0445    0.2570    0.1559    0.0501    0.3759
0.0761    0.2406    0.0093    0.0289   -0.0296    0.0135    0.0450   -0.0207
0.7617    1.7282    0.0337    0.0757    0.1616    0.0380    0.0743    0.3209
2.0920    4.0279    0.2068    0.0626    0.4905    0.2172    0.0565    0.6763
0.1037    0.2161    0.0032    0.0222   -0.0285    0.0088    0.0369   -0.0455
```

The lower the `nLogL` values, the better fit between the trained model and validation data.

STAGE 8 is depicted in the lowest row in Fig. 4.7. It aims to compare spectral fingerprints obtained as a result of STAGE 5. Regional fingerprints are compared pairwise and then assigned to clusters. Clusters are compared and successively joined

to form a binary tree. This procedure utilizes the hierarchical clustering algorithm for this purpose. In their seminal paper, Keitel and Gross described the method under the section called "Similarity across Areas" [9].

First, spectral fingerprints obtained from STAGE 5 and pooled clusters computed in STAGE 3 are loaded into the workspace (`singleRoiSpectralFingerprint` and `pooledClusters` variables, respectively). Each pair of regions are compared according to the procedure described in Algoritm 14. This way, the comparison matrix is formed, which is represented by `fitMatrix` variable. Then each row of the similarity matrix was treated as a separate point, and similarity between points was used to form a binary tree using UPGMA procedure [219]. Both binary tree and fit matrix are saved inside `networkAnalysisResult` structure:

```
networkAnalysisResult =
struct with fields:
binaryLinkage: [1×1 struct]
fitMatrix: [113×113 double]
```

The fit matrix is of size A^2, where A is the number of ROI in the brain atlas. If the list of regions to be analyzed is smaller than A, some entries of the matrix remain `NaN` ("not-a-number"), and only a few entries are used. This is obviously not a memory-efficient strategy. However, it helped prevent errors in managing ROI indices between stages. A better solution could be proposed and implemented in future revisions of the toolbox.

The result of hierarchical clustering is stored in the `binaryLinkage` field:

```
>> networkAnalysisResult.binaryLinkage
ans =
struct with fields:
Linkage: [7×3 double]
hierarchicalClusterTreeLeafs: [7×2 double]
TreeFull: [1×1 struct]
TreeTruncated: [1×1 struct]
nSimilarityClusters: 4
simClusterInfo: [1×1 struct]
```

The most important field is `Linkage`. It contains three columns, for example, for $A = 8$ regions being analyzed:

```
>> b.Linkage
ans =
4.0000    7.0000    0.0002
1.0000    2.0000    0.0039
3.0000    6.0000    0.0040
5.0000    8.0000    0.0086
9.0000   11.0000    0.0328
12.0000   13.0000    0.1033
10.0000   14.0000    1.2184
```

Each row corresponds to one newly formed cluster with the first two columns containing indices of linked regions (warning: each index corresponds to the position in the list of areas selected for the analysis, not ROI ID in the brain atlas containing all possible regions), and the third column contains the linkage distance between the two clusters merged in this row. If there are $A = 8$ regions to be linked, I-th row receives a new cluster index equals $A + I$. This means that subsequent rows get indices: $9, 10, 11, 12, 13, 14, 15$. These numbers are present in the `Linkage` columns. For example, a row containing 9 and 11 means that cluster 9 and cluster 11 were connected. Number 9 corresponds to a cluster formed in $9 - 8 = 1$ row, i.e., region 4. and region 7. Number 11 corresponds to the cluster formed in $11 - 8 = 3$ row, i.e., region 3. and region 6. `hierarchicalClusterTreeLeafs` field contains the first two columns of the Linkage field. `TreeFull.leafNodesIndices` contains a sequence $1, 2, \ldots, A$ indexing regions being analyzed. `TreeTruncated.leafNodesIndices` assigns new cluster indices to regions number in `TreeFull.leafNodesIndices` vector:

```
>> b.TreeTruncated.leafNodesIndices'
ans =
1     1     3     4     2     3     4     2
```

This means that the first two regions were linked together to form a cluster denoted as "1". Similarly, third and sixth region constitute a cluster denoted as "3", and so on. Maximum number depends on how many clusters were set to be formed in configuration (at the level of how many clusters the tree was truncated) and is equal to the number stored in `nSimilarityClusters` field. `simClusterInfo` contains information about the linked regions:

```
>> b.simClusterInfo
ans =
struct with fields:
indices: {[6 25]  [70 105]  [54 89]  [67 102]}
label: {{2×1 cell}  {2×1 cell}  {2×1 cell}  {2×1 cell}}
```

`indices` field index areas with their indices from the atlas, which helps identify them. For convenience, the names of the linked regions are stored in cell arrays inside `label`

field.

As one can see, some of the fields are redundant and may be deleted in future releases of the toolbox to increase readability and memory efficiency.

4.4.4 Module IV: Presentation

The Presentation module (IV) is a collection of auxiliary scripts used to visualize particular results of performed computations for easier interpretation. It aids the researcher in answering some of the questions that might have arisen during result interpretations and helps to make a decision about what questions should follow or what other analyses are needed.

There are three main parts of Presentation (IV) related to three distinct types of results that can be obtained with the toolbox. Stages 1 to 5 allow to construct spectral fingerprints (1), stages 6 and 7 check fingerprints specificity via identification procedures (2), and in Stage 8 similarity between fingerprints is assessed (3). Each part is placed in a separate directory. The content of each directory is displayed below:

(1) `PRESENTATION/fingerprints/`:

- `plotIndividualFingerprint.m`,
- `plotSpectralFingerprint.m`.

(2) `PRESENTATION/identification/`:

- `groupIdentificationPlot.m`,
- `groupHitMatrix.m`,
- `individualIdentificationPlot.m`,
- `indHitMatrix.m`.

(3) `PRESENTATION/network_analysis/`:

- `networksPlot.m`.

These `.m` scripts are not launched automatically. The user needs to open one of the directories, then open one of the scripts, configure `%% SETTINGS` section in the script according to his needs (proper guidance is provided in code comments), and run the script. After a few moments, visualizations are ready and displayed.

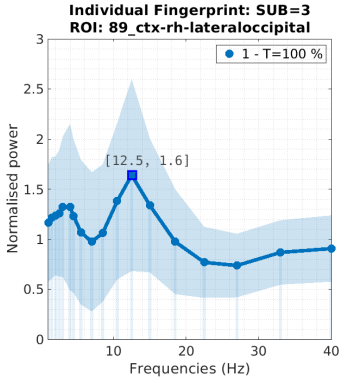
`plotIndividualFingerprint.m` script allows to visually examine `individualFingerprint` data structure described in the previous sections. To remind, individual fingerprint (IF) represents clustered normalized power spectra obtained

from reconstructed multichannel recording describing a given brain area activity of a single subject (called "1st-level GM models" in [9], computed in Algorithm 7). Such visualization can help answer questions like:

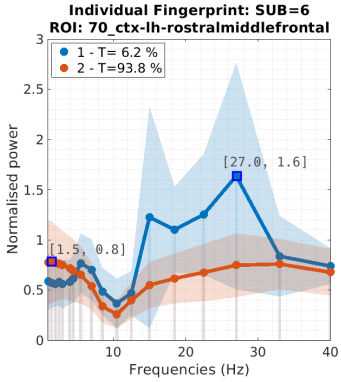
- What are frequency peaks of brain activity for a given region and subject?
- Which spectral modes are dominant, i.e., exist in the largest percentage of recording duration?
- How variable spectral modes are?

User can adjust how many subjects and how many brain areas should be displayed after single script execution. Line widths, marker sizes, and opacity can be adjusted. Power peaks can be labeled at will. When analyzing a large number of cases, the user can also turn off the figure display and decide on saving figures to the hard drive for later examination. Figures are saved both in `.fig` and `.png` formats.

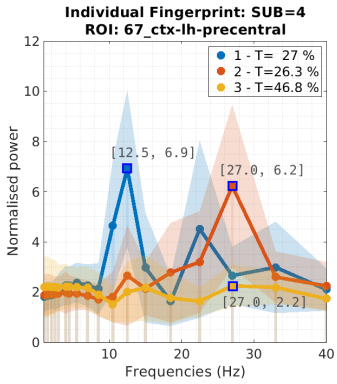
Fig. 4.9 shows an example of three IFs characterizing spatially distant brain regions (for their location on the cortex, see Fig. 4.30).



(a) Right hemisphere occipital fingerprint for subject no. 3. Single spectral mode with a clear dominance of the alpha peak visible.



(b) A fingerprint of the anterior middle frontal gyrus for subject no. 6. Two spectral modes, one of which is dominant (orange) and the other is supportive (blue). See the text for the explanation.



(c) A fingerprint of the left precentral gyrus for subject no. 4. Multiple spectral modes peaking at different frequencies with no clear dominance visible.

Fig. 4.9. Resting-state individual fingerprints for Desikan–Killiany atlas in the 1–40 Hz frequency interval. Legends show the corresponding duration of each spectral mode (i.e., the percentage of trials in which each spectrum was present on average during recording). The frequency axis was configured to be logarithmic in order to optimize the lower frequencies resolution. Y-axis depicts the power normalized in relation to the average spectrum of the whole brain. Shaded regions depict the standard deviation (1σ) of the corresponding spectral mode. For i-th of total F frequencies of interest, standard deviation was estimated as $\sqrt{\Sigma_{i,i}}$, where $\Sigma_{i,i}$ is the i-th diagonal entry of the covariance matrix of the Gaussian Mixture Model component corresponding to the given spectral mode. Different types of fingerprints can occur: a) fingerprints with single spectral mode, b) fingerprints with dominant and supportive spectral modes, and c) fingerprints with more than two modes, switching between one another during the recording. Regions tend to differ in terms of peaking frequencies.

`plotSpectralFingerprint.m` script help to visually examine

`singleRoiSpectralFingerprint` structure described above. To remind, spectral fingerprint (SF) represents clustered normalized power spectra obtained from reconstructed multichannel recording describing brain area activity common for the analyzed group of subjects (called also "2nd-level GM models" in [9], computed in Algorithm 7). Such visualization can help answer questions like:

- What are frequency peaks for a given brain region?
- Which spectral modes are dominant, i.e., exist in the largest percentage of the recording?
- How variable spectral modes are?
- Which modes of activity are stable, i.e., common for the majority of the analyzed subjects?

A user can adjust how many brain areas should be displayed after a single run of the script. Line widths, marker sizes, and opacity can be adjusted. Power peaks can be labeled at will. When analyzing a large number of cases, the user can also turn off the figure display and decide on saving figures to the hard drive for later examination. Figures are saved both in `.fig` and `.png` formats. Fig. 4.33 shows an example of three SFs characterizing spatially distant brain regions (for their location on the cortex, see Fig. 4.30).

`groupIdentificationPlot.m` allows for generating a bar plot with the summary result of the group-level identification procedure (STAGE 6). The X-axis labels brain areas taken for the analysis, and Y-axis depicts accuracy or mean rank, depending on the user's choice (Fig. 4.32a). The result can be displayed as an average of all cross-validation repetitions or for each repetition separately (mean across folds). One can display all the regions or only the subset of all areas. The user can also turn off the figure display and decide on saving the figure to the hard drive for later examination. The figure is saved both in `.fig` and `.png` formats.

Another visualization routine related to STAGE 6 is `groupHitMatrix.m` script. It allows for closer examination of which areas tend to be confused with other regions. This information is given in a form that mimics *confusion matrix* (or *error matrix*), well known in machine learning [220]. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. Here different classes correspond to different regions of interest. These visualizations can help answer questions like:

- Which regions are the easiest/hardest to be recognized from its spectral fingerprint?
- Which regions have similar fingerprints, and thus they are often confused?

- Do homologous (i.e., located symmetrically to the longitudinal fissure) regions share similar fingerprints or are they distinct?

The user can adjust visualization by deciding what subset of regions the matrix should display and what range of repetitions and folds it should take to account. The user can also turn off the figure display and decide on saving the figure to the hard drive for later examination. The figure is saved both in `.fig` and `.png` formats. An exemplary result of running `groupHitMatrix.m` script is shown in Fig. 4.32b.

Similarily to the group-level analyses, there are two scripts related to the visualization of the individual-level identification procedure (STAGE 7): `individualIdentificationPlot.m` and `indHitMatrix.m`. They help to answer similar questions to those addressed by `groupIdentificationPlot.m` and `groupHitMatrix.m` scripts, but in the context of single subjects. The structure of these scripts and generated pictures are almost the same, so further description would be redundant, and thus it is omitted. For details, one can consult the toolbox documentation (p. 21 therein; link to the documentation provided in Appendix C).

The last script in the Presentation (IV) component is `networksPlot.m` script used to visualize the results of STAGE 8. It can generate a 3D-display of the human cortex colored with regions grouped as similar. It can also produce a dendrogram. These visualizations can help answer questions like:

- Which regions have similar fingerprints?
- Do homologous (i.e., located symmetrically to the longitudinal fissure) regions share similar fingerprints or are they distinct?
- Do similar regions form spatial patterns similar to networks reported in the literature?

The user can turn off the figure display and decide on saving the figure to the hard drive for later examination. The figure is saved both in `.fig` and `.png` formats. Examples of figures that can be produced with `networksPlot.m` are shown in Fig. 4.31.

4.4.5 Module V: Maintenance

Maintenance routines (V) are used to automate some parts of the workflow, e.g., run all the stages with a single command, backup output data files, remove spare data and logs, etc. Most of these scripts were coded in Bash, having Linux users in mind. On Windows/MacOS platforms, one can try shell emulators like cygwin, cmder, etc., but their successful operation with provided scripts is not guaranteed. The following files are included in Maintenance (V) module:

- RUN_SELECTED.m,
- RUN_SELECTED_SLURM.sh,
- check_errors.sh,
- rm_logs_autosaves.sh,
- rm_output_data.sh.
- backup_output.sh,
- restore_from_backup.sh,

RUN_SELECTED.m is a short MATLAB script (Listing 2) that can be used to run selected STAGES in the desired order, assuming they are already configured (see Section 4.4.6). This script is used for serial computations. For parallel computations, RUN_SELECTED_SLURM.sh script should be used. To launch it, the user needs to put a subset of numbers from 1 to 8 into stages variable in the desired order, and then, in MATLAB's Command Window, the user needs to write:

```
>> RUN_SELECTED
```

and press Enter. The computations will start and the user should wait for the output saying "ALL DONE !".

Listing 2: Code of RUN_SELECTED.m script.

```
clear; close all; clc;

% stages to be ran
stages=[1 2 3 4 5 6 7 8];

%%
p = pwd();

for iStage = stages
  cd(['STAGE_', num2str(iStage), '/scripts/'])
  RUN

  % run integration for selected stages
  if(ismember(iStage, [2 4 6]))
    INTEGRATE
  end

  cd(p)
end

disp 'ALL DONE !'
```

Note that this script not only launches RUN.m scripts that are located in stages directories, but for stages 2, 4, and 6 it also launches corresponding INTEGRATE.m script to consolidate results of the given stage into single files.

RUN_SELECTED_SLURM.sh has similar purpose to RUN_SELECTED.m but in contrast it is to be used to run stages in parallel (listing was omitted for brevity; one can find in

ToFFi repository — see Appendix C). In current release, it is only possible on Linux machines with SLURM workload manager[18] installed.

To run this script (for example stages 3 7 6 in parallel in a given order), in Linux terminal the user should enter:

```
. RUN_SELECTED_SLURM 3 7 6
```

First, the number of input arguments passed to this command will be checked. If there are no arguments, the user will be warned, and the default order of stages will be assumed (i.e., $1, 2, 3, \ldots, 7, 8$). Before running, a question appears, which reminds the user that configuration need to be done before running. Finally, first stage from the list of arguments is queued using `sbatch` — a command predefined in SLURM. There are a couple of `if` statements that take care to queue additional `INTEGRATE.m` routines for selected stages (Stage 2, 4, and 6). Their role is to reorganize the output data into a more manageable set of files. After the first job is queued, the rest of them are scheduled too, but with building proper dependencies to guarantee that the next stage on the list will be executed only after previous stages are finished (`sbatch -dependency` command is used). This does not preclude the benefits of concurrent computing because usually, only part of single stage steps can compute in parallel due to the limit of possible MATLAB licenses assigned to the supercomputer user. Refactoring ToFFi Toolbox into non-commercial languages like Python could solve this issue, but then `RUN_SELECTED_SLURM.sh` script should be modified as well to run more stages at a time while securing their interdependence (Fig. 4.10).

When parallel computations are performed, by default, log files are created along the way. They help to catch errors in processing. Searching them manually would be laborious, thus `check_errors.sh` comes in handy (Listing 3). It searches in `.log` and `.out` files produced by SLURM workload manager for keywords such as "error", "undefined" and notes down file names and a line where a given phrase occurred inside a newly created file `errors.log`. To use this script, the user should run the following command in the terminal:

```
. check_errors.sh
```

Visual inspection of `errors.log` file speeds up debugging considerably.

To remove `.log` files and `.out` files, the user can use `rm_logs_autosaves.sh` script (Listing 4). To use this script, the user should run the following command in the terminal:

```
. rm_logs_autosaves.sh
```

[18] `https://slurm.schedmd.com/` (accessed: June 4, 2022)

Listing 3: Code of `check_errors.sh` script.

```
#!/bin/bash

# This script searches for lines containing "error" and "undefined"
# (case insensitive) word in *.log and *.out files and write them
# in errors.log file as an output.
#
# USAGE:
# 1. Run in shell: . check_errors.sh

echo 'Preparing error log file ...'
thisdir=`pwd`
error_log_file=$thisdir/errors.log
tmp_file=$thisdir/tmp_file_check_errors_routine
rm $error_log_file
touch $tmp_file

grep -Hni "error" *.log  >> $tmp_file
grep -Hni "error" *.out  >> $tmp_file
grep -Hni "slurmstepd: error:" *.out  >> $tmp_file
grep -Hni "undefined" *.log  >> $tmp_file
grep -Hni "undefined" *.out  >> $tmp_file

mv $tmp_file $error_log_file

echo DONE! Check ${error_log_file}
```

All the `.log`, `.out`, and `~` (autosave) files in the main repository directory will be removed.

After all desired computations are done, `STAGE_X/output/` folders are populated with several files. If for some reason, one would like to remove them quickly, one should consider using `rm_output_data.sh` script (Listing 5). To execute this script, the user should run the following command in the terminal:

```
. rm_output_data.sh
```

the user will be warned and asked for a confirmation before the data is removed irreversibly.

Instead of removing the data, the user can automatically make a backup of the output data of all the stages with `backup_output.sh` shell script (Listing 6). Additionally, it copies the `CONFIGURE.m` file, which contains the code to restore the toolbox configuration that led to backed up results, if needed. Inside `backup_output.sh` file, the user can set a `prefix` variable that will begin the name of the new directory that will be created for the stored data. `number` variable contains the current date to make it easier to distinguish different backups. Two additional suffixes can be added (`suffix1` and `suffix2` variable). `stages` variable should contain the indices of stages results of which the user would like to keep. Entering the command in the terminal:

```
. backup_output.sh
```

Listing 4: Code of `rm_logs_autosaves.sh` script.

```
#!/bin/sh
# This is script for
# - erasing all logs from parallel computations
# - erasing ~ files
#
# USAGE:
# 1. Run in shell: . rm_logs_autosaves.sh
echo "Called script for log files removal."
read -r -p "Are you sure? [y/N] " response
case "$response" in
[yY][eE][sS]|[yY])
echo "Erasing .log .out and ~files ..."
rm -v *.log
rm -v *.out
rm -v *~
echo "Done !"
;;
*)
echo "Ok. Aborted."
;;
esac
```

will begin the backup. A new directory with the name containing set prefixes and suffixes will appear, for example, `backup_2022-04-12_my_data`. Content of `STAGE_X/output/` directory (`X` denotes the stage number) and `CONFIGURE.m` file will be copied into that new directory.

`restore_from_backup.sh` do the reverse (Listing 7). It copies output data from the selected backup directory (provided as the command argument) and puts it inside the proper `STAGE_X/output/` directories. Here is an example of how to use this command:

```
. restore_from_backup.sh backup_2022-04-12_my_data
```

the user can also ask to restore the data from only selected stages. It is advised to empty `STAGE_X/output/` directories first, before running `restore_from_backup.sh` to prevent mixing data from different studies.

4.4.6 Configuration

Configuration is centralized. Initially, the user needed to jump around several files to configure the toolbox. This was obviously prone to errors and arduous. Instead, in the current revision of the toolbox, there is a single configuration file that adjusts the behavior of all the stages — `CONFIGURE.m`. This file is also copied to the backup directories when using `backup_output.sh` script.

Code enclosed in `CONFIGURE.m` contains parameters common to all the stages of Spectral Fingerprinting (II) and Analysis (III), as well as settings relevant to individual stages. When run, it creates `CFG.mat` files in all `STAGE_X/output/` directories. Each

Listing 5: Code of `rm_output_data.sh` script.

```
#!/bin/bash
# Script for
# - erasing content  from STAGE_*/output/ directory
#
# USAGE:
# 1. To clean selected stages output (e.g. output from stages 3 7 6)
#    run in shell: . rm_output_data.sh 3 7 6
# OR
# 1. Modify value of rm_out_stages variable below.
# 2. Run in shell without command line arguments: . rm_output_data.sh

# ### SETTINGS ######################################
declare -a rm_out_stages=(1 2 3 4 5 6 7 8) # all: rm_out_stages=(1 2 3 4 5 6 7 8)
# WARNING ! It will take effect if no command line arguments were passed !
###################################################

if [ $# -eq 0 ]
then
echo "WARNING: No argument supplied ! Cleaning stages defined inside rm_output_data.sh script ..."
else
rm_out_stages=( "$@" )
fi

echo "Declared stages to be cleared.  <-- PLEASE CHECK "
echo "${rm_out_stages[@]}"
echo "Root directory where stages directories reside:  <-- PLEASE CHECK "
echo `pwd`
read -r -p "Are you sure you want to remove output data? [y/N] " responseRM

case "$responseRM" in
[yY][eE][sS]|[yY])
echo "Removing output data  ..."

for i in `seq 0 $((${#rm_out_stages[@]}-1))`
do
iStage=${rm_out_stages[$i]}

echo "Removing data from STAGE_$iStage ..."
find STAGE_$iStage/output/. -exec rm -fr {} +

done

echo "OK. Done."
;;
*)
echo "Ok. Aborted. ...."
;;
esac
```

Listing 6: Code of `backup_output.sh` script.

```
#!/bin/bash
#
# Tool for selective backup.
# Each backup is identified by the date of creation and suffix (set by the user).
# You can also edit the prefix ('backup_' is default).
#
# NOTE ! Avoid whitespaces.
#
# USAGE:
# 1. Modify prefix, number, suffix1 and suffix2 variables (hints in comments)
# 2. Run in shell: . backup_output.sh

################################################################################

# CHOOSE STAGES TO BACKUP
declare -a stages=(1 2 3 4 5 6 7 8) # e.g. stages=(1 2 5 3)

prefix=backup_      # add _ at the end
number=`date +'%Y-%m-%d'`

# !!! START WITH UNDERSCORE _
suffix1=_my_data

# OPTIONAL; !!! START WITH UNDERSCORE _ IF suffix2 VARIABLE NOT EMPTY
suffix2=;

################################################################################

backupDir=$prefix$number$suffix1$suffix2

# adds header with current backup identifier
echo $prefix$number > output_list.log

# adds newlines
echo > output_list.log
echo > output_list.log

# finds all output files, no symlinks, not counting already backuped files
find `pwd -P` | grep /STAGE_ --color=always | grep output | grep -v backup >>
    output_list.log

mkdir $backupDir

# does backup to dir with correct prefix and number
for iStage in "${stages[@]}"
do
rsync -hvvaz --partial --progress --checksum STAGE_${iStage}/output ${backupDir}/
    STAGE_${iStage}
done

# copy configure so you can recreate results
rsync -hvvaz --progress --checksum CONFIGURE.m ${backupDir}/CONFIGURE.m
```

Listing 7: Code of `restore_from_backup.sh` script.

```
#!/bin/bash
#
# Tool for selective backup restoration.
# It copies output data from selected backup directory and puts inside output directories
# in each STAGE directory.
#
# NOTE: Configuration file stays in the backup directory (it is not copied).
#
# USAGE:
# Suppose you have backup directory called 'backup_2020-11-20'.
# 1. Modify value of stages variable below.
# 2. Run in shell: . restore_from_backup.sh backup_2020-11-20

# which stages to restore data from
declare -a stages=(1 2 3 4 5 6 7 8)

backupDir=$1

if [ $# -eq 0 ]
then
echo "WARNING: No argument (backup dir name) supplied ! Aborted."
exit
fi

restoreOutputFromBackup()
{
for iStage in "${stages[@]}"
do
echo "Restoring data from STAGE_$iStage ..."
rsync -hvvaz --partial --progress  ${backupDir}/STAGE_${iStage}/output STAGE_${iStage}/.
done

echo 'ALL DONE !'
}

restoreOutputFromBackup
```

of the files contains `CFG` structure, different for each stage, as it contains stage-specific parameters. For example, STAGE 1 configuration structure might look like this (the output was truncated for brevity):

```
>> load STAGE_1/output/CFG.mat
>> CFG
CFG =
struct with fields:
Global: [1×1 struct]
STAGE_1: [1×1 struct]
>> CFG.Global
ans =
struct with fields:
TAG: 'illustrative_example_tryton'
VERSION: 'develop_2e056a5'
DATE: '2021-04-01-14:16'
commonDataDir: './commonData/'
goodSubjects: [1 2 3 4 5 6 7 8 9 10]
goodROI: [6 25 54 67 70 89 102 105]
>> CFG.STAGE_1
ans =
struct with fields:
MODE: 'serial'
rngSeed: 2021
normalizationMethod: 'wholebrain'
dummySignalMode: 'no'
frequenciesOfInterest: [1 1.2143 1.4745 1.7904 ... 27.1283 32.9413 40.0000]
```

and STAGE 8 configuration might look like this (the output was truncated for brevity):

```
>> load STAGE_8/output/CFG.mat
>> CFG.Global
ans =
struct with fields:
TAG: 'illustrative_example_tryton'
VERSION: 'develop_2e056a5'
DATE: '2021-04-01-14:16'
commonDataDir: './commonData/'
goodSubjects: [1 2 3 4 5 6 7 8 9 10]
goodROI: [6 25 54 67 70 89 102 105]
>> CFG.STAGE_8
ans =
struct with fields:
MODE: 'serial'
linkageMethod: 'average'
linkageDistanceMetric: 'cosine'
```

Note that both structures share the same `Global` field. Having these files precisely located, the toolbox loads them whenever each stage is launched and refers to them to adjust the processing parameters.

`CONFIGURE.m` file is divided into sections related to global parameters and individual stages. Code comments help to navigate the user and also point to the blocks of code that shall not be edited.

In the current revision of the toolbox, `CONFIGURE.m` file does **not** cover the configuration of the parameters for parallel processing. These parameters are contained within `.sl` files that are the interface scripts between the user and SLURM workload manager. To alleviate this problem, there is another configuration method instead of modifying and running `CONFIGURE.m` directly. There is a file named `CFG_illustrative.sh`, which can serve as a template to be copied by the user and adjusted to configure all the parameters, including those regarding SLURM workload manager. The user can have many `CFG_*.sh` files (star denotes arbitrary name suffix) but only one `CONFIGURE.m` file. Each `CFG_*.sh` can represent a different processing toolbox configuration the user wants to perform.

After running `CFG_*.sh`, there is no need to additionally configure and run CONFIGURE.m afterward. Actually, `.sh` script modifies the content `CONFIGURE.m` using several `sed` instructions, then opens MATLAB in the command line, and finally runs `CONFIGURE.m`.

It needs to be noted that `CFG_*.sh` is a script written in Bash, which unintentionally favors Linux users. Windows user may try terminal emulators like `cygwin`[19] or `cmder`[20], but their correct operation with the script is not guaranteed.

The structure of `CFG_illustrative.sh` file is similar to `CONFIGURE.m` file. There is an extensive description in the toolbox documentation of how to modify (section 7.4.2. therein) and run (section 8.5. therein) both configuration files (link to the documentation provided in Appendix C). Also, the instructions on the configuration in the illustrative example are worthwhile to read (section 4.5. therein; link to the documentation provided in Appendix C).

To sum up, Windows user should open `CONFIGURE.m` in MATLAB, modify it, and run it in the Command Window:

```
>> CONFIGURE
```

If the user wants to perform parallel computations, additional modification of `.sl` files present in the main repository directory is needed. For Linux users, it is enough

[19] `https://www.cygwin.com/` (accessed: June 4, 2022)
[20] `https://cmder.net/` (accessed: June 4, 2022)

to modify `CFG_*.sh` file and run it in the terminal (example for the file named `CFG_illustrative.sh`):

```
. CFG_illustrative.sh
```

4.5 Typical workflow

The input data should comply with the details described in the toolbox documentation (chapter 5 therein; link to the documentation provided in Appendix C). Chapter 8 in the documentation provides details about how to use the toolbox in different possible scenarios:

- serial computations (under Windows/MacOS),
- parallel computations involving one computer (under Windows/MacOS),
- serial computations (under Linux),
- parallel computations involving one computer (under Linux),
- parallel computations involving multiple computers (under Linux).

Here is a brief description of the typical workflow:

1. Download the toolbox and get the input data,
2. Prepare the data using `DATA_PREPARATION` (I) scripts,
3. Configure the toolbox (modifying and running `CONFIGURE.m` or `CFG_*.sh`),
4. Remove old data and logs from previous computations if needed (manually or using `rm_output_data.sh` and `rm_logs_autosaves.sh` scripts),
5. Run selected stages (using `RUN_SELECTED.m` or `RUN_SELECTED_SLURM.sh` script),
6. When finished, consult logs for errors (if logs were created),
7. Refine configuration in selected stages if needed (again modifying and running `CONFIGURE.m` or `CFG_*.sh`),
8. Delete old data from the selected stages (manually or using `rm_output_data.sh`),
9. Run the selected stages again,
10. Consult the logs for errors again,
11. If satisfied, make a backup of the output data (manually or using `backup_output.sh` script),
12. Visualize and analyze the results (e.g., using scripts from `PRESENTATION` (IV) module).

Note that stages cannot be run in utterly arbitrary order, as some of the output of a given stage is the input to another stage. Diagram presented in Fig. 4.10 illustrates this.

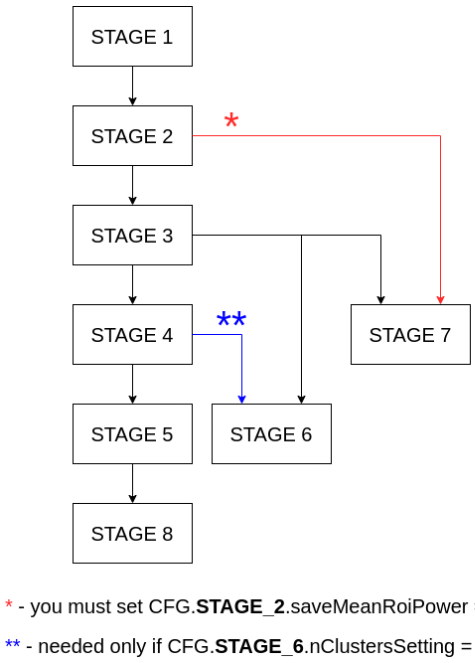


Fig. 4.10. Stages interdependence diagram. An arrow means that output from stage written in a parent node is needed to run stage written in a child node. Stages in the same row can run concurrently.

Presented interdependence needs to be considered especially when planning to run parallel computations to prevent situations where the user demands running a particular stage without having computed its input data yet. Below, there are examples of correct running order lists:

- 1, 2+I, 3, 4+I, 5, 6+I, 7, 8;
- 1, 2+I, 3, 7, 4+I, 6+I, 5, 8;
- 1, 2+I, 3, 4+I, 5, 8, 7, 6+I;
- 1, 2+I, 3, 4+I, 5, 8, 6+I, 7.

Letter "I" denote "Integration", which signifies the necessity of running not only `RUN.m` script for a particular stage but also to follow it by launching the corresponding `INTEGRATE.m` script. Of course, the user need not remember this, as `RUN_SELECTED.m` and `RUN_SELECTED_SLURM.sh` contain control sequences assuring that proper integration script is executed after running stages 2, 4, and 6.

4.6 Time complexity analysis

In this section, the notion of algorithm time complexity will be explained. Spectral Fingerprinting is composed of many steps, and in real cases, it may take a long time to complete, sometimes even days. It is therefore important to evaluate its complexity. The complexity of some stages is difficult to evaluate from a theoretical point of view

due to the stochastic nature of used algorithms, so empirical tests were run to estimate time dependence on various parameters. Each stage of the Spectral Fingerprinting module will be discussed from the running time perspective. It could help the user predict how the running time is affected by a given change of toolbox parameters.

Time complexity definition originated from a set of mathematical methods collectively known as the *asymptotic analysis* [221]. Time complexity is a measure of an upper bound of time a given algorithm will compute, i.e., *worst-case* running time. If one denotes the algorithm running time as a function f of the algorithm input size $N \in \mathcal{N}$, we say that:

$$f(N) = \mathcal{O}(g(N)) \iff \exists_{k>0}\exists_{N_0}\forall_{N>N_0}|f(N)| \leq k \cdot |g(N)| \tag{4.1}$$

where $f(N)$ is the estimated running time, $g(N)$ is called the *comparison function*. For example, $f(N) = 3N^2 - 1.3N + 100$ is said to be *of the order of* $\mathcal{O}(N^2)$ because one can adjust the value of k in such a way that starting from input size $N > N_0$ function f is no larger than $g(N) = kN^2$. Conversely, $f(N)$ is not $\mathcal{O}(N)$ as there always exists such N_0 for which $f(N \geq N_0)$ "exceeds" $g(N) = kN$, no matter how big k is chosen.

Already, some useful properties of the big-$\mathcal{O}$ notation can be recognized. Below is the list of selected properties used in the analyses conducted in the next paragraphs. For readability, $f(N)$ is simply denoted as f, and $g(N)$ as g:

1) For constant $k > 0$: $f = \mathcal{O}(g) \implies kf = \mathcal{O}(g)$,

2) $f_1 = \mathcal{O}(g)$, $f_2 = \mathcal{O}(g) \implies f_1 + f_2 = \mathcal{O}(g)$

3) $f_1 = \mathcal{O}(g_2)$, $f_2 = \mathcal{O}(g_2) \implies f_1 f_2 = \mathcal{O}(g_1 g_2)$

4) $f_1 + f_2 = \mathcal{O}(\max\{g_1, g_2\})$

Property 4) allows to omit lower-order terms and estimate f complexity with the highest order term, e.g., $f(N) = 4N^3 - N^2 + 5N + 9 = \mathcal{O}(4N^3)$, which becomes $\mathcal{O}(N^3)$ thanks to property 1).

The interested reader can develop some intuition about complexity by referring to [211]. More formally, complexity is discussed in [221]. Big-$\mathcal{O}$ notation often appears in books describing algorithms, such as work by Cormen et al. [222], and elaborations regarding numerical methods — "Numerical Recipes" by Press et al., is a good reference [223].

Alas, there is often no closed-form expression for $f(N)$, and the researcher needs to estimate the time complexity of an algorithm. In the following paragraphs, the

estimation of the time complexity of each stage of Spectral Fingerprinting was given, based on their simplified flowcharts [224]. As was previously mentioned, each stage can be configured using a multitude of input parameters. Two of them are usually of the utmost interest to the researcher:

- the number of subjects (denoted as S),
- the number of brain regions of interest, ROI (K).

It is common in neuroscience that a given analysis pipeline repeats the same set of procedures for each subject or for each brain area, assuming their independence. Flowcharts presented below (Fig. 4.11—4.25) comply with this scheme. During the toolbox development, I discovered that for some stages number of repeats/iterations is heavily affecting the running time. For this reason, Spectral Fingerprinting flowcharts also expose the following parameters:

- the number of cross-validation repeats (R),
- the number of folds of data (F),
- the number of iterations of cluster number evaluation (I).

Both R and F are used in cross-validation in stages 6 and 7. Parameter I is used in stages 2, 4, and 7. Wherever it was possible, loops depending on these parameters were subjected to concurrent computing (see Table 4.2 and Section 4.7).

It is worth noting that for different studies (i.e., involving a different number of participants and/or number of areas) R, F, and I can be set to the same appropriate large enough common values. During the following tests this guideline was obeyed. This means that effectively only S and K parameters differed and thus affected the running time. In other words, R, F, I were assumed constant, and S and K were the input sizes important for the complexity analysis.

The following computations were performed on a single machine, equipped with 24 CPU cores, instead of using multiple nodes to simplify the analysis and exclude the possible time overhead that may result from queuing jobs. Because of the limited time to perform computations, to check estimated complexity empirically, the number of subjects and the number of regions were limited to values $S, K = 4, 6, 12, 24$. In the accompanying flowcharts, operations insignificant to the running time or not consisting of loops with the parameters mentioned above were hidden inside so-called *preparation outlines* and *predefined process outlines*, depicted in [224], for better readability and easier analysis.

In the next paragraphs, time complexity using big-$\mathcal{O}$ notation was estimated with the assumptions stated above. Table 4.1 summarizes the theoretical complexities estimated for each of the stages. It was then empirically tested, by measuring the running time (so-called "CPU-time", instead of the so-called "wall-clock time") of each stage for a range of values of S and K parameters. To estimate standard error, for each (S, K) pair, computations were repeated five times. Curve fitting was performed to check whether theoretical time complexity agrees with empirical data.

Table 4.1. Stages theoretical complexity estimates based on flowcharts presented in Fig. 4.11—4.25. S stands for the number of subjects, K denotes the number of brain areas.

Time complexity								
Stage	1	2	3	4	5	6	7	8
$f(S,K)$	$\mathcal{O}(S)$	$\mathcal{O}(SK)$	$\mathcal{O}(SK)$	$\mathcal{O}(K)$	$\mathcal{O}(K)$	$\mathcal{O}(SK^2)$	$\mathcal{O}(SK^2)$	$\mathcal{O}(K^2)$

Time complexity of STAGE 1

Fig. 4.11 shows the simplified flowchart for STAGE 1 algorithms, exposing operations relevant for time complexity estimation. From this point of view, running time depends solely on the number of subjects S, thus time complexity for this stage can be simply stated as $f(S, K) = \mathcal{O}(S)$.

Empirical running times were depicted in Fig. 4.12. Despite high variability of the repetitions for each (S, K) pair, Fig. 4.12a suggests that STAGE 1 running time is linearly dependent on the number of subjects S. At the same time, Fig. 4.12b reveals much larger standard errors of prediction and smaller slopes of fitted curves, suggesting running time to be invariant to the number of brain areas K. Both observations are in line with the hypothesized time complexity $f(S, K) = \mathcal{O}(S)$.

Time complexity of STAGE 2

Fig. 4.13 shows the simplified flowchart for STAGE 2 algorithms, exposing operations relevant for time complexity estimation. The input size is given by: the number of subjects S, the number of brain areas K, and the number of iterations of cluster evaluation I. From this point of view, running time depends on parameters S, K, and I, thus time complexity for this stage can be at first approximated as $\mathcal{O}(SKI)$. However, as was previously stated, one can treat I as a constant, thus expression for time complexity can be simplified to $f(S, K) = \mathcal{O}(SK)$.

Empirically, Fig. 4.14 provides strong evidence for a linear dependence of the running time on both the number of subjects S and the number of areas K. Repetitions

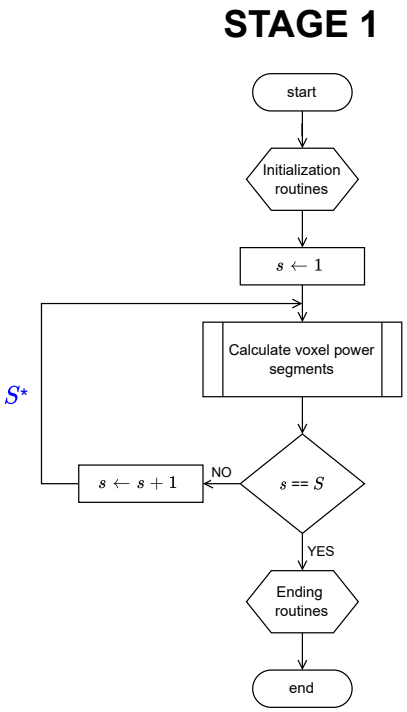


Fig. 4.11. Flowchart depicting STAGE 1 of the ToFFi's Spectral Fingerprinting module (II). Notation: $s = 1, 2, \ldots, S$ is the subject index. Blue letters label loops' maximal number of iterations. Starred * letters refer to loops adapted for concurrency.

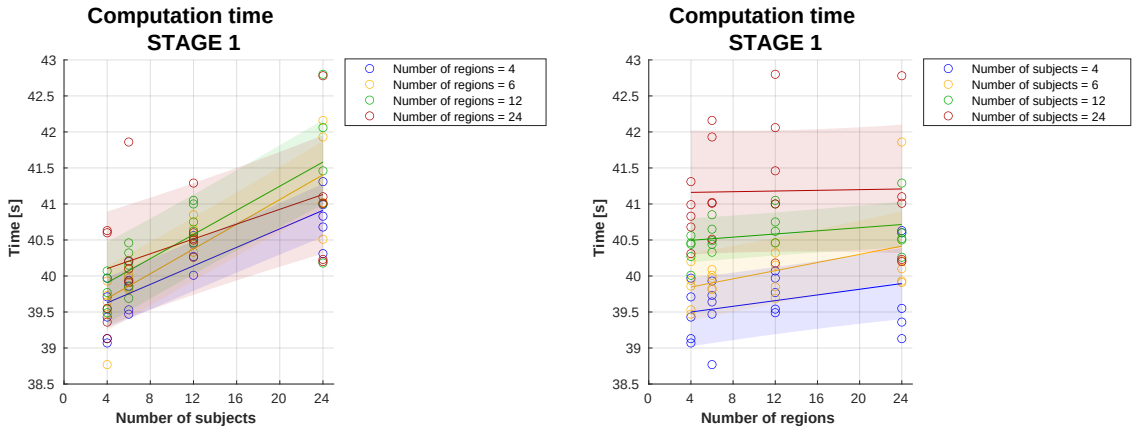


Fig. 4.12. STAGE 1 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). Each point represents one of five total repetitions for a given (S, K) pair. To assess whether data agrees with the estimated time complexity ($\mathcal{O}(S)$), linear function $y = ax + b$ was fitted. Its color-related shade represents standard error for prediction. See text for interpretation.

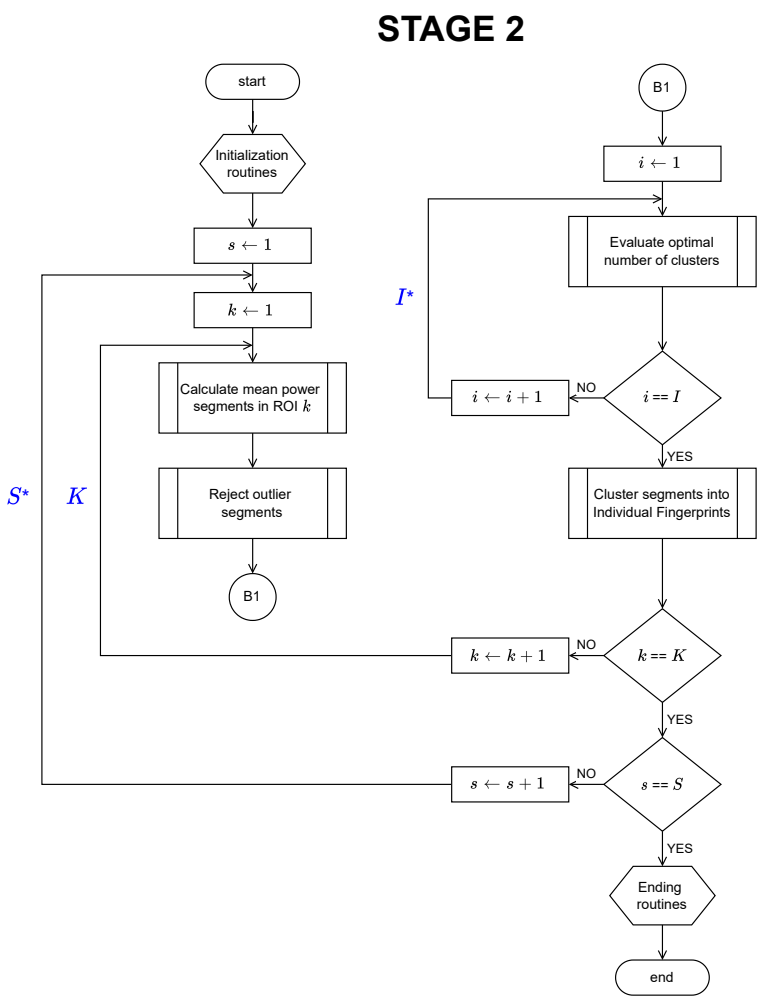


Fig. 4.13. Flowchart depicting STAGE 2 of the ToFFi's Spectral Fingerprinting module (II). Notation: $s = 1, 2, \ldots, S$ is the subject index, $k = 1, 2, \ldots, K$ is the brain area index, $i = 1, 2, \ldots, I$ is the index of cluster evaluation iteration. Blue letters label loops' maximal number of iterations. Starred * letters refer to loops adapted for concurrency.

display small variance compared to the order of magnitude of analyzed time. Slopes of the fitted curves are significantly larger than 0, indicating clearly the linear increase of the dependent variable with the increase of the studied parameters. Subsequent fitted curves appear one above another in the expected order, aligned with the increase of the corresponding independent variable. For this stage, hypothesized time complexity $f(S, K) = \mathcal{O}(SK)$ can be confirmed with confidence.

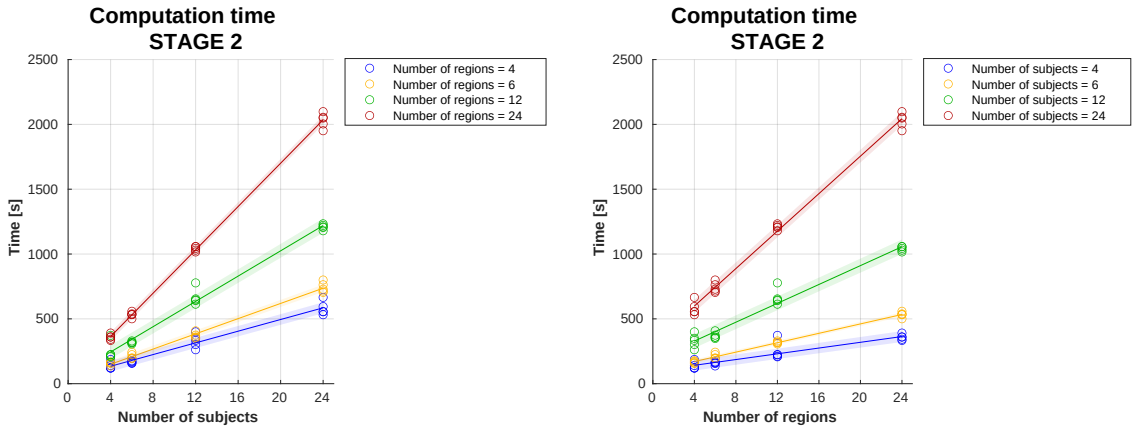


Fig. 4.14. STAGE 2 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). Each point represents one of five total repetitions for a given (S, K) pair. To assess whether data agrees with the estimated time complexity ($\mathcal{O}(SK)$), linear function $y = ax + b$ was fitted. Its color-related shade represents standard error for prediction. See text for interpretation.

Time complexity of STAGE 3

Fig. 4.15 shows the simplified flowchart for STAGE 3 algorithms, exposing operations relevant for time complexity estimation. The input size is given by the number of subjects S and the number of brain areas K. From this point of view, running time depends on parameters S and K, thus time complexity for this stage can be at first approximated as $\mathcal{O}(S + SK)$. By recognizing SK as the dominant term, the expression for time complexity can be simplified to $f(S, K) = \mathcal{O}(SK)$.

Fig. 4.16 shows STAGE 3 running time analysis. Data for $S, K = 4, 6, 12$ exhibit relatively large variance that prevents making a strong statement rejecting or not rejecting the $f(S, K) = \mathcal{O}(SK)$ time complexity hypothesis. Only the data for $S, K = 24$ may indicate its verity. Related points are characterized by low enough variance, and corresponding linear fitting curves have considerable steepness. For this stage, results seem to be inconclusive.

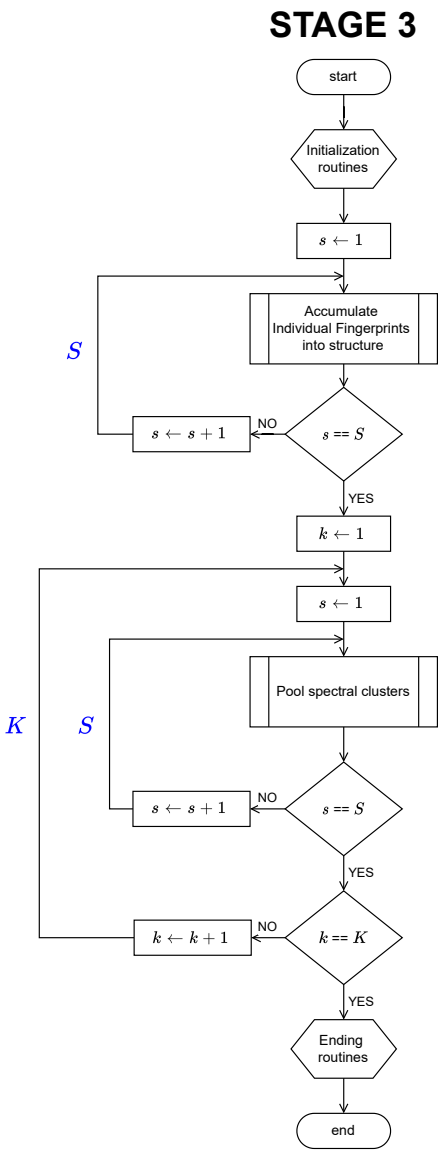


Fig. 4.15. Flowchart depicting STAGE 3 of the ToFFi's Spectral Fingerprinting module (II). Notation: $s = 1, 2, \ldots, S$ is the subject index, $k = 1, 2, \ldots, K$ is the brain area index. Blue letters label loops' maximal number of iterations.

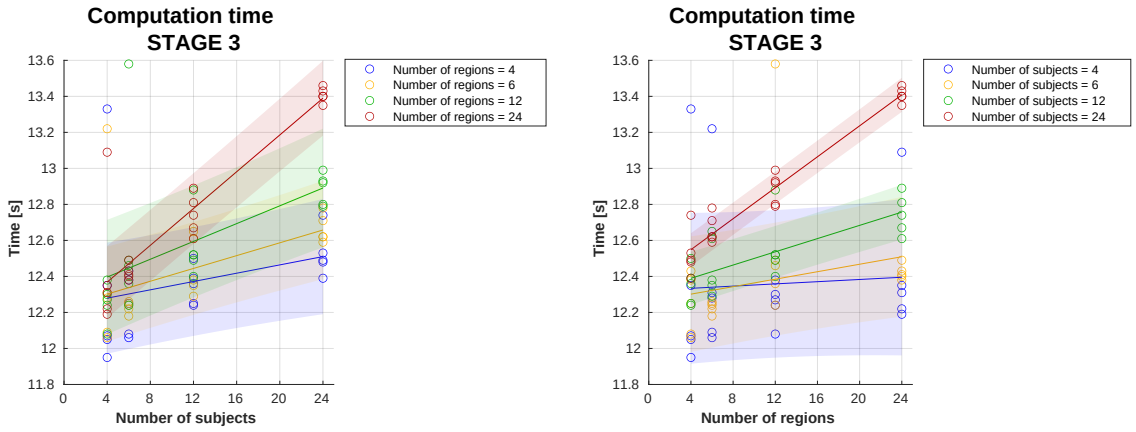


Fig. 4.16. STAGE 3 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). Each point represents one of five total repetitions for a given (S, K) pair. To assess whether data agrees with the estimated time complexity ($\mathcal{O}(SK)$), linear function $y = ax + b$ was fitted. Its color-related shade represents standard error for prediction. See text for interpretation.

Time complexity of STAGE 4

Fig. 4.17 shows the simplified flowchart for STAGE 4 algorithms, exposing operations relevant for time complexity estimation. The input size is given by the number of brain areas K and the number of iterations of cluster evaluation I. From this point of view, running time depends on parameters K and I, thus time complexity for this stage can be at first approximated as $\mathcal{O}(KI)$. However, as was previously stated, one can treat I as a constant, so the expression for time complexity can be simplified to $f(S, K) = \mathcal{O}(K)$.

Fig. 4.18 shows STAGE 4 running time analysis. The plot presented in 4.18b seems convincing to accept $\mathcal{O}(K)$ as the proper time complexity estimate for the stage. The reason for such a dramatic difference between the curve for $S = 24$ and the rest of the fitting curves remained unknown. Unexpectedly, the plot presented in 4.18a suggests the dependence of the running time also on the number of subjects, even though data from different participants were collapsed during STAGE 3. The question of which of the suggested time complexity estimates is valid — ($\mathcal{O}(K)$ or rather $\mathcal{O}(SK)$) remained unanswered. Tests with a wider range of S, K parameters as well as with more repetitions should lead to a satisfying conclusion.

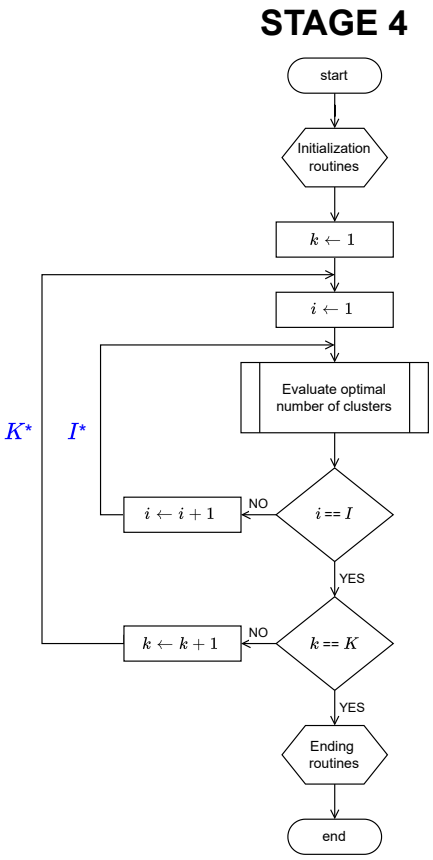


Fig. 4.17. Flowchart depicting STAGE 4 of the ToFFi's Spectral Fingerprinting module (II). Notation: $k = 1, 2, \ldots, K$ is the brain area index, $i = 1, 2, \ldots, I$ is the index of cluster evaluation iteration. Blue letters label loops' maximal number of iterations. Starred * letters refer to loops adapted for concurrency.

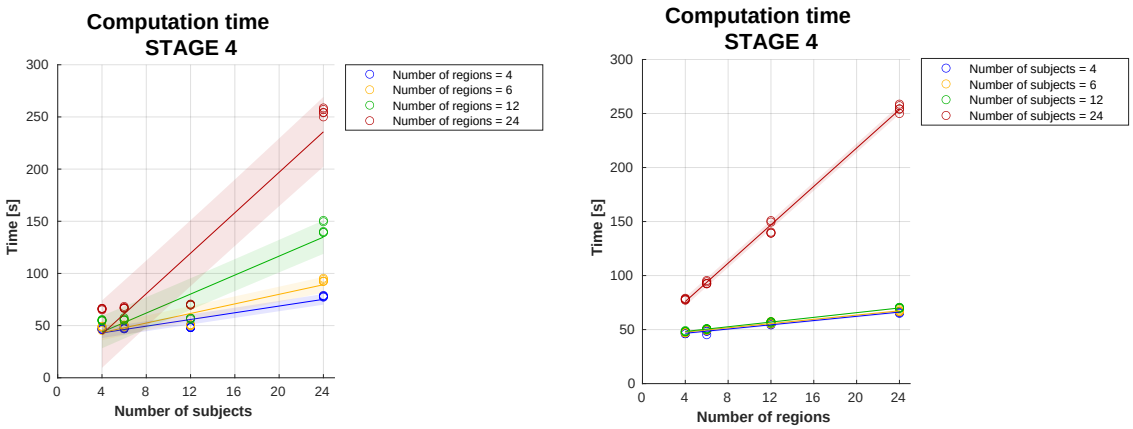


Fig. 4.18. STAGE 4 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). Each point represents one of five total repetitions for a given (S, K) pair. To assess whether data agrees with the estimated time complexity ($\mathcal{O}(K)$), linear function $y = ax + b$ was fitted. Its color-related shade (barely visible in the right plot) represents standard error for prediction. See text for interpretation.

Time complexity of STAGE 5

Fig. 4.19 shows the simplified flowchart for STAGE 5 algorithms, exposing operations relevant for time complexity estimation. The input size is given by the number of brain areas K. From this point of view, running time depends solely on the number of areas K, thus time complexity for this stage can be simply stated as $f(S, K) = \mathcal{O}(K)$.

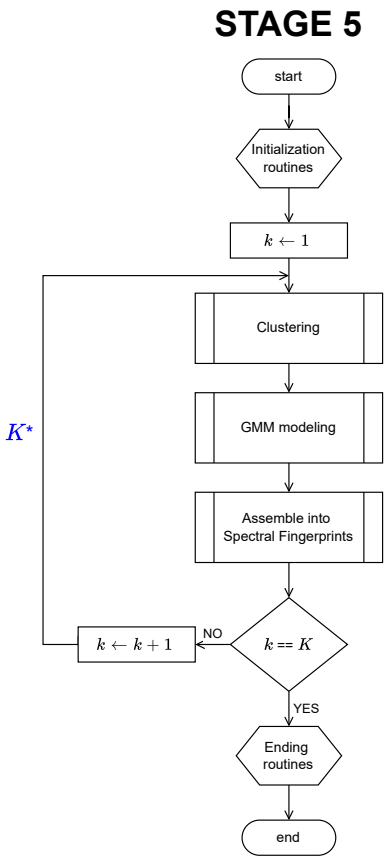


Fig. 4.19. Flowchart depicting STAGE 5 of the ToFFi's Spectral Fingerprinting module (II). Notation: $k = 1, 2, \ldots, K$ is the brain area index. Blue letters label loops' maximal number of iterations. Starred letters refer to loops adapted for concurrency.

Unfortunately, due to the large variance, the data presented in Fig. 4.20 cannot be an argument to support the verity of theoretically estimated time complexity $\mathcal{O}(K)$ for this stage. Again, tests with a wider range of S, K parameters as well as with more repetitions should deliver more decisive observations.

Time complexity of STAGE 6

Fig. 4.21 shows the simplified flowchart for STAGE 6 algorithms, exposing operations relevant for time complexity estimation. The input size is given by: the number of

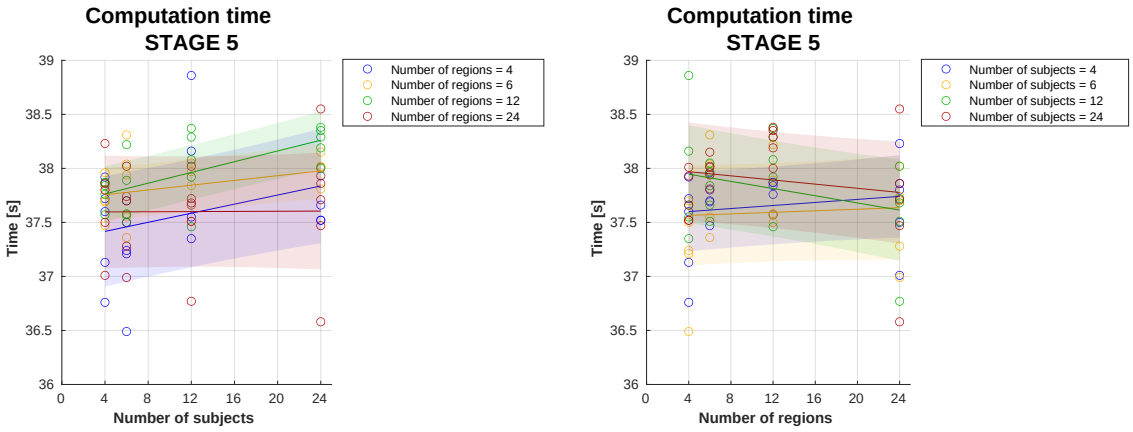


Fig. 4.20. STAGE 5 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). Each point represents one of five total repetitions for a given (S, K) pair. To assess whether data agrees with the estimated time complexity ($\mathcal{O}(K)$), linear function $y = ax + b$ was fitted. Its color-related shade represents standard error for prediction. See text for interpretation.

cross-validation repetitions R, the number of data folds F, the number of brain areas K, and the number of subjects S. From this point of view, running time depends on parameters R, F, K, and S. Analyzing the loops structure, the time complexity for this stage can be at first approximated as $\mathcal{O}(RF(SK + SKK))$. As was previously stated, one can treat R and F as constants, thus expression for time complexity can be simplified to $\mathcal{O}(SK + SK^2)$. Finally, by identifying the dominant term, one can write $f(S, K) = \mathcal{O}(SK^2)$.

Fig. 4.22 shows STAGE 6 running time analysis. Unfortunately, due to the relatively large variance of the data points, both linear and quadratic relationships present in the estimated time complexity, $f(S, K) = \mathcal{O}(SK^2)$ equation is not supported enough. More repetitions and a wider ranges for S and K should give better data to assess STAGE 6 time complexity.

Time complexity of STAGE 7

Fig. 4.23 shows the simplified flowchart for STAGE 7 algorithms, exposing operations relevant for time complexity estimation. The input size is given by: the number of subjects S, the number of cross-validation repetitions R, the number of data folds F, the number of brain areas K, and the number of iterations of cluster evaluation I. From this point of view, running time depends on parameters R, F, I, K, and S. Analyzing loop structure, the time complexity for this stage can be at first approximated as $\mathcal{O}(SRF(IK + KK))$. As was previously stated, one can treat R, F, and I as constants,

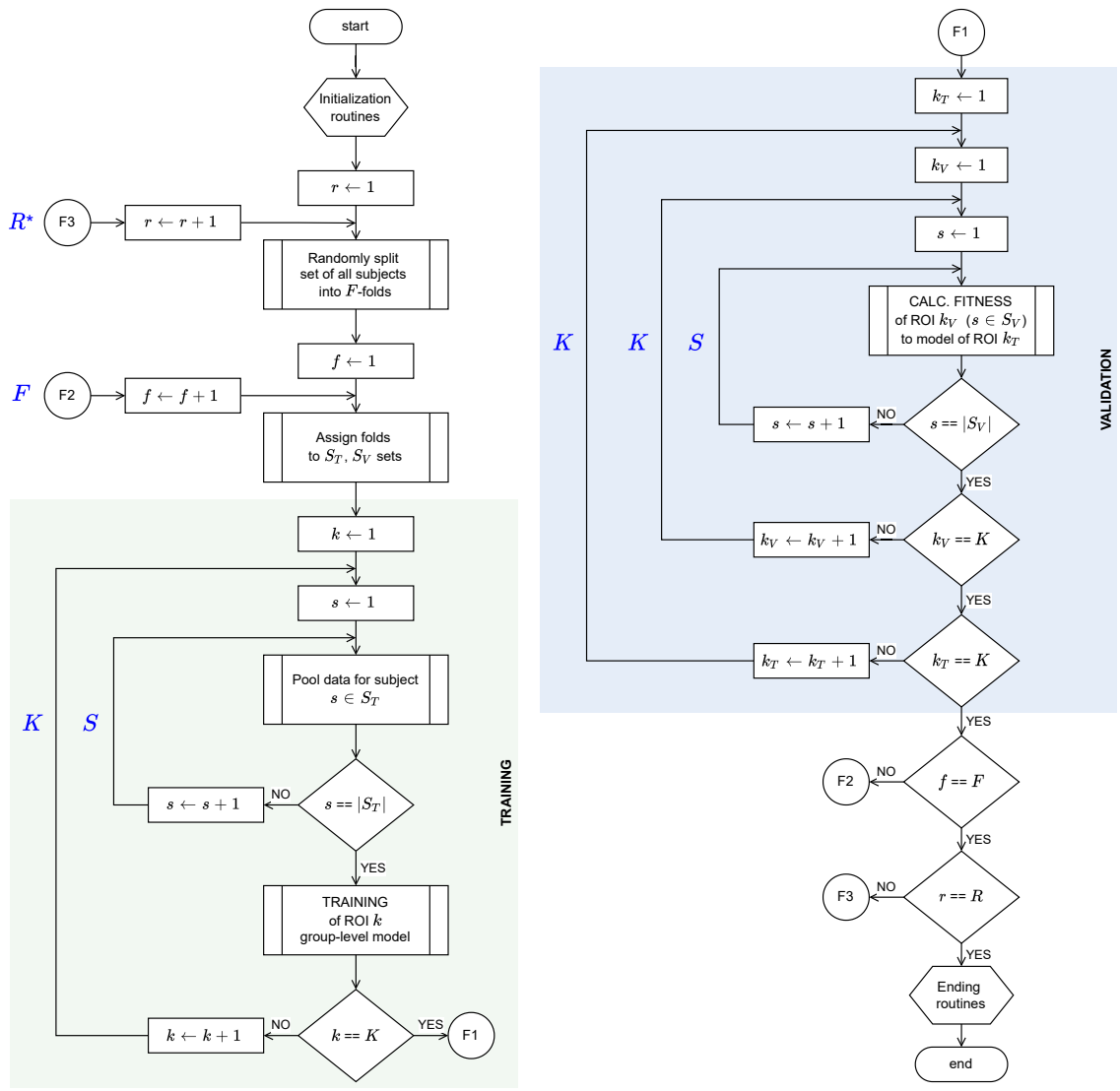


Fig. 4.21. Flowchart depicting STAGE 6 of the ToFFi's Spectral Fingerprinting module (II). Notation: $r = 1, 2, \ldots, R$ is the cross-validation repetition index, $f = 1, 2, \ldots, F$ is the data fold index, $k = 1, 2, \ldots, K$ is the brain area index, $s = 1, 2, \ldots, S$ is the subject index. Blue letters label loops' maximal number of iterations. Starred * letters refer to loops adapted for concurrency.

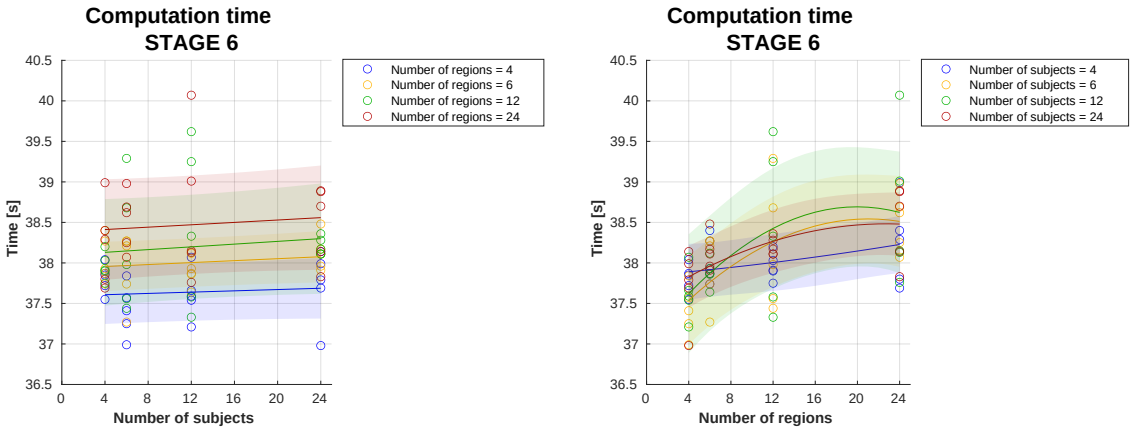


Fig. 4.22. STAGE 6 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). To assess predicted linear dependence of time on S present in the estimated time complexity $\mathcal{O}(SK^2)$, linear function $y = ax + b$ was fitted. **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). To assess predicted quadratic dependence of time on K, quadratic function $y = ax^2 + bx + c$ was fitted. Each point represents one of five total repetitions for a given (S, K) pair. Shades represent standard error for prediction. See text for interpretation.

thus expression for time complexity can be simplified to $\mathcal{O}(SK + SK^2))$. Finally, by identifying the dominant term, one can write $f(S, K) = \mathcal{O}(SK^2)$.

The predicted time complexity for STAGE 7 $f(S, K) = \mathcal{O}(SK^2)$ was tested empirically. Fig. 4.24 provides strong evidence for a linear dependency of the running time on the number of subjects S and number of areas K, despite the expected quadratic dependency on the number of areas. Repetitions display small variance compared to the order of magnitude of analyzed time. Slopes of the fitted curves are significantly larger than 0, indicating clearly the linear increase of the dependent variable with the increase of the studied parameters. Subsequent fitted curves appear one above another in the expected order, aligned with the increase of the corresponding independent variable. Testing further for more areas could suggest an answer to the question of whether the time complexity for this stage is $f(S, K) = \mathcal{O}(SK^2)$ or rather $f(S, K) = \mathcal{O}(SK)$. The second case would be in opposition to the stage structure presented in the flowchart.

Time complexity of STAGE 8

Fig. 4.25 shows the simplified flowchart for STAGE 8 algorithms, exposing operations relevant for time complexity estimation. The input size is given by the number of brain areas K. From this point of view, running time depends solely on the number of areas K, thus time complexity for this stage can be simply stated as $f(S, K) = \mathcal{O}(K^2)$.

Data presented in Fig. 4.26 seems to support the verity of predicted time complexity

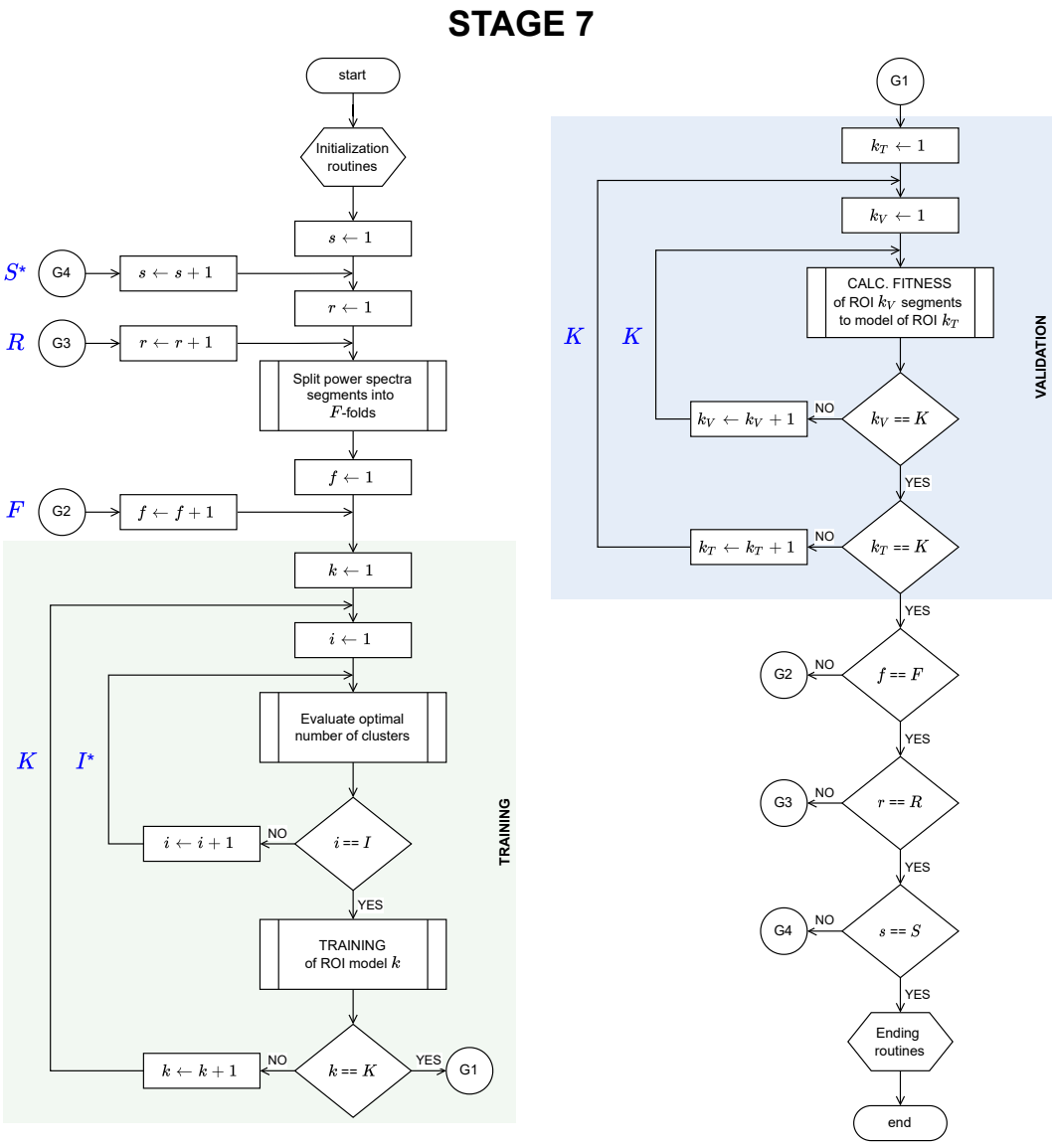


Fig. 4.23. Flowchart depicting STAGE 7 of the ToFFi's Spectral Fingerprinting module (II). Notation: $s = 1, 2, \ldots, S$ is the subject index, $r = 1, 2, \ldots, R$ is the cross-validation repetition index, $f = 1, 2, \ldots, F$ is the data fold index, $k = 1, 2, \ldots, K$ is the brain area index, $i = 1, 2, \ldots, I$ is the index of cluster evaluation iteration. Blue letters label loops' maximal number of iterations. Starred * letters refer to loops adapted for concurrency.

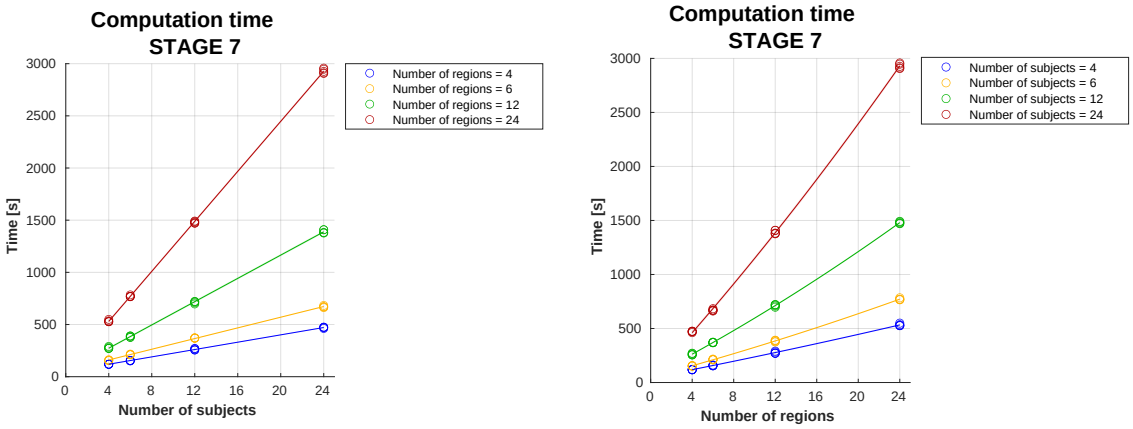


Fig. 4.24. STAGE 7 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). To assess predicted linear dependence of time on S present in the estimated time complexity $\mathcal{O}(SK^2)$, linear function $y = ax + b$ was fitted. **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). To assess predicted quadratic dependence of time on K, quadratic function $y = ax^2 + bx + c$ was fitted. Each point represents one of five total repetitions for a given (S, K) pair. Shades (barely visible) represent standard error for prediction. See text for interpretation.

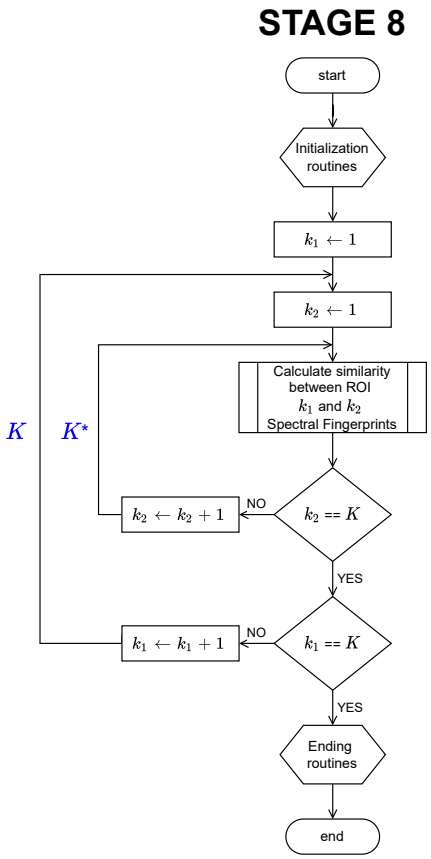


Fig. 4.25. Flowchart depicting STAGE 8 of the ToFFi's Spectral Fingerprinting module (II). Notation: $k = 1, 2, \ldots, K$ is the brain area index. Blue letters label loops' maximal number of iterations. Starred letters refer to loops adapted for concurrency.

$f(S, K) = \mathcal{O}(K^2)$. The increase in the number of subjects does not lead to a quadratic or even linear increase in the computation time, as clearly seen in Fig. 4.26a. Thus one can confidently say that for the presented range of input parameters, STAGE 8 computation time is independent of the number of subjects, as expected. Successfully, Fig. 4.26b suggests quadratic dependence on the number of regions. Low data variance for both plots allows for some degree of certainty in stated claims, however, a test conducted for the larger number of subjects and brain areas would be decisive.

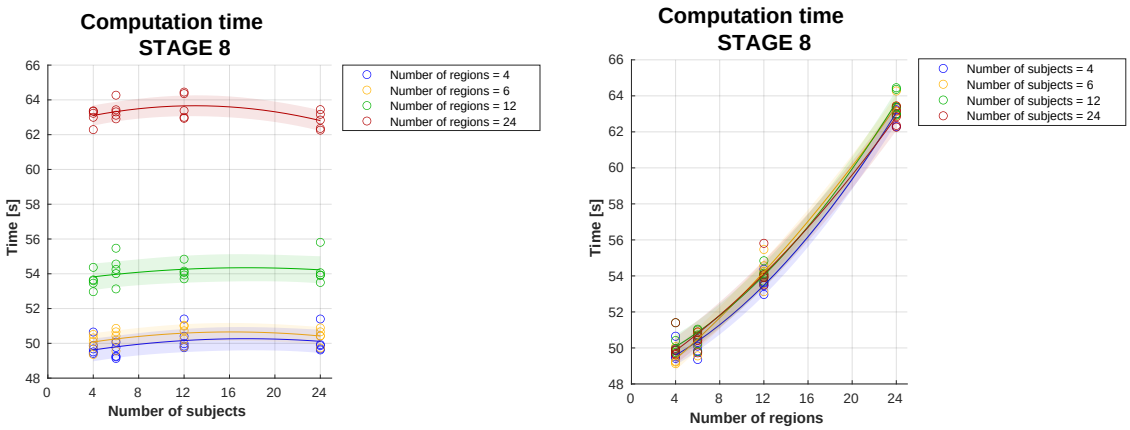


Fig. 4.26. STAGE 8 running time analysis. **a)** Computation time in seconds presented as a function of the number of subjects ($S = 4, 6, 12, 24$). **b)** Computation time in seconds presented as a function of the number of regions ($K = 4, 6, 12, 24$). Each point represents one of five total repetitions for a given (S, K) pair. To assess whether data agrees with the estimated time complexity ($\mathcal{O}(K^2)$), quadratic function $y = ax^2 + bx + c$ was fitted. Its color-related shade represents standard error for prediction. See text for interpretation.

Summary of the time complexity analyses

As the reader may have noticed, none of the statements regarding theoretically estimated time complexities were definitive. This reserved tone has arisen from the fact that due to the time constraint, too little data was collected to make stronger claims. Increasing the number of repetitions to dozens or hundreds would provide much tighter prediction error estimates. Another shortcoming was a relatively small range of parameters covered that may not be enough to see the dominance of higher-order terms in time complexity polynomials. Discrepancies between theoretical and empirical complexity might be related to the lower-order terms' influence. To alleviate this problem, purely serial operation (i.e., computations done on a single machine and single CPU core) of the data for S, K in the order of hundreds or thousands may be appropriate, however, practically nearly impossible to achieve — there are no currently available consistent MEG databases having that many subjects recorded. The number of areas may not be

a problem at the same time, as one may treat single brain voxels as separate regions of interest, for the sake of the experiment. Finally, appropriate statistical testing would be needed in order to gain confidence in drawing conclusions about the time complexity.

In practice, flowcharts shown in Fig. 4.11—4.25 as well as Tables 4.1 and 4.2 provide enough assistance to the ToFFi Toolbox users in predicting how much time a planned experiment might consume.

4.7 Concurrency

Research for initial results of Spectral Fingerprinting reproduction and replication [12] exposed one major problem with the method — it was relatively slow. To calculate fingerprints for a single data set and analyze them, it took about 5 days to complete all necessary computations, which was not acceptable. Decision on performing code refactorization to allow concurrent computing was made. MATLAB offers various solutions to aid parallel processing of the data[21], and choosing the CI-TASK supercomputer infrastructure implied the usage of SLURM[22] workload manager.

Implementation

A crucial question that has arisen was what repeating parts of the processing could be done independently from one another. These parts would then be subjected to independent parallel processing units (separate CPU cores or separate computers in a cluster). The answer needed to be found for each stage separately. Considering the construction of each stage (look at Fig. 4.11-4.25), one can see that many of the loops consist of independent computations. For example, for STAGE 1, source projection is performed for each subject separately, as the loop in Fig. 4.11 shows. Each task related to a single subject can be handled by one of the CPU cores. The limit of 24 CPUs[23] per machine in CI-TASK forced to dividing the set of subjects into 24-element subsets so each machine could handle a single subset. A similar case was STAGE 5, where regions needed to be divided into 24-element subsets first. The size of a subset can be changed in the toolbox configuration (Section 4.4.6). Knowing that computations can be distributed both across machines (each computing single so-called "job") and CPUs, I could optimize two loops simultaneously. For example, in STAGE 4, there are two

[21] MATLAB Parallel Computing Toolbox offers set of instructions such as `spmd`, `parfor`, `pareval` ; `https://www.mathworks.com/products/parallel-computing.html` (accessed: June 4, 2022)

[22] `https://slurm.schedmd.com/` (accessed: June 4, 2022)

[23] Physically it possesses 12 physical processing units, split virtually into two concurrent units. Thus there are 24 independent workers available. In the text, to avoid confusion, by "CPU" I mean those virtual workers that can perform independent parallel calculations.

Table 4.2. Division of the input data across machines (jobs) and CPU cores for each of the stages when performing concurrent computations using ToFFi.

	Single **job** handles ...	Num. of **jobs** possible	Single **CPU core** handles ...
STAGE 1	set of subjects	≥ 1	1 subject
STAGE 2	set of subjects	≥ 1	set of iterations
STAGE 3	n/a	1	n/a
STAGE 4	set of regions	≥ 1	set of iterations
STAGE 5	set of regions	≥ 1	subset of the set of regions
STAGE 6	set of CV-repetitions	1	subset of the set of CV-repetitions
STAGE 7	set of subjects	≥ 1	set of iterations
STAGE 8	set of regions	1	subset of the set of regions

main loops where most of the labor happens — the regions loop and the evaluation iterations loop (Fig. 4.17). STAGE 3 is an exception — its computations are too short and too simple to benefit from concurrency, so they remained serial. Table 4.2 summarizes which data elements were treated as independent.

`RUN_SELECTED_SLURM.sh` is used to launch multiple stages of Spectral Fingerprinting at once on a computer cluster. This script contains calls to `sbatch` function with `STAGE_X.sl` file as an argument. `.sl` file includes parameters (lines starting with `#SBATCH` keyword) describing needed resources to be allocated and the script's name to be launched. An example of `STAGE_1.sl` file responsible for running STAGE 1 of Spectral Fingerprinting (II) module is presented in Listing 8.

Listing 8: Code of `STAGE_1.sl` script.

```
#!/bin/bash
#SBATCH -J H1_S1
#SBATCH -p small
#SBATCH --oversubscribe
#SBATCH -N 1
#SBATCH --ntasks-per-node 3
#SBATCH --mem 250000
#SBATCH --time 00:20:00
#SBATCH --mail-type=NONE
#SBATCH --array=1-1
#SBATCH --output=S1-%A_%a.out
#SBATCH --license=matlab_dct@licencje.task.gda.pl:1

iStage=1

cd "STAGE_$iStage/scripts/"
folder=`pwd`
module load tryton/matlab/2021a
matlab -nodisplay -nodesktop -logfile $folder/../../STAGE_${iStage}.log <  $folder/RUN
    .m
```

SLURM interprets the contents of `.sl` files as a request to schedule the launch of a given script in a proper location inside the queue that consists of requests made by

other users of the cluster. One of the parameters specified by the user is the number of machines (also called *nodes*) to be allocated. A set of instructions to be executed by a machine is called a *job*. Each machine will execute the same set of instructions contained within `.sl` file. MATLAB can identify which machine executes the given script and return the machine ID. Using this index, each machine can process a different chunk of the data. In this vein, `RUN_SELECTED_SLURM.sh` causes the execution of the following Bash scripts:

- `STAGE_1.sl` → allocates and queues multiple nodes to execute `RUN.m` [STAGE 1]
- `STAGE_2.sl` → allocates and queues multiple nodes to execute `RUN.m` [STAGE 2]
- ...
- `STAGE_8.sl` → allocates and queues multiple nodes to execute `RUN.m` [STAGE 8]

Position of a particular job in a queue is determined by the availability of several resources including free nodes, requested computation time, and memory. Taking STAGE 1 as an example, let us assume SLURM allocated M-nodes and that there are N-CPU cores on each node. Then job identifier is used to divide data into chunks:

- Job 1 → Node 1 → `RUN.m` processes `data(1, :)`
- Job 2 → Node 2 → `RUN.m` processes `data(2, :)`
- ...
- Job M → Node M → `RUN.m` processes `data(M, :)`

where `data` is M by N matrix and `data(i, :)` denotes i-th row of the `data` matrix. This division can be depicted as in Fig. 4.27.

Launching Spectral Fingerprinting with `RUN_SELECTED.m` script does not allow to perform computations on more than one machine, however, `spmd` blocks can still be used, which is equivalent to splitting `data` matrix into columns only.

The user also needs to consider the stages interdependence mentioned before (Fig. 4.10). A certain stage cannot be executed before related stages are done first. This is a natural computational barrier that is not explicitly put in the code, resulting in errors when not respected.

Another valuable piece of advice is to use as many CPU cores as possible instead of multiplying the number of jobs, as the job is a separate MATLAB session that needs a separate license to be used — their quantity per user is limited most of the times.

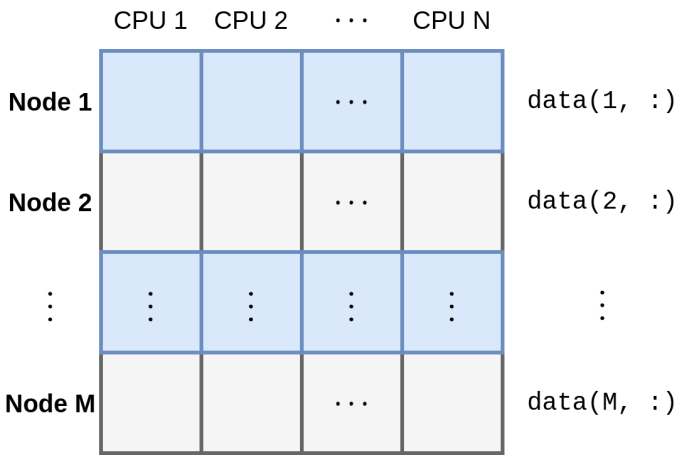


Fig. 4.27. Illustration of how `data` matrix is divided across machines in a cluster (nodes) and their CPUs. It is assumed that each node has at least N CPUs.

However, since there might be a time and memory limit per job set in the workload manager, sometimes increasing the number of jobs is necessary to meet these resource constraints.

The need for splitting computations across many machines and CPUs left its mark on the code design of Spectral Fingerprinting (II). First, in `CONFIGURE.m` script, one can notice `CFG.(STAGE_NAME).MODE` parameter for each of the configured stages (or `SX_MODE` variable when configuring via `.sh` file; `X`$= 1, 2, \ldots, 8$). It can be assigned one of two possible values: `'serial'` or `'parallel'`. When it is assigned `'serial'`, computations in a given stage will be performed on a single CPU core on a single machine. When `'parallel'` is set, then computations in a given stage will be automatically split across machines and CPUs.

`STAGE_X/scripts/RUN.m` files, serving as an entry point for respective stage, share an almost identical structure dictated by the `switch MODE` statement (STAGE 3 is an exception, as it can only be executed serially). `RUN.m` files follow roughly the same template:

In Listing 9 line 5 introduces `switch` instruction, which directs the flow of the program depending of which mode of processing was chosen. In the `case 'serial'` block, computational load (`dataSet`) is split into portions (`data`) and processed serially inside a for a loop. Note that for each data the same processing pipeline called `CORE` is used. `data` can represent a single subject, single region, or another independent piece of information to be processed. In the `'parallel'` block, a single `dataChunk` is extracted from the load according to which machine `RUN.m` script is executed on (this information is contained within `jobInfo` structure). Each machine gets different `dataChunk` to operate on. Then, different parts of the `dataChunk` are processed inside

Listing 9: Template of the `RUN.m` files.

```
1  %% RUN.m
2  [Initialization statements]
3  [Data check statements]
4
5  switch MODE
6    case 'serial'
7
8      for data in dataSet
9            CORE(CFG, data);
10     end
11
12   case 'parallel'
13
14       dataChunk = extractDataChunk(dataSet, jobInfo);
15
16     spmd(numCPU)
17       data = dataChunk(labindex);
18           CORE(CFG, data);
19     end
20
21   otherwise
22     error('ERROR: Wrong value of MODE !');
23 end
24 %%
```

`spmd` block[24]. Statements inside `spmd(n)` block are copied to n so-called *workers* (which usually denote separate CPU cores) for simultaneous execution. The order of statements is preserved. Each MATLAB worker has a unique value of `labindex` variable thus it is used to extract a particular piece of load (`data`) from `dataChunk` variable (line 14). Then `CORE` is executed simultaneously on those separate data pieces.

Speedup analysis

There is a considerable speedup when using more than one node with multiple CPU cores, as was shown in Fig. 4.28).

Eight data sets, precomputed in the past, were used in this example. They were different in size and were analyzed using ToFFi Toolbox set up with different processing parameters. For each data set, the computational load required to complete all the stages of Spectral Fingerprinting (II) was split into multiple jobs. Each job was assigned a separate node equipped with 24 CPU cores. The number of jobs differed across data sets as the size of data sets differed as well. Wall-clock time (i.e., the difference between the time at which a task finished and the time the task started) of each job being processed was recorded. Knowledge about which job belongs to which stage is

[24] `https://www.mathworks.com/help/parallel-computing/spmd.html` (accessed: June 4, 2022)

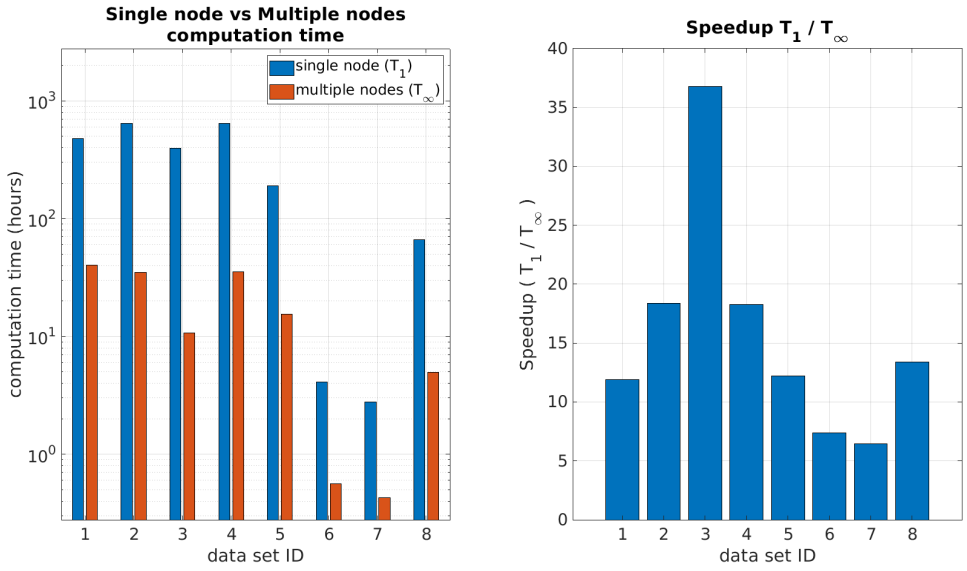


Fig. 4.28. Left: Spectral Fingerprinting (II) module total computation time for eight different data sets. Blue bars represent computation time for using a single node equipped with 24 CPU cores (T_1; Eq. 4.2). Red bars denote computation time when stages of Spectral Fingerprinting are computed concurrently using multiple 24-CPU nodes (T_∞; Eq. 4.3). **Right**: Speedup achieved when using multiple nodes, expressed as T_1/T_∞ ratio. Speedup value is constrained by the stage interdependence (Fig. 4.10).

important as it is impossible to run stages in totally arbitrary order (see Fig. 4.10). By:

$$\tau_j^{(s)}$$

the wall-clock time of a j-th job for s-th stage of processing was denoted ($s = 1, 2, \ldots, 8$). In single-node approach, one job at a time can be processed and total computation time for stages $s = 1, 2, \ldots, 8$ were denoted as $t_1^{(1)}, t_1^{(2)}, \ldots, t_1^{(8)}$, precisely:

$$t_1^{(s)} = \sum_j \tau_j^{(s)}$$

For each stage, the number of jobs may differ thus upper bound for this sum is omitted. Spectral Fingerprinting total computation time in the single-node approach, presented in Fig. 4.28, was thus given as:

$$T_1 = \sum_{s=1}^{8} t_1^{(s)} = \sum_{s=1}^{8} \sum_j \tau_j^{(s)} \tag{4.2}$$

In the multiple-node approach, it was assumed that an unlimited number of nodes can be used, however, one must **not** estimate Spectral Fingerprinting total computation time as $T_\infty = \max_s(\max_j \tau_j^{(s)})$ because of stage interdependence (Fig. 4.10). Accounting

for that constraint, the formula for T_∞ depicted in Fig. 4.28 was given as:

$$T_\infty = t_\infty^{(1)} + t_\infty^{(2)} + t_\infty^{(3)} + \max(t_\infty^{(4)}, t_\infty^{(7)}) + \max(t_\infty^{(5)}, t_\infty^{(6)}) + t_\infty^{(8)} \tag{4.3}$$

where $t_\infty^{(s)} = \max_j \tau_j^{(s)}$. According to Fig. 4.10, stages 1, 2, and 3 must be computed in succession first to produce output that fulfills requirements to launch both stages 4 and 7. They can be computed concurrently. Completion of those produces output necessary to run both stages 5 and 6 concurrently. Finally, Spectral Fingerprinting was concluded by launching stage 8. After substitution and rearrangement, Eq. 4.3 can be written as:

$$T_\infty = \left(\sum_{s\in\{1,2,3,8\}} \max_j \tau_j^{(s)} \right) + \max\left(\max_j \tau_j^{(4)}, \max_j \tau_j^{(7)} \right) + \max\left(\max_j \tau_j^{(5)}, \max_j \tau_j^{(6)} \right)$$

Ideally, to fully express the benefits of implemented parallelism, a plot with time difference between fully serial and fully concurrent computation times would be appropriate, however, it would demand time-consuming data collection that was not possible at the time of writing this thesis. For this reason, a slightly simplified approach was presented, which still proves considerable speedup achieved when calculating fingerprints with the ToFFi Toolbox's `'parallel'` mode.

4.8 Reproducibility

A growing number of reports show that the results of experiments in many domains of science could not be recreated. There is a serious concern about the reproducibility of scientific research (Fig. 4.29). Computer scientist Jon Claerbout coined the term *reproducibility* [225] to remind that an investigator must have the means to recreate figures and data tables from the data of the given piece of research. Researchers need to have the ability to duplicate the results of a prior study using the same materials used by the original investigator [226]. Otherwise, how can one be sure that drawn and published conclusions are credible?

Implementation

In the ToFFi toolbox, reproducibility is supported by several means. First, the code of the toolbox will remain open-source for the community as the software disclosure is demanded by the GNU LGPL 2.1 license[25]. Second, along with the toolbox code, extensive documentation is provided (a link to the documentation can be found in

[25] `https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html` (accessed: June 4, 2022)

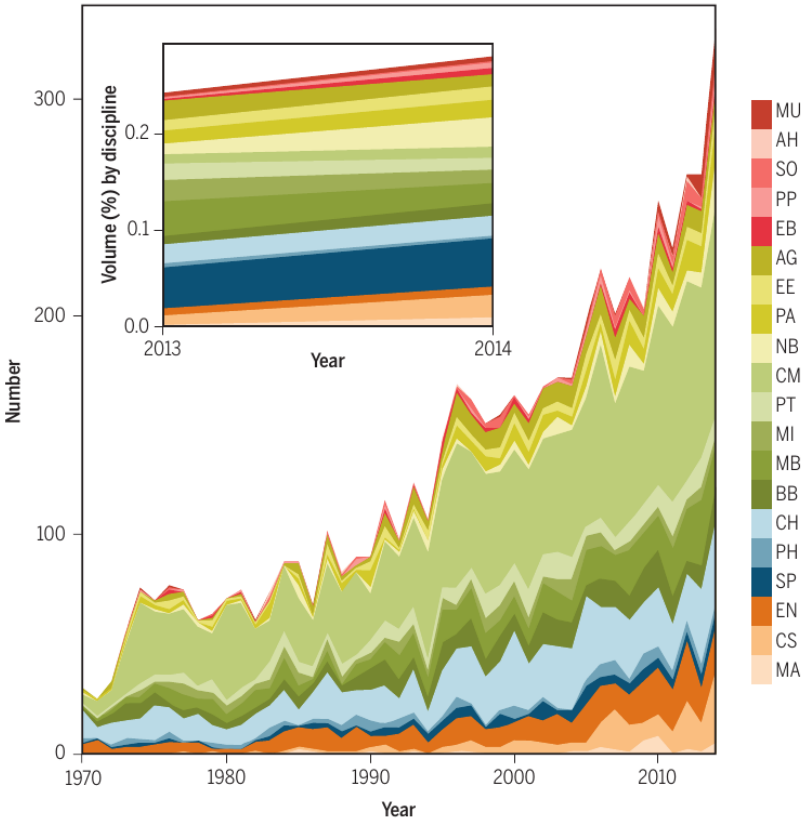


Fig. 4.29. The number of publications recorded in Scopus that have, in the title or abstract, at least one of the following expressions: "research reproducibility", "reproducibility of research", "reproducibility of results", "results reproducibility", "reproducibility of study", "study reproducibility", "reproducible research", "reproducible finding", or "reproducible result". Papers are grouped by discipline on the basis of the journal, following an adaptation and expansion of Thomson Reuters' Essential Science Indicators classification system. Journals not included in the latter database were hand-classified on the basis of their name. The subplot reports the percentage over the total number of records for each discipline, in the last 2 years of the series. Disciplines legend: MA, mathematics; CS, computer sciences; EN, engineering; SP, space science; PH, physics; CH, chemistry; BB, biology and biochemistry; MB, molecular biology; MI, microbiology; PT, pharmacology and toxicology; CM, clinical medicine; NB, neurobiology and behavior; PA, plant and animal sciences; EE, environment and ecology; AG, agricultural sciences; EB, economics and business; PP, psychology and psychiatry; SO, social sciences, general; AH, arts and humanities; MU, multidisciplinary. The time series was truncated at 2014. Source: [226].

Appendix C). Moreover, coauthors of the toolbox are willing to publish their work in journals that demand open-source code and data.

Some parts of the Spectral Fingerprinting pipeline (e.g., clustering) demand simulating randomness. This theoretically threatens reproducibility. In practice, this is not a problem, as I implemented in the toolbox configuration routines sufficient means of control of MATLAB's pseudo-random number generators. The description of this will follow.

For each stage individually, one can initialize *pseudo-random numbers generator* (abbreviated here as RNG) in two possible ways:

1) with a fixed seed value,
2) with a time-based seed value.

In terms of dynamical systems, the seed value can be thought of as the number that determines the starting point of the generator's trajectory. Generated sequences of pseudo-random numbers are entirely determined by the seed value.

Using method 1), one can get full reproducibility of results as long as the other configuration parameters or input data remain the same between subsequent runs. Using method 2), one will obtain different results from subsequent algorithm runs.

Seed values are passed to jobs and CPU cores according to the hierarchy described below.

Seed value hierarchy

Seed values from the configuration file are called *Global Seeds*. Each stage has its own Global Seed that can be set by the user by modifying `rngSeed` field of the `CFG.STAGE_X` variable during configuration (Section 4.4.6). For method 1), the user inputs a natural number. For method 2), the user inputs `'time'` as the value of the `rngSeed` field, causing the generator to be initialized based on the current time.

Global RNG is seeded with Global Seed and generates a sequence of "child" random seeds - one for each job (*Job Seed*). Jobs are separate MATLAB sessions that compute **simultaneously**. Each job has its own RNG. Inside a job, data are divided into separate batches. Batches inside a single job are handled **sequentially** — one after another.

A new set of random numbers is generated inside the job. These numbers then serve as *Batch Seeds*. Each batch obtains a unique Batch Seed. This Batch Seed will serve as a seed to initialize a random stream generator (see `RandStream` in the MATLAB Documentation[26]) which will create a separate stream of random numbers for each CPU

[26] `https://www.mathworks.com/help/matlab/ref/randstream.html` (accessed: June 4, 2022)

core. Inside each batch, there is a set of CPU cores processing a given chunk of the data. CPU cores work **in parallel**. An individual stream of pseudo-random numbers is assigned to each CPU core, as was said before. In consequence, each CPU core has its own set of pseudo-random numbers for its disposal. This guarantees that there is no overlap between sets of random numbers of CPU cores, batches, and jobs.

Suppose that data is divided between 2 machines (thus between 2 jobs), where each data subset is divided into 3 batches. Assume that, by a proper configuration (`CFG.Global.maxNumSpmdWorkers` variable), it is allowed to use 4 CPUs. Each batch is then handled in parallel by 4 CPUs. Then, the generation of seeds and pseudo-random number streams can be illustrated as follows:

```
Global Seed (G; generates random numbers J1 and J2)
├── Job 1 Seed (J1; generates random numbers B11, B12, B13)
│   ├── Batch 1 Seed (B11)
│   │   └── set of unique random streams (set initialized with B11)
│   │       ├── stream for CPU 1
│   │       ├── stream for CPU 2
│   │       ├── stream for CPU 3
│   │       └── stream for CPU 4
│   ├── Batch 2 Seed (B12)
│   │   └── set of unique random streams (set initialized with B12)
│   │       ├── stream for CPU 1
│   │       ├── stream for CPU 2
│   │       ├── stream for CPU 3
│   │       └── stream for CPU 4
│   └── Batch 3 Seed (B13)
│       └── set of unique random streams (set initialized with B13)
│           ├── stream for CPU 1
│           ├── stream for CPU 2
│           ├── stream for CPU 3
│           └── stream for CPU 4
└── Job 2 Seed (J2; generates random numbers B21, B22)
    ├── Batch 1 Seed (B21)
    │   └── set of unique random streams (set initialized with B21)
    │       ├── stream for CPU 1
    │       ├── stream for CPU 2
    │       ├── stream for CPU 3
    │       └── stream for CPU 4
    └── Batch 2 Seed (B22)
        └── set of unique random streams (set initialized with B22)
            ├── stream for CPU 1
            ├── stream for CPU 2
            ├── stream for CPU 3
            └── stream for CPU 4
```

As long as the input data and configuration are the same, and as long as the user configured seeds to a fixed value, computations will yield the same result each time repeated. This guarantees reproducibility of research done using ToFFi Toolbox.

4.9 Demonstration

The following section demonstrates the ToFFi toolbox capabilities by explaining how to calculate and study fingerprints in the 1–40 Hz frequency interval. Different distant parts of the brain were analyzed to show that fingerprints vary among areas significantly. Characteristic frequencies of regional activity were found. The similarity of fingerprints was assessed, as well as how well one can discriminate between them (identification). This example was limited to group-level computations only to limit the length of this section. A complete step-by-step version of this demonstration and guidance needed to obtain the input data can be found in the toolbox documentation (Chapter 4 therein; a link to the documentation provided in Appendix C).

Eight brain regions were chosen from the Desikan-Killiany atlas (Fig. 4.30). This atlas is one of the implemented brain parcellations available in ToFFi. Chosen regions occupy different parts of the brain, thus their activity reflects various functional roles (described in Fig. 1.10 in Section 1.3). This way, computed fingerprints can capture different modes of activity of the human brain.

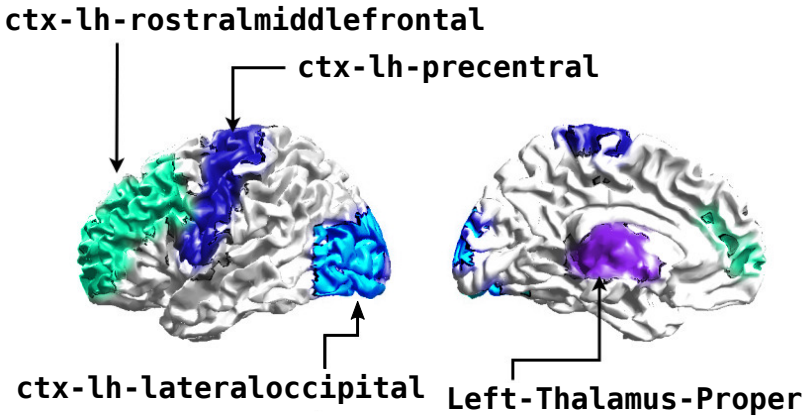


Fig. 4.30. Eight brain regions chosen from the Desikan-Killiany atlas for the purpose of the demonstration. Here only the left counterpart is depicted, whereas right hemisphere homologues were chosen as well.

Data for the first ten participants were selected from The Human Connectome Project (HCP) Database[27]. They all have anatomical head models, source models, and

[27] `https://www.humanconnectome.org/study/hcp-young-adult/document/1200-subjects-data-release` (accessed: June 4, 2022)

approximately three minutes of 248-channel MEG segmented resting-state recordings available [200]. Preprocessed MEG signals, anatomy data containing head models, and source models were needed for source projection (STAGE 1). The steps described in the next paragraphs followed the typical workflow presented in Section 4.5.

Initial adjustments preceding Spectral Fingerprinting were made. Data Preparation (I) module scripts were executed in MATLAB in the following order: `PREPROCESSING.m`, `COVARIANCE_GEN.m`, `LEADFIELD_GEN.m`, `LCMV_FILTER_GEN.m`. which resulted in MEG recording being detrended and divided into subsequent short 1-second long segments, individual-subject average data covariance matrices estimated, and beamformer spatial filter constructed. Names of the output data files were adjusted using `RENAMING_TOOL.m`.

Next, Spectral Fingerprinting (module II) was configured by adjusting input parameters stored in `CONFIGURE.m` file and then launched in MATLAB. Parameters concerning the formation of spectral clusters were set to provide the optimal number of clusters both for individual and group-level fingerprints. Pseudo-random number generators were set to fixed values to enable reproducibility (Section 4.8) of the example.

To obtain group-level fingerprints and analyze them according to the plan of this demonstration, stages 1, 2, 3, 4, 5, 6, and 8 were run using `RUN_SELECTED.m` script. STAGE 7 was omitted as it estimates individual fingerprints identification accuracy, which is outside the scope of this demonstration and is time-consuming. The interested reader is referred to the full version of this example in the toolbox documentation. The running time of the demonstration presented here, on a machine equipped with a 64-bit Intel© Core™ i5-6200U 2.3 GHz processor and 16 GB RAM, should last about 40 minutes when executed concurrently using 2 CPU cores.

Completion of `RUN_SELECTED.m` script without errors led to the presence of `.mat` files containing results in each of the `STAGE_X/output/` directories. These files were visualized and interpreted with the help of Presentation scripts (module IV).

First, group-level spectral fingerprints computed in STAGE 5 were visualized by running `plotSpectralFingerprint.m` routine (Fig. 4.33).

Fingerprints have a similar shape among homologue areas, as it was for individual fingerprints, suggesting that homologue areas may have similar functions. Precentral areas situated in the proximity of the motor cortex tend to peak in the beta band. Occipital areas have the strongest power in the alpha band. Frontal areas peak mainly in the theta and gamma band. Thalamus has many spectral modes which peak at different frequencies. The location of the peaks is in line with the properties known from brain physiology (see brain rhythms characteristics outlined in Section 1.5).

Each spectral mode's time is computed as a sum of points' duration (i.e., spectral modes from individual fingerprints) divided by the number of unique subjects that

these points belong to (Algorithm 7 Step 17). This procedure was inspired by the original publication [9]. This could yield cumulative percentages above 100%, which also happened in this demonstration.

Every area has a single *stable mode* (i.e., one that is represented by at least five subjects - filled dots in legends) which happens to be a *dominant mode* as well (having the largest duration percentage). Other modes represent individual variability that happens to be similar across few subjects, rendering them as less important, are termed *unstable*.

Second, fingerprints computed in STAGE 5 were subjected to similarity analysis in STAGE 8. The result of it was depicted using `networksPlot.m` routine (Fig. 4.31). As expected, similar fingerprints representing homologue brain areas were matched together.

Finally, the results of the group-level identification procedure were illustrated with the aid of two scripts: `groupIdentificationPlot.m` and `groupHitMatrix.m` (Fig. 4.32). Due to the similarity of fingerprints discovered above, it was expected that one region of the homologous pair would achieve a high accuracy value, whereas the other would be identified with low accuracy, which indeed happened. Hit matrix allowed to confirm that homologue areas contributed mostly to the confusion, but also other areas with similar fingerprints did, as some hits were located outside the 2x2 diagonal submatrices outlined in red (Fig. 4.32b).

During the course of this demonstration, Maintenance scripts (module V) helped with managing the data whenever needed.

In summary, in this chapter, the Toolbox for Frequency-based Fingerprinting was presented. Its development was properly motivated as there was no software available to facilitate systematic spectral analysis of human brain fingerprints. The capabilities of the toolbox were also demonstrated. ToFFi is able to discover common spectral patterns in resting-state multichannel recordings using Spectral Fingerprinting pipeline. It discriminated between different modes of operation for a range of brain areas defined in the atlas of choice. Dominant and supportive oscillation profiles were recognized and separated. Functional similarity between homologue areas was confirmed using hierarchical clustering analysis. Identification accuracy was high for the individual fingerprints. Recognition of the brain areas based on the spectral fingerprints turned out to be challenging among homologue areas due to the aforementioned functional similarity, yet remained still informative.

The toolbox's operation can be adjusted to solve independently different problems thanks to the versatile configuration and modularity of its architecture. Parallel computations using multiple machines help to analyze bigger data sets in a fraction

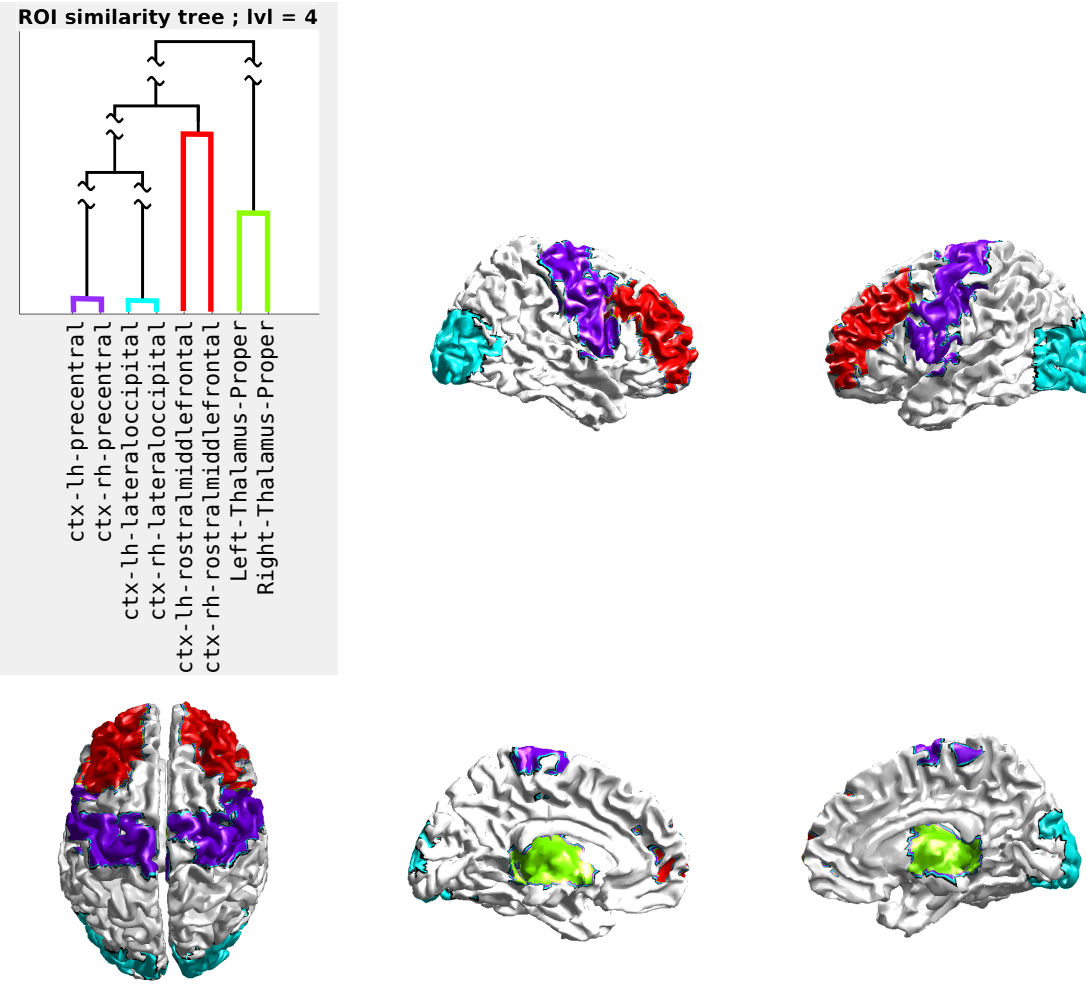


Fig. 4.31. Result of the hierarchical agglomerative clustering of the spectral fingerprints presented in Fig. 4.33. Homologue areas were automatically matched together according to the similarity of their fingerprints. The similarity tree has disproportionately long branches that were broken for clarity (waved lines).

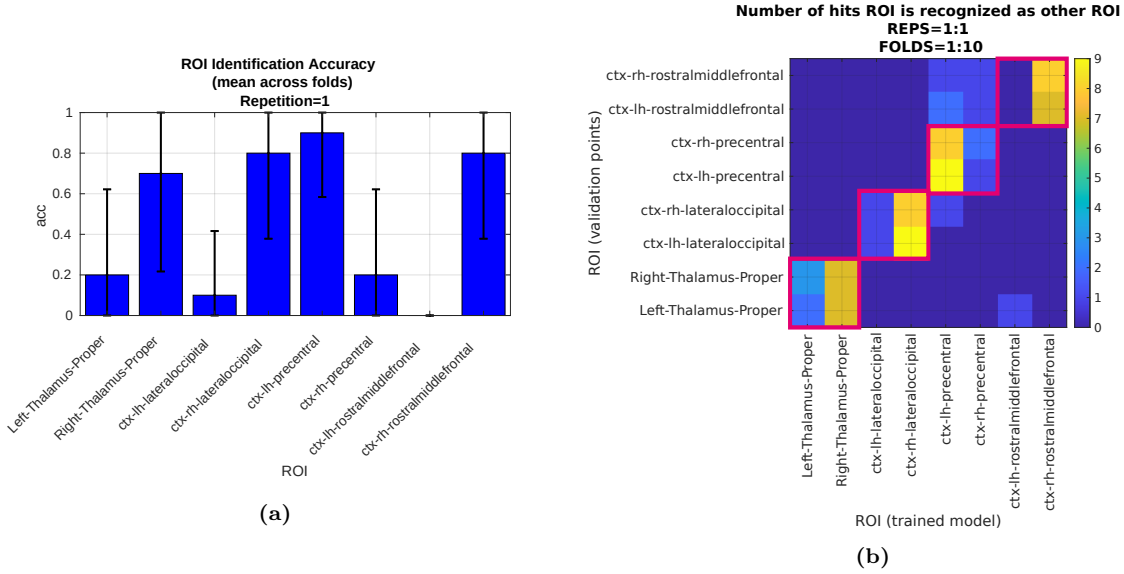


Fig. 4.32. Group-level identification accuracy: a) bar plot showing the average identification accuracy across cross-validation iterations (leave-one-out), b) confusion matrix showing in each row a distribution of "votes" for each ROI. Each ROI was tested ten times (model trained on nine subjects versus one validation subject). For ideal identification, this matrix would have a value of 10 for the diagonal elements and zeros elsewhere. Confusion happens mostly between homologue areas (2x2 red boxes). Left hemisphere ROIs are recognized as the right hemisphere homologue areas.

of time, compared to the serial calculations, as was illustrated in the computational speedup analysis. Consideration regarding time complexity using asymptotic analysis based on the ToFFi algorithms flowcharts gives some intuition on how the input data size affects computations of the toolbox. Toolbox implementation embraced the growing awareness of reproducibility by providing means to control the generation of pseudo-random number streams. The possibility for further development of the toolbox remains open, as this software is given to the community as an open-source package.

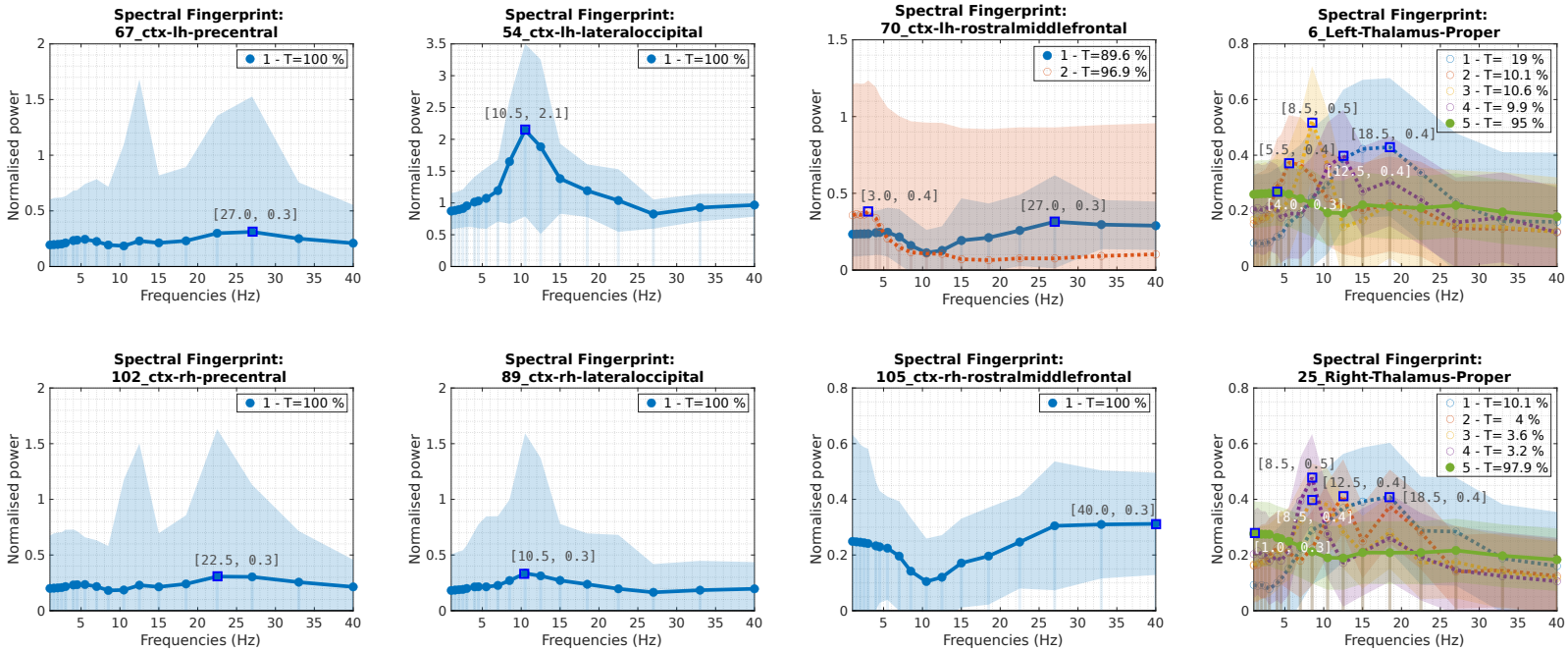


Fig. 4.33. Resting-state spectral fingerprints for Desikan-Killiany brain atlas in the 1–40 Hz frequency interval. Each column shows two homologue brain areas. Legends show the corresponding duration of each spectral mode (i.e., the percentage of trials in which each spectrum was present on average during recording) and whether the mode was present for at least five subjects (filled dot) or not (empty dot). The frequency axis was configured to be logarithmic in order to optimize the lower frequencies resolution. Y-axis depicts the power normalized in relation to the average spectrum of the whole brain. Shaded regions depict the standard deviation (1σ) of the corresponding spectral mode. For i-th of total F frequencies of interest, standard deviation was estimated as $\sqrt{\Sigma_{i,i}}$, where $\Sigma_{i,i}$ is the i-th diagonal entry of the covariance matrix of the Gaussian Mixture Model component corresponding to the given spectral mode. Standard deviations have relatively large values due to the small number of subjects used in the illustrative example. Homologue areas have very similar fingerprints.

5 Exploring oscillations across brain networks with ToFFi Toolbox

The previous chapter elaborated on ToFFi Toolbox architecture and functionalities. This chapter shows that ToFFi Toolbox can aid in solving real-life research problems.

I have outlined the initial exploratory study of oscillatory hierarchies in the resting human brain networks. Its novelty lies in the synthesis of magnetoencephalography (Chapter 1), spectral modeling of the human brain activity (Chapter 3), and the non-linear topology analysis — a combination that has not appeared in the literature. The presented material will be extended in a journal publication that is currently in preparation. For this reason, I have focused here more on results and observations, limiting the Discussion section. Values of the ToFFi toolbox configuration parameters used for the study can be found in Table A.2.

The study is in progress and emerged as the collaborative effort of the author of the thesis and the following coauthors: dr Katarzyna Jurewicz, dr Joanna Dreszer, mgr Jakub Wojciechowski, dr Krzysztof Rykaczewski, dr Anne Keitel, dr hab. Tomasz Piotrowski, prof. Włodzisław Duch.

5.1 Introduction

One of the interesting questions posed in neuroscience articles was whether there is a hierarchy of frequencies across the human brain cortex. There is no consensus on this matter. For the 1-45 Hz band, there is the anterior-posterior gradient of frequencies across the human brain [92], however, if we study faster oscillations up to 150 Hz, locally characteristic frequencies do not constitute a simple gradation [91].

Frequency modulations are claimed to serve as communication carriers between different functional brain networks [227][228][168]. A large body of work showed that one could analyze these networks using different modalities, including MEG [160][159], and extract features of networks' activity that organize them in a particular hierarchy. If hierarchical networks exhibit oscillations at specific frequencies, then frequency relation could form a hierarchy as well.

In our study, definition of brain resting-state networks (RSNs; Section 2.2 in Chapter 2) served as an organizing principle. Looking at human cognition through the lens of RSNs proved to be beneficial, as the number of publications concerning this topic constantly grows (Fig. 5.1).

To account for the dynamic nature of brain activity, spectra that corresponded to short segments of data were processed with Spectral Fingerprinting. This way, multiple

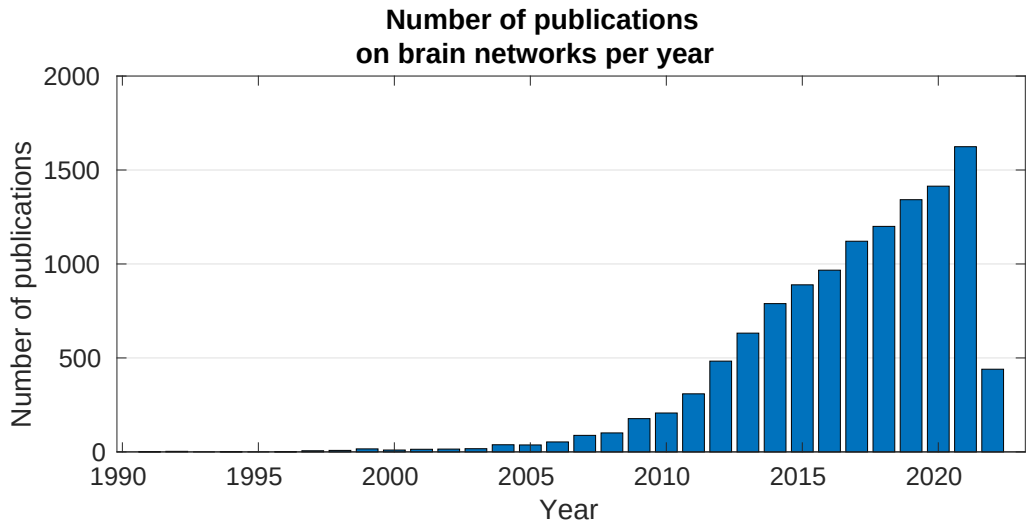


Fig. 5.1. The number of publications concerning brain networks in the years 1991–2022. Based on: Web of Science. URL: `https://www.webofscience.com/wos/woscc/summary/d252b5f5-1d41-44d2-8380-3eea4385feeb-39c7366e/relevance/1` (accessed: May 23, 2022).

spectral profiles per area, instead of a single spectrum for each region, enriched further steps of the pipeline with information that may be otherwise lost.

Frequency hierarchy that we were looking for was not expected to be necessarily linear, so diffusion embedding, which is a non-linear dimensionality reduction technique, was used. To enable pairwise spectral fingerprint comparison needed as the input for diffusion embedding, I have used a quasi-norm based on the Wasserstein metric (also known as the *Earth Mover's Distance*, EMD; [229]).

The choice of diffusion embedding immediately placed our study on the fertile ground of so-called gradient methods [230][13], employed successfully in different contexts [231]–[241]. In particular, Margulies and colleagues discovered that RSNs follow the sensory-to-transmodal hierarchy [13], which means that areas responsible for the initial processing of internal and external stimuli are placed at one end of the proposed gradient. Areas serving as integrators to preprocessed initial information are located at the other end of the gradient. Our study examined the connections between resting-state gradients proposed by Margulies, based on fMRI data similarity (functional connectivity; Section 2.3), and gradients based on resting-state MEG spectral fingerprints similarity.

As a result of the methodology described in the next section, we obtained two significant components that explain over 90% of data variability. In [13], these components were called *gradients*, and we follow this nomenclature here, however, note that they are not equivalent to the mathematical definition of the gradient, which is:

$$\nabla f : \mathbb{R}^n \to \mathbb{R}^n$$

$$\nabla f(p) = \left[\frac{\partial f}{\partial x_1}(p), \ldots, \frac{\partial f}{\partial x_n}(p)\right], \quad p = (x_1, \ldots, x_n)$$

In neuroscience, gradient is a graded variation of a neuroanatomical feature within a region or across the whole brain.

Our gradients separate well visual from somatomotor regions, but associative regions cannot be easily told apart. Moreover, the gradients are moderately correlated with the gradients found by Margulies and colleagues [13], and they are also partially linked to the distance of a brain region to the anatomical center of the brain, confirming the significance of anatomical centrality as a contributing factor in describing the functional geometry of the resting brain proposed by Zhang and collaborators in 2019 [238].

5.2 Methods

Mathematical symbols follow the convention presented in the Notation section of Appendix A.

5.2.1 Participants

We selected N=89 participants (age: mean = 29, sd = 3.6, range = 22-35; gender: 41 females) from The Human Connectome Project (HCP) that have anatomical head models, source models and MEG resting-state recordings available [200]. We downloaded their preprocessed MEG signal, anatomy data containing head models, and source models needed for source projection. For more details on the dataset, consult WU-Minn HCP 1200 Subjects Data Release Reference Manual[28].

5.2.2 Data acquisition and preprocessing

Data acquisition hardware, protocols, and preprocessing routines are described in detail in [242]. We chose the first of three runs of MEG acquisition for all subjects. Two other runs were discarded because short intervals between runs may have impacted the subjects' resting state.

Data preprocessing was done using the HCP Megconnectome 3.0 pipeline[29], which excluded bad channels and increased the signal-to-noise ratio by automatic artifact removal, aided with the visual inspection.

For Spectral Fingerprinting, we employed ToFFi Toolbox. For preprocessing, we used the Data Preparation (I) component. Data were thus cut into time segments of

[28] `https://www.humanconnectome.org/study/hcp-young-adult/document/1200-subjects-data-release` (accessed: June 4, 2022)

[29] `https://www.humanconnectome.org/software/hcp-meg-pipelines` (accessed: June 4, 2022)

1-second length, and linear trends were removed from each segment (as in [9]). Data covariance matrices were estimated as an average of single-segment covariances (Step 2 of Algorithm 3). Next, single-shell scalar lead fields were computed based on the individual 8 mm^3 source models and head models obtained from the anatomical part of the HCP Megconnectome 3.0 pipeline. The lead field defines how unit activity at every source grid location propagates to the MEG sensor array according to Eq. A.1. Lead field matrix coefficients were obtained assuming optimal (i.e., giving the maximum spatial filter output power) dipole direction using EVD decomposition shown in Algorithm 2 [174]. Finally, in order to reconstruct source activity, LCMV spatial filters were generated from the data covariance matrices and scalar lead fields according to the Van Veen formulation (Eq. A.3; [142]), assuming linear mixing of the sources (Eq. A.1).

In summary, after preprocessing steps, the data ready for Spectral Fingerprinting presented itself as follows: 89 subjects, 1 run of cleaned MEG eyes-open resting-state multichannel signal, number of channels left: mean = 243, sd = 1; the number of 1-second segments: mean = 218, sd = 17; 1-150 Hz bandwidth (that was reduced later during Spectral Fingerprinting into 1-120 Hz to match the default frequency band proposed by Keitel and Gross [9]); 509 Hz sampling frequency. Each subject was provided with a single-shell model LCMV spatial filter, defined in the common MNI space 8 mm^3 grid, consisting of 5798 inside-brain voxels. This set of routines was coded using MATLAB 2021a and Fieldtrip Toolbox (release 20210816).

5.2.3 Constructing spectral representation of regional resting-state activity

Figure 3.1 is a reminder of how the construction of spectral representations of regional resting-state activity is achieved by Algorithm 7.

Computing spectral fingerprints started with the source activity reconstruction step. For each individual, a spatial filter matrix was applied to the sources signal matrix to obtain an array of reconstructed activity, according to Eq. A.5.

Multitaper approach for Fast Fourier Transform (FFT) with the first three discrete prolate spheroidal sequence windows (DPSS, [243][214]) was used to estimate the spectrum of each 1-second segment (Algorithm 1). Spectra were spatially normalized by dividing the spectrum of each segment and voxel by the average power spectrum across all segments and voxels, for each participant separately, according to Step 6 of Algorithm 3.

Each voxel of the source model was assigned to a region of interest (ROI) according

to the Schaefer brain segmentation with 100 regions [217]. The segmentation is available online[30]. This brain atlas is based on the resting-state functional connectivity, and it contains regions that are units of the 7 resting-state networks outlined by Yeo [153], Fig. 2.2. These networks are: Dorsal Attention Network (DAN), Default Mode Network (DMN), Frontoparietal Network (FPN), Cingulo-opercular Network (CON), Visual Network (VIS), Somatomotor Network (SMN), Limbic Network (LIM). Their functional role aligns with the roles of different brain lobes presented in Chapter 1 (Fig. 1.10).

Having ROIs defined, spectral power inside each ROI was averaged across all its voxels. For each subject and region separately, the mean power of each 1-second segment and segment z-score relative to all the segments were computed. Segments with a z-score exceeding the threshold of $\Gamma = 2.5$ standard deviations were rejected from the further analyses (Algorithm 3 Step 8).

Individual-level fingerprints were calculated for each subject and each ROI separately. Single-segment normalized power spectra were grouped into an optimal number of clusters per anatomical area using k-means (Algorithm 4), which treated each single segment power spectrum as a point in $F = 42$-dimensional space, forming k mutually exclusive clusters. The final solution minimizes the centroid distance of observations within clusters and maximize the distance between clusters. The optimal value of the k parameter was found using 250 independent iterations of Silhouette evaluation (Algorithm 5) which examined solutions between 1–15 clusters. From all 250 results, the most frequent solution was selected as the optimal.

Then, Gaussian Mixture Models (GMM) were fitted to the k-means clusters using Expectation-Maximization (EM) method (Algorithm 6).

For all 89 subjects and 100 ROIs, clustering was performed again, but this time at the group level. It resulted in the construction of a single spectral fingerprint for each brain region accounting for the activity that was common for all the participants (Fig. 5.2). For each ROI, spectral modes from all the subjects were gathered and grouped using the k-means algorithm (maximum 100 iterations, 10 replications, cosine distance), and the GMM algorithm again. All modes were taken into account for further analysis ($\xi = 1$ parameter in Algorithm 7). Some of the obtained spectral fingerprints are shown in Fig. 5.2. Note that for a single spectral fingerprint, durations can yield a cumulative percentage above 100%, which is explained in Notes to Algorithm 7. Shaded regions depict the standard deviation (1σ) of the corresponding spectral mode. For i-th of total F frequencies of interest, standard deviation were estimated as $\sqrt{\Sigma_{i,i}}$, where $\Sigma_{i,i}$ is the i-th diagonal entry of the covariance matrix of the GMM component corresponding to

[30] `https://github.com/ThomasYeoLab/CBIG/tree/master/stable_projects/brain_parcellation/Schaefer2018_LocalGlobal/Parcellations/MNI` (accessed: June 4, 2022)

the given spectral mode.

5.2.4 Finding resting-state spectral geometry

Diffusion embedding [244][245] reduces the dimensionality of the data by decomposing a symmetric positive semi-definite nonnegative data similarity matrix into a family of vectors called *diffusion maps* embedded in a new vector space. Transformed data can form shapes that provide useful information about the relationship between observations in the studied dataset.

To determine the distance between spectral fingerprints, accounting for their mathematical structure (Eq. A.21), we used Wasserstein's distance formula, also known as the Earth Mover's Distance (EMD) [229]. EMD naturally extends the notion of a distance between single elements to that of a distance between sets or distributions of elements. The spectral fingerprint of a single region can be viewed as a *signature*, i.e., a set of feature clusters defined by Rubner in [229] described by Eq. A.21. Each pair of two regions being compared, l_1, l_2, are represented by their fingerprints:

$$SF_{l_1} = \{(m_1, t_1), (m_2, t_2), \ldots, (m_{k_1}, t_{k_1})\}$$

$$SF_{l_2} = \{(p_1, \tau_1), (p_2, \tau_2), \ldots, (p_{k_2}, \tau_{k_2})\}$$

thus EMD between two fingerprints is defined as:

$$\mathrm{EMD}(SF_{l_1}, SF_{l_2}) = \frac{\sum_{i=1}^{k_1} \sum_{j=1}^{k_2} d_{ij} f_{ij}}{\sum_{i=1}^{k_1} \sum_{j=1}^{k_2} f_{ij}} \tag{5.1}$$

where l_1, l_2 are indices of regions being compared, k_1, k_2 are the number of spectral modes, $d_{ij} = d(m_i, p_j)$ is the cosine distance (defined in Notation section of Appendix A) between pair of spectral modes of two compared brain regions, and f_{ij} is the optimal flow computed as a solution of the following optimization problem:

$$\text{minimize} \quad \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} d_{ij} f_{ij}$$

under the following constraints:

$$f_{ij} \geq 0$$

$$\sum_{j=1}^{k_2} f_{ij} \leq t_i \quad 1 \leq i \leq k_1$$

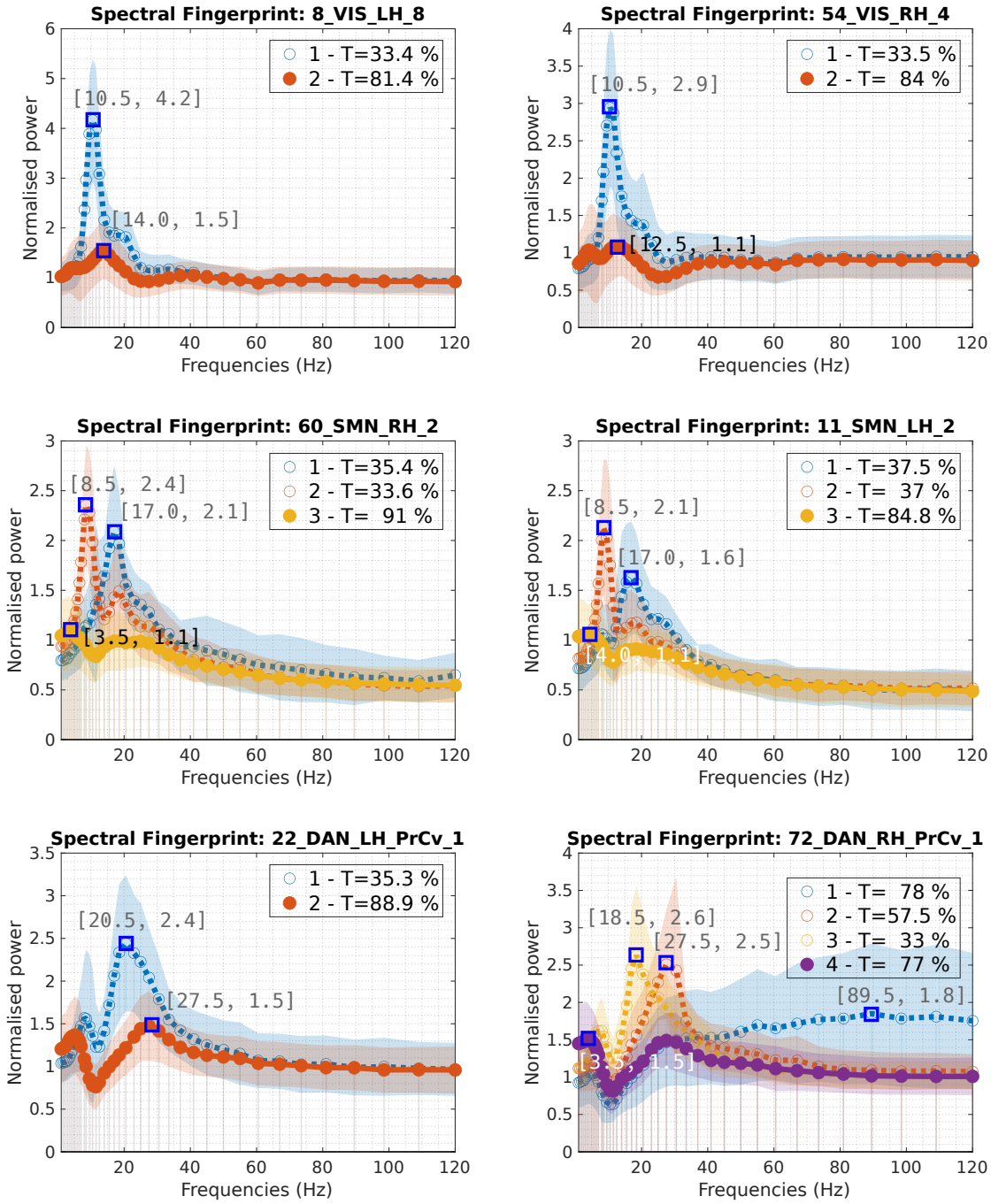


Fig. 5.2. Resting-state spectral fingerprints for Schaefer atlas in the 1–120 Hz frequency interval. Each row shows two homologue brain areas. Legends show the corresponding duration of each spectral mode (i.e., the percentage of trials in which each spectrum was present on average during recording) and whether the mode was present for at least 64 subjects (filled dot) or not (empty dot). Cumulative percentages can exceed 100% for the reasons mentioned in the text. The frequency axis is logarithmic to optimize the lower frequencies resolution. Y-axis depicts the power normalized in relation to the average spectrum of the whole brain. Different modes of operation of a given region are represented by different spectral curves with characteristic frequency peaks (blue boxes).

$$\sum_{i=1}^{k_1} f_{ij} \leq \tau_j \quad 1 \leq j \leq k_2$$

$$\sum_{i=1}^{k_1}\sum_{j=1}^{k_2} f_{ij} = \min\left(\sum_{i=1}^{k_1} t_i \ , \sum_{j=1}^{k_2} \tau_j\right)$$

Known variables are t_i, τ_j, d_{ij}, and the unknown variable is f_{ij}. We additionally normalized t_i and τ_j values to satisfy $\sum_i^{k_1} t_i = 1$ and $\sum_j^{k_2} \tau_j = 1$. This was necessary to compensate for the fact that total spectral modes' duration can exceed 100% and make fingerprints comparable. The equations above correspond to the transportation problem [246][247][248], which EMD originates from. Using EMD allows comparing pairs of sets that contain different numbers of weighted elements. Each spectral mode was an element of a set whose duration was the element's weight. After calculating EMD between each pair of regional spectral fingerprints, dissimilarity values were organized in a matrix form. As a result, EMD $\in \mathbb{R}^{p\times p}$, $p = 100$, dissimilarity matrix was obtained. Then it was normalized using the following formula:

$$\widetilde{\text{EMD}} = \frac{\text{EMD} - \min(\text{EMD})}{\max(\text{EMD}) - \min(\text{EMD})}$$

where min(EMD) and max(EMD) are minimal and maximal entries in the EMD matrix, respectively. Finally, in order to determine the similarity between fingerprints, which served as an input into a diffusion embedding procedure, we defined the fingerprint similarity matrix (Fig. 5.3) as follows:

$$S = 1 - \widetilde{\text{EMD}} \in R^{p\times p} \tag{5.2}$$

The distribution of similarity values for all defined brain networks is depicted in Fig. 5.4.

We have applied the diffusion embedding method with the default parameters ($\alpha = 0.5$) to balance the influence of the data sampling density, following the reasoning presented in the supplement of the work [13]. The implementation[31] used by Margulies [13] was translated to MATLAB code and employed in our study. Given the similarity matrix S, the following steps were executed:

1. Computation of a diagonal matrix $D_{i,i} = \sum_{j=1}^{p} S_{i,j}, \quad D \in \mathbb{R}^{p\times p}$
2. Normalization of the similarity matrix: $\overline{S} = D^{-\alpha} S D^{-\alpha}, \quad \overline{S} \in \mathbb{R}^{p\times p}$,
   where $D^{-\alpha}$ denotes the exponentiation of each matrix element to the power of

[31] `https://github.com/satra/mapalign` (accessed: June 4, 2022)

$-\alpha$,

3. Update of the diagonal matrix: $\overline{D_{i,i}} = \sum_{j=1}^{p} \overline{S_{i,j}}$,

4. Computation of the Markov chain transition matrix: $M = \left(\overline{D}\right)^{-1} \overline{S}$, where $(\overline{D})^{-1}$ is the matrix inverse,

5. Calculation of n largest eigenvalues $\lambda_1, \ldots, \lambda_n$ of M^t and their corresponding eigenvectors $\psi_1, \ldots, \psi_n, \in \mathbb{R}^{1\times p}$, where t is the diffusion time determined automatically (see description below), and $r = 1, \ldots, p$ indexes brain regions.

6. Obtaining the diffusion maps (gradients):

$$\Psi_t = \begin{pmatrix} \lambda_1^t \psi_1 \\ \lambda_2^t \psi_2 \\ \vdots \\ \lambda_k^t \psi_n \end{pmatrix} = \begin{pmatrix} x_1^{(1)} x_1^{(2)} & \ldots & x_1^{(p)} \\ x_2^{(1)} x_2^{(2)} & \ldots & x_2^{(p)} \\ & \vdots & \\ x_k^{(1)} x_k^{(2)} & \ldots & x_k^{(p)} \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{pmatrix}$$

Then, $x_1^{(r)}, \ldots, x_k^{(r)}$ are r-th brain region coordinates in k-dimensional space, $g_1, \ldots, g_k$ are gradients. In our study, the first two diffusion maps, g_1, g_2, explained 90% of the data variance (see Fig. 5.5), so we kept them for further analyses. Explained variance by a given gradient is the percentage of data variance explained by its corresponding eigenvector. The variance of the i-th gradient was calculated as $|\lambda_i| / \sum_i |\lambda_i|$. Following [13], we automated the estimation of the diffusion time using a damped regularization process. This approach was previously tested [249]–[253], and it provides robustness against small noisy eigenvalues by dividing each eigenvalue λ_i by $1 - \lambda_i$, which implies that diffusion times needed to be expressed as $t_i = 1 - log(1 - \lambda_i)/\lambda_i$. FMRI-based gradients obtained by Margulies, used for the further comparisons with our MEG-based gradients, were downloaded from the authors' online Neurovault repository[32], and their values were averaged for each ROI separately across its voxels.

5.2.5 Cortical gradient analysis

Center of the brain was set as an origin of the right-anterior-superior (RAS) coordinate system defined in the FreeSurfer toolbox [254] (center of isotropic 2 mm 128 × 128 × 128 voxel grid volume), according to Schaefer brain regions MNI coordinates[33]. Vector

[32] `https://neurovault.org/collections/1598/` (accessed: June 4, 2022)

[33] `https://github.com/ThomasYeoLab/CBIG/tree/master/stable_projects/brain_parcellation/Schaefer2018_LocalGlobal/Parcellations/MNI/Centroid_coordinates/Schaefer2018_100Parcels_7Networks_order_FSLMNI152_2mm.Centroid_RAS.csv` (accessed: June 4, 2022)

of regional distances was calculated using Euclidean metric:

$$d_{[0,0,0]} = [\sqrt{X_r^2 + Y_r^2 + Z_r^2}]_{r=1,\ldots,p} \in \mathbb{R}^p \tag{5.3}$$

where X_r, Y_r, Z_r are the MNI coordinates of the r-th brain region.

Correlation between two vectors $x = (x_1, \ldots, x_p)$, $y = (y_1, \ldots, y_p)$ was calculated using sample Pearson correlation coefficient:

$$\mathrm{Corr}(x, y) = \frac{1}{p-1} \sum_{r=1}^{p} \left(\frac{x_r - \bar{x}}{s_x} \right) \left(\frac{y_r - \bar{y}}{s_y} \right)$$

where $\bar{x} = \frac{1}{p} \sum_{r=1}^{p} x_r$, $\bar{y} = \frac{1}{p} \sum_{r=1}^{p} y_r$, $s_x = \sqrt{\frac{1}{p-1} \sum_{r=1}^{p} (x_r - \bar{x})^2}$, $s_y = \sqrt{\frac{1}{p-1} \sum_{r=1}^{p} (y_r - \bar{y})^2}$
Correlations calculated for current study: $\mathrm{Corr}(g_1, \pi_1)$, $\mathrm{Corr}(g_1, \pi_2)$, $\mathrm{Corr}(g_2, \pi_1)$, $\mathrm{Corr}(g_2, \pi_2)$, $\mathrm{Corr}(g_1, d_{[0,0,0]})$, $\mathrm{Corr}(g_2, d_{[0,0,0]})$, where

$$g_1, g_2 \in \mathbb{R}^p$$

are our first and second MEG-based gradients obtained from spectral fingerprints' similarity matrix (Eq. 5.2) by diffusion embedding procedure, and

$$\pi_1, \pi_2 \in \mathbb{R}^p$$

denote the first and the second regionally averaged[34] Margulies fMRI-based gradient, respectively.

Relation between g_1, g_2, π_1, π_2, and $d_{[0,0,0]}$ was examined using the forward stepwise regression procedure. It is a method from the multiple regression family in which predictor variables are entered into the model based on the explanatory power quantified by the F-statistic (Eq. 5.4). Predictors are incrementally added to the model. The explanatory power of the updated model is compared with the explanatory power of the previous model via statistical criterion based on the squared sum of errors expressed via F-statistic:

$$F = \frac{SS_0 - SS_1}{df_0 - df_1} \cdot \frac{df_1}{SS_1} \tag{5.4}$$

where $SS_m = \sum_{r=1}^{p} (y_r - \widehat{y}_r)^2$ is the residual sum of squares of a given model, $\widehat{y}_r$ is r-th fitted value related to the empirical observation y_r, df_m denotes the number of a given model's degrees of freedom, and $m \in \{0, 1\}$, where 0 and 1 indexes those quantities as related to the current and updated (with added predictor) model, respectively. Critical

[34] In [13], gradient values were computed for each brain voxel. To match the lengths of π_1 and π_2 to the length of g_1 and $g_2 \in \mathbb{R}^p$, and thus obtain a single fMRI-gradient value representing a given brain area, values were averaged across voxels assigned to the given brain area.

p-values for the significance of the difference between compared models for predictor entrance and removal were $p_{\text{enter}} = 0.05$, and $p_{\text{remove}} = 0.1$, respectively). The portion of explained variance was calculated using the adjusted coefficient of determination:

$$R^2_{\text{adj}} = 1 - (1 - R^2)\frac{(n-1)}{n-k-1}$$

where n - number of observations, k - number of predictors in the model, R^2 is the coefficient of determination defined as:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

where $SS_{\text{res}} = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2$ is the residual sum of squares of the model, $\widehat{y}_i$ is i-th fitted value related to the empirical observation y_i, and $SS_{\text{tot}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$ is the total sum of squares, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

5.3 Results

Distributions of the similarity matrix values (Fig. 5.3 and 5.4) between pairs of resting-state networks [153] show that for the regions from networks related to associative functions, their fingerprints are less similar to fingerprints of other regions (distributions are wider). Two major similarity modules are visible: sensory — VIS, SMN, DAN, and associative — LIM, FPN, DMN. Both modules are linked to the CON network. Visual fingerprints are almost identical to each other (95% of similarity values between VIS fingerprints are greater than 0.87). SMN fingerprints are still very similar to each other, however, not as much as it is for VIS fingerprints (95% of similarity values between SMN fingerprints are greater than 0.80). Networks more on the sensory side tend to have similar fingerprints (tighter distributions, concentrated near 0.8-0.9 similarity values). Regions from more associative areas (DMN, FPN, LIM) share less similarity to regions belonging to the same networks, as well as to the regions from other networks.

Diffusion embedding based on the spectral fingerprint similarity matrix yielded two significant diffusion map vectors. They describe the geometry of the spectral activity of different brain regions (Fig. 5.6) and explain 90% of the data variance (Fig. 5.5), which signifies that most of the mutual relations between fingerprints can be well approximated by just two obtained gradients.

Gradients π_1 and π_2 from [13], shown in the scatter plot of Fig. 5.6, were plotted using the public data[35] accompanying the publication.

[35] `https://neurovault.org/collections/1598/` (accessed: June 4, 2022)

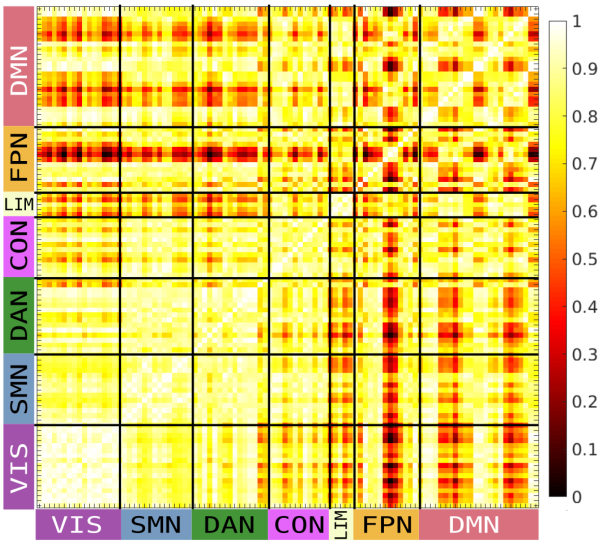


Fig. 5.3. Similarity matrix of resting-state MEG spectral fingerprints between 100 regions outlined in the Schaefer atlas [217]. Similarity values are between 0 (complete dissimilarity) and 1 (identity). Lines depict boundaries between seven functional networks of regions found by Yeo [153]. Abbreviations stand for the following networks: Dorsal Attention Network (DAN), Default Mode Network (DMN), Frontoparietal Network (FPN), Cingulo-opercular Network (CON), Visual Network (VIS), Somatomotor Network (SMN), Limbic Network (LIM). Interpretation in the text.

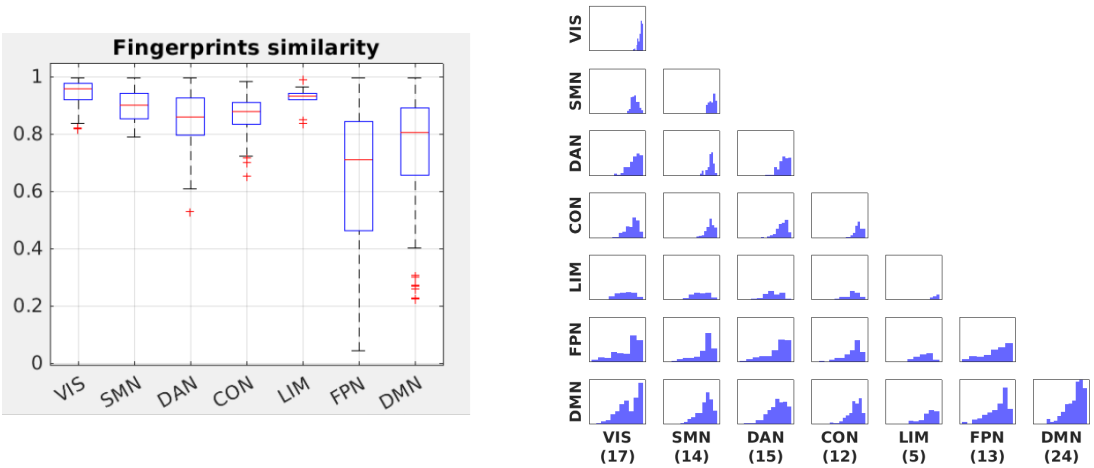


Fig. 5.4. Distributions of similarity values shown in Fig. 5.3. **Left**: Boxplot showing similarity values grouped by networks. **Right**: Distributions of similarity values between pairs of seven functional networks of regions found by Yeo et al., 2011. Diagonal elements of the similarity matrix, which all equal 1, were excluded. X-axes are limited between 0 and 1, and Y-axes are between 0 and 120. The numbers in parentheses show how many ROIs belong to the given network. Networks on the sensory side tend to have similar fingerprints (tighter distributions, concentrated near 0.8-0.9 similarity values).

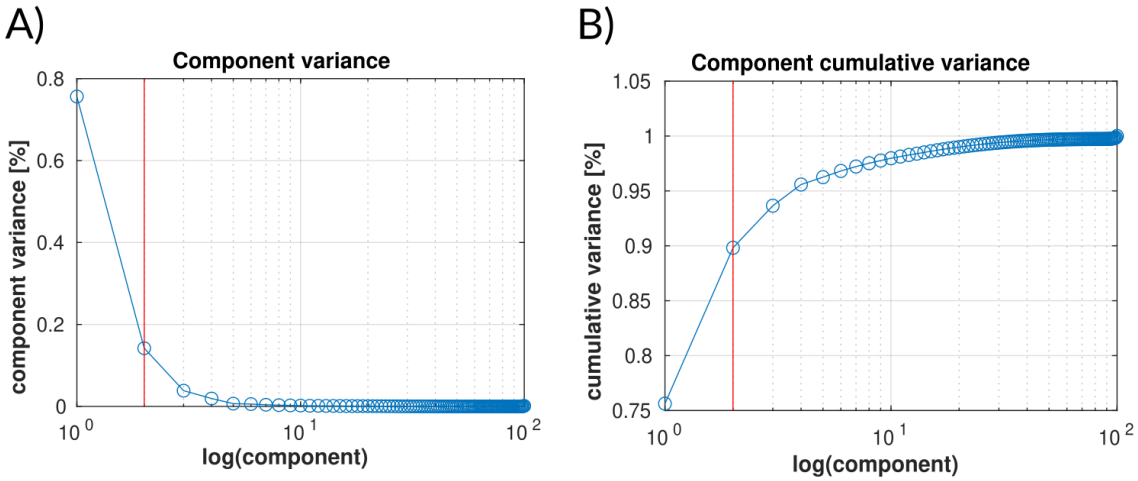


Fig. 5.5. The λ values of the diffusion embedding components. **A)** The λ value of all 100 components of the magnetoencephalographic spectral fingerprints similarity data and **B)** cumulative λ values. The red line marks values at the second component. The first two components explain 90% of the data variance.

The similarity between the triangular shapes of the Margulies gradients and our gradients is visible. Cortical distributions of the gradients are similar as well.

Every of the one hundred regions was assigned to one of the three categories of brain networks (see Fig. 5 in [255]): visual (VIS; blue color), sensory/motor (SMN; green color), transmodal (TRANS; red color). Our gradient nr 2 separates sensory/motor regions and visual regions well, however, there is no clear boundary between transmodal and the rest of the networks (Fig. 5.6).

Gradient values were grouped by seven resting-state networks and depicted using box plots (Fig. 5.7). As concluded in [13], the means of the first fMRI-based gradient (Fig. 5.7A) anchor, at one end, networks serving primary sensory/motor function (SMN, VIS), and at the other end, transmodal networks (DMN, LIM). The first MEG-based gradient (Fig. 5.7B) replicated the order of means for networks only partially — for VIS, DAN, FPN, and DMN. Our MEG-based second gradient is not distributed similarly to the fMRI-based second gradient.

We wanted to check what variables explain g_1, g_2. To this end, we have performed the stepwise regression with the forward selection of the following variables: π_1, π_2, $d_{[0,0,0]}$. In terms of modeling g_1, the first and thus the most important independent predictor was π_1, which explained 27% of the variance ($R^2_{\text{adj}} = 0.265$). In the next regression step, $d_{[0,0,0]}$ was the second independent predictor automatically added to the regression model because it provided the best fit in conjunction with the first predictor. π_2 turned out to be insignificant in terms of explaining g_1, as it has not met the inclusion criterion. In conclusion, g_1 turned out to be related predominantly to π_1

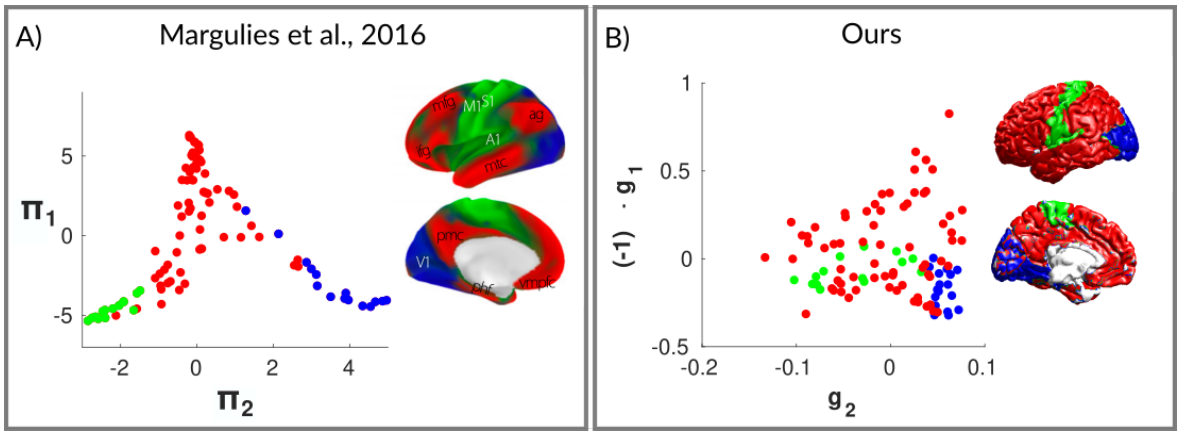


Fig. 5.6. Comparison between fMRI- and MEG-based sensory-associative gradients. **A)** Two gradients π_1 and π_2 obtained by Margulies et al., 2016 on the resting-state fMRI data. Gradients' cortical distribution was copied from [13]. **B)** Gradients g_1 and g_2 obtained from spectral fingerprints based on MEG resting-state data. Note that Y-axis depicts an inverted gradient (hence multiplied by -1) to facilitate the visual assessment of gradients similarity between panels A) and B). Stepwise regression revealed that, indeed, gradient g_1 values from A) and B) are inversely correlated (Table 5.1). Each point represents one of 100 areas from the Schaefer brain atlas [217]. Regions were divided into three functional classes according to [255]: visual (blue), sensory/motor (green), transmodal (red).

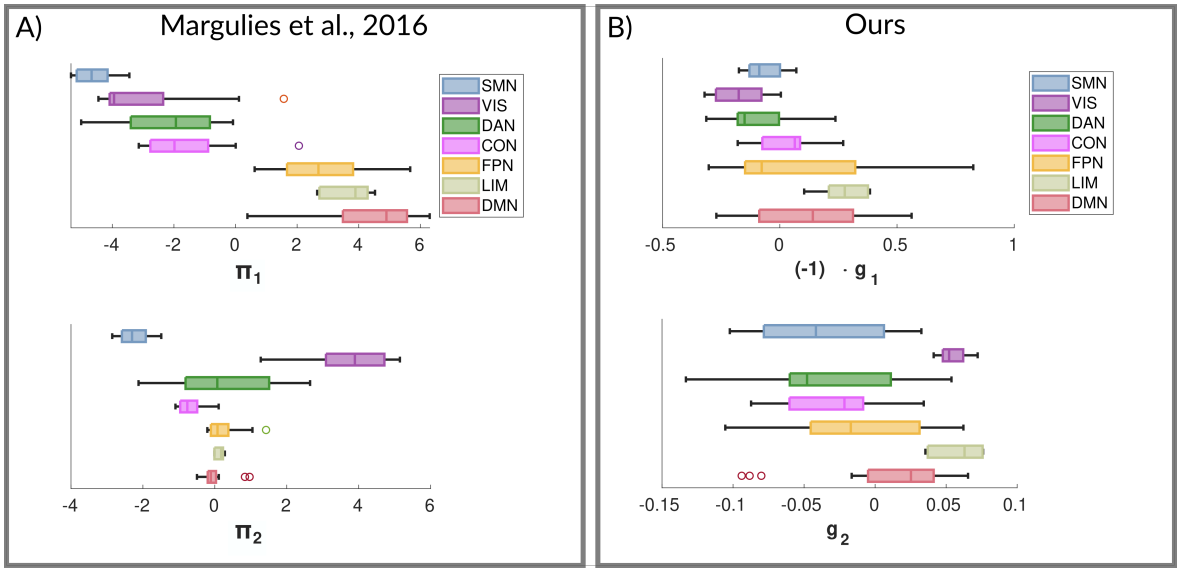


Fig. 5.7. Comparison between distributions of fMRI-based gradients (π_1, π_2) and MEG-based gradients (g_1, g_2). Data were grouped by functional networks. Values were grouped by resting-state functional networks proposed by Yeo [153]: Dorsal Attention Network (DAN), Default Mode Network (DMN), Frontoparietal Network (FPN), Cingulo-opercular Network (CON), Visual Network (VIS), Somatomotor Network (SMN), Limbic Network (LIM). Note that panel B) depicts an inverted g_1 gradient (hence multiplied by -1) to facilitate the visual assessment of gradients similarity between panels A) and B). Stepwise regression revealed that, indeed, gradients g_1 and π_1 are inversely correlated (Table 5.1).

($\beta = -0.414, p < 0.001$) and (to the lesser extent) to $d_{[0,0,0]}$ ($\beta = +0.363, p < 0.001$), explaining 38% of the data variance ($R^2_{\text{adj}} = 0.379$). In terms of modeling g_2, the first and thus the most important independent predictor was π_2, which explained 35% of the variance ($R^2_{\text{adj}} = 0.347$). In the next regression step, π_1 was the second independent predictor automatically added to the regression model because it provided the best fit in conjunction with the first predictor. $d_{[0,0,0]}$ turned out to be insignificant in terms of explaining g_2, as it has not met the inclusion criterion. In conclusion, g_2 turned out to be related predominantly to π_2 ($\beta = +0.607, p < 0.001$) and to the lesser extent to π_1 ($\beta = +0.190, p < 0.05$), explaining 39% of the data variance ($R^2_{\text{adj}} = 0.377$).

By performing stepwise regression, we discovered that our first MEG-based gradient is inversely proportional to the first fMRI-based gradients, i.e.,

$$g_1 \sim (-1) \cdot \pi_1$$

and our second MEG-based gradient

$$g_2 \sim (+1) \cdot \pi_2$$

The result of the stepwise regression is summarized in Table 5.1.

The negative relation between g_1 and π_1, obtained from both correlation analysis and stepwise regression, confirms the intuition behind reversing the axes in Fig. 5.6 and 5.7 related to g_1 for easier comparison.

Table 5.1. Stepwise regression model for explaining MEG-based gradients g_1 and g_2 constructed from resting-state spectral fingerprints. Dependent variables are MEG-based gradients π_1, π_2. Independent variables are: fMRI-based gradients π_1, π_2 proposed by Margulies [13], area distance from the brain center $d_{[0,0,0]}$ defined in Eq. 5.3. Insignificant predictors are marked with '-'.

	Predictors' β			**Model statistics**		
	X1: π_1	X2: π_2	X3: $d_{[0,0,0]}$	p	Explained variance	R^2_{adj}
Y1: g_1	-0.414	-	$+0.363$	< 0.001	38%	0.379
Y2: g_2	$+0.190$	$+0.607$	-	< 0.001	39%	0.377

5.4 Discussion

The hierarchy of oscillations in human brain resting-state activity was explored using the novel approach based on Spectral Fingerprinting [9] and the gradient method proposed in [13]. Regional representations of the high-quality MEG signal from the renowned Human Connectome Project database [200] were generated employing the

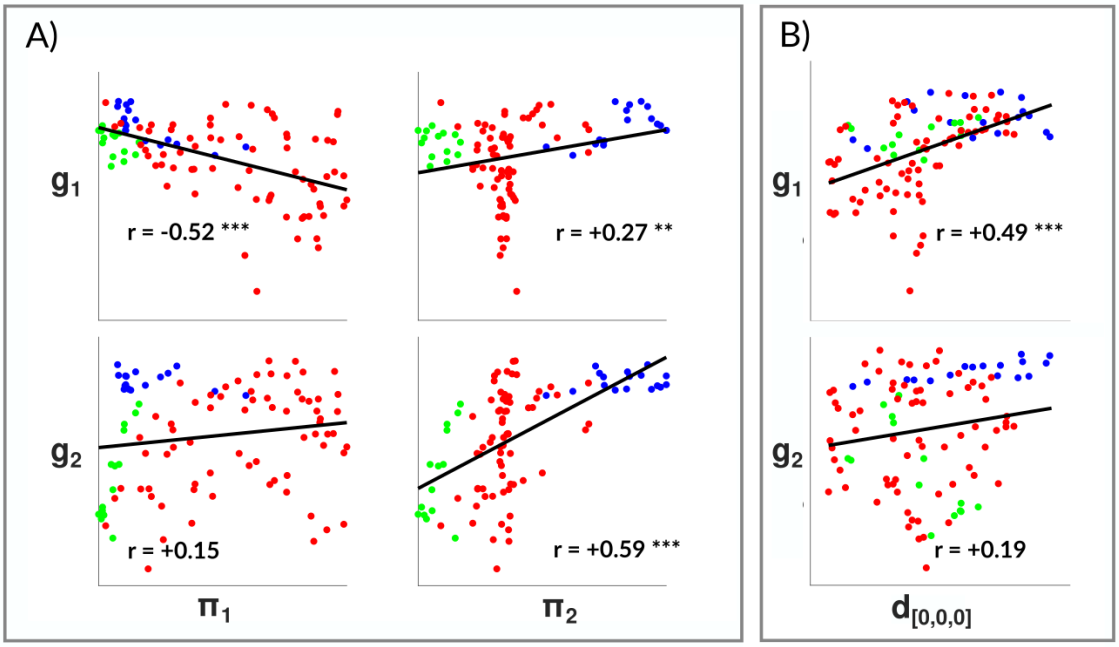


Fig. 5.8. Pearson correlations between fMRI-based gradients (π_1, π_2) and MEG-based gradients (g_1, g_2). Colors correspond to the assignment of each of one hundred brain areas to one of three categories of brain networks by McKeown et al., 2020 (Fig. 5 in [255]: visual (blue), sensory/motor (green), transmodal (red). Legend: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

ToFFi Toolbox — an original software proposed and implemented by the author of this dissertation.

As was described in Section 1.5, specific brain areas in a healthy brain exhibit characteristic rhythms. The spatial distribution of frequency peaks across the cortex in obtained fingerprints agrees with oscillation ranges and locations reported in other studies about spectral hierarchies across the human brain [9][91][92][79]. For example, in Fig. 5.2, fingerprints in the first row refer to occipital areas, usually dominated by the alpha oscillations. Indeed, in the figure, fingerprints peak at 10.5 Hz, 12.5 Hz, and 14 Hz. Fingerprints from the second row (parts of the Somatomotor Network, SMN) refer to motor cortices which are common generators of beta oscillations (12-30 Hz). Peaks in the figure appeared at 3.5 Hz, 4.0 Hz, 8.5 Hz, and 17.0 Hz. Two slowest rhythms may represent delta rhythms sometimes present in the motor system [106], 8.5 Hz oscillation may point to the mu rhythm, and 17 Hz is β_1 oscillation typical for somatomotor areas. Finally, the last row of Fig. 5.2 depicts fingerprints from Dorsal Attention Network (DAN) nodes located in the lateral prefrontal cortex. Frequency peaks located in these regions' fingerprints are: 18.5 Hz, 20.5 Hz, 27.5 Hz (beta-band oscillations), 3.5 Hz and 89.5 Hz (theta-gamma coupling). All of them agree with the natural frequency map presented in [79] and may hint at the working memory processes [94][95] attributed

partially to DAN brain network [256]. The agreement of the regional representations generated with the ToFFi Toolbox confirms the utility of both Spectral Fingerprinting and the presented software in brain studies.

In our study, we have proposed a novel method for calculating the distance between spectral fingerprints based on the Wasserstein measure (Earth Mover's Distance, EMD; [229]). The difficulty of comparing spectral fingerprints lies in the fact that fingerprints from two different regions being compared are not single elements but sets. Each set may contain a different number of spectral modes, and additionally, each mode is weighted by its duration. Previous literature does not propose any solution to measure fingerprint similarity [9][11], which renders our method a novelty.

By using the diffusion embedding method, we have obtained two gradients g_1 and g_2 that explain 90% of the data variance (Fig. 5.5), whereas Margulies reports about 40% explained variance by the first two fMRI-based gradients (π_1, π_2). However, despite more explained variance, g_1 does not provide such a clear sensory-to-associative brain network hierarchy as π_1 (Fig. 5.7). There is some resemblance between gradients that can be seen in the presented scatter plots (Fig. 5.6), which is also suggested by significant correlations (Fig. 5.8A). It would be appropriate in further studies to check how much noise was fitted by the proposed model to control for possible overfitting. Smoothing MEG data or downsampling it to match the slow oscillation of BOLD signal measured with fMRI was successful in recreating large-scale brain networks from EEG data in [156]. Perhaps a similar approach can be used when comparing g and pi gradients. Relations between regions seen through different frequency bands in isolation, as investigated in [159][228], can be informative and serve as validation.

Stepwise regression has shown that π_1 is a moderate predictor inversely related to g_1 and it is also partially related to the regional distance from the brain center $d_{[0,0,0]}$. π_1 is also related to the g_2. Reasons for that are not clearly understood yet, however, Zhang et al., in their study of fMRI resting-state [238], obtained three gradients, with the emphasis on their inter-dependence, which might be visible in our results. Zhang's third discovered component is called the "anatomical centrality" gradient. This means that distance from the brain center might play a role in the functional specialization of different areas. $d_{[0,0,0]}$ turned out to be a significant predictor in our model for explaining the first gradient inversely related to the sensory-associative gradient. Finding other meaningful predictors could help improve the model.

5.5 Limitations

In our study, we have used one of the total three runs of the MEG signal available in the HCP database, as short intervals between the runs may have impacted the signal quality. Overcoming this problem would triple the total length of the data, significantly reducing the variance of fingerprints' spectral modes.

Finding a complex pattern in the data is troublesome using the k-means algorithm[36]. Methods like DBSCAN, Flexscan, or RevDBScan may yield more meaningful clusters [257][258], accounting for probably complex relationships.

Zhang and collaborators proposed the idea of network motifs, which allows a given brain area to be part of more than one functional ensemble, contrary to the typical parcellation approach, where each region can belong to only one network [238]. Designing our study to reflect area engagement in multiple networks simultaneously could be beneficial.

LCMV spatial filtration is a popular but not the most efficient strategy for beamforming. Improvement in the source activity approximation could be achieved by utilizing the MV-PURE filter, which accounts for source correlation and works well in a low signal-to-noise regime [145].

Spectral fingerprinting seeks to find stable patterns of regional brain activity, however, the brain is a dynamic system for which periods of stability and transient behavior are interwoven. The current method does not differentiate between transitions and stable periods of brain activity, which may impair the quality of spectral estimates. Microstate analysis could help to label those transition periods in order to exclude them from further analyses, leading to estimates reflecting stable patterns faithfully [259].

We have used the preprocessed HCP MEG data set, consisting of signal segments that do not form a continuous timeline, i.e., some of the segments were discarded in the preprocessing in order artifacts. Data discontinuity limits the ability to see the complete trajectory of brain dynamics in the phase space. Adjusting the preprocessing to maintain high signal quality while securing its continuity would be beneficial.

Signal segment length was set to 1-second following [9], however, Lubinus et al., 2019 used 200 milliseconds shorter segments to study fingerprints altered by congenital blindness [11]. Exploring fingerprints for different segment lengths may provide a more complete view of brain dynamics and inform decisions at different stages of the study.

For diffusion embedding, parameter alpha was chosen $\alpha = 0.5$ — the default value advised in [13] to balance the influence of the data sampling density. Finding a suitable

[36] An interesting and illustrative comparison between performance of different clustering methods can be found here: `https://scikit-learn.org/0.18/modules/clustering.html` (accessed: June 4, 2022)

criterion for optimizing α would be beneficial.

Finally, the search for meaningful predictors for stepwise regression needs to be reconsidered in order to improve the proposed model.

5.6 Conclusions

The goal of the outlined study was to explore the hierarchy of the oscillations in the human brain under the framework of resting-state networks. We have shown that Spectral Fingerprinting allows replicating the distribution of frequencies across the cortex found in other reports and allows discovering functional components similar to gradients that can be extracted from fMRI-based functional connectivity. Unfortunately, we have not yet established a clear link between frequencies and networks to provide an insightful oscillation hierarchy description. The similarity between fMRI-based gradients and MEG-based gradients suggests that there are some invariant features in the resting brain oscillations hierarchies that can be detected using different methods. Studying those features may lead to the design of useful markers modulated by different diseases, mental states, and cognitive in a systematic way, which is the ultimate goal of brain fingerprinting methods. At the moment, the study is in progress, and more analyses are needed to characterize networks by their spectral content based on obtained fingerprints. Nevertheless, this chapter has shown that ToFFi Toolbox can substantially aid research methodology.

Summary

Neuroinformatics, being the marriage of computer science and neuroscience, is a constantly growing field, which is pinned great hopes on to solve the current burning demands of modern society. The mystery of the human brain constantly inspired many brilliant minds to develop outstanding tools like brain imaging techniques or deep learning, searching for the neural correlates of our everyday experiences. Computer programs are a big part of our advancement in every life area, especially in research.

The main goal of this dissertation was to present the authorial modular, versatile, and open-source MATLAB toolbox to compute area-specific *spectral fingerprints* complying with the definition coined by Keitel and Gross, 2016, and address the lack of such software available for the scientific community. I guided the reader through the basics of neuroanatomy and physics behind the generation of brain's oscillatory activity, which facilitated the understanding of modern brain imaging techniques — magnetoencephalography, electroencephalography, and functional magnetic resonance imaging. These methods were utilized in the body of research presented in Chapter 2, which focuses on the concept and goal of brain fingerprinting. In the context of the thesis goal, spectral approaches play a major role in explaining the inner workings of human cognition. Notably, the third chapter described *Spectral Fingerprinting* being the central technique that I have implemented in the ToFFi Toolbox. With this knowledge, the reader was prepared to dive into Chapter 4 (supplemented with Appendix A), where the toolbox implementation, its functionalities, and utilized algorithms were reviewed in detail. The last chapter illustrated how Spectral Fingerprinting and ToFFi Toolbox could be applied to solve real-life scientific problems. In conclusion, the goal of this dissertation was addressed and hopefully met the audience's expectations.

This thesis has been written from the computer science point of view. However, I am confident that analysis of spectral fingerprints could have many applications as a diagnostic tool that should help to differentiate between various brain disorders, analyze individual differences of brain functions, and be useful in many neuroscientific projects.

I believe that this dissertation, presenting novel Toolbox for Frequency-based Fingerprinting and its applications, proves that the human brain dynamics, being locally specific, can be automatically modeled using spectral features of the MEG/EEG signals and that it is worth the effort.

A Spectral Fingerprinting method in detail

This appendix presents the mathematics behind modules (I), (II), and (III) of the ToFFi Toolbox, organized in the form of algorithm pseudocodes. Table A.1 relates algorithms to ToFFi modules (Chapter 4). As was explained in Chapter 4 (Section 4.2), original group-level identification (Algorithms 8, 9) was upgraded to a more efficient procedure (Algorithms 10, 11). At the end of this appendix, the reader can find a table with the values of parameters used in the application presented in Chapter 5 (Table A.2). Major part of the mathematical notation was developed by dr Krzysztof Rykaczewski.

List of algorithms

Table A.1. ToFFi Toolbox modules and their related algorithms. Spectral Fingerprinting is contained entirely within Algorithm 7, which refers to other algorithms.

Stage name	**Steps in Algorithm 7.**	**Involved algorithms**
Data Preparation (I)	n/a	Alg. 2
Spectral Fingerprinting (II)	1. - 19.	Alg. 1, 2, 3, 4, 5, 6
Stage 1: Source reconstruction	1. - 2.	Alg. 1, 2
Stage 2: Individual Fingerprints	3. - 9.	Alg. 3, 4, 5, 6
Stage 3: Data pooling	10.	n/a
Stage 4: Optimal clusters evaluation	11.	Alg. 5
Stage 5: Spectral Fingerprinting	12. - 18.	Alg. 4, 5
Analysis (III)	n/a	Alg. 10, 11, 12, 13, 14
Stage 6: Group-level identification	n/a	Alg. 10, 11 (originally Alg. 8, 9)
Stage 7: Individual-level identification	n/a	Alg. 12, 13
Stage 8: Network analysis	n/a	Alg. 14

Notation

For a set $\mathcal{A}$, symbol $|\mathcal{A}|$ denotes the number of its elements. For $x \in \mathbb{R}$, by $|x|$ we denote its absolute value. For a vector $x = (x_1, \ldots, x_F) \in \mathbb{R}^F$, $f = 1, \ldots, F$, symbol x_f stands for its f-th coordinate (unless otherwise stated). The vector norm is denoted as $\|x\| = \sqrt{\sum_{f=1}^{F} |x_f|^2}$. The dot product of two vectors, $x, y \in \mathbb{R}^F$, is defined as $x \cdot y = \sum_{f=1}^{F} x_f y_f$. Unless otherwise stated, d denotes the *cosine distance*, i.e., $d(x, y) := 1 - \frac{x \cdot y}{\|x\| \|y\|}$. If it does not lead to confusion, we abbreviate $\mathbb{R}^{1 \times F}$ with $\mathbb{R}^F$. The transpose of a matrix $B \in \mathbb{R}^{p \times q}$ is denoted as $B^\top \in \mathbb{R}^{q \times p}$, and the matrix inverse as B^{-1}.

Common parameters

Here is the list of parameters commonly used in all subsequent algorithms. All other parameters are defined locally in the algorithms' text.

- $M \in \mathbb{N}$ - the number of brain activity sources,
- $L \in \mathbb{N}$ - the number of sensors (can vary from subject to subject),
- $T \in \mathbb{N}$ - the number of signal samples in a single time segment,
- $\mathcal{S}$ - the set of subject indices,
- $p \in \mathbb{N}$ - the number of brain regions of interest,
- $F \in \mathbb{N}$ - the number of frequencies of interest,
- $\rho > 0$ - the volume of a single source voxel (unit: mm^3),
- $\mathsf{LS} \in \mathbb{R}^F$ - the vector of frequencies of interest (usually **l**og-**s**paced).

[*this space intentionally left blank*]

Pseudocodes for Spectral Fingerprinting

Algorithm 1 *Operator $\mathcal{F}$ (spectral power operator)*

1. **Input**: $x \in \mathbb{R}^T$ be the signal, and $\mathsf{LS} \in \mathbb{R}^F$ be vector of frequencies which are interesting for the user.
2. Calculate 3 different FFT for 3 leading (first) DPSS (Slepian) windows: $\mathrm{FFT}_k^f(x)_t = \sum_{t=1}^{T} h_{t,k} x_t e^{-i2\pi f t \Delta t} \in \mathbb{R}$, $k = 1, 2, 3$, $f \in \mathsf{LS}$, where $h_{t,k}$ is the kth Slepian taper, and Δt is the reciprocal of sampling frequency.
3. Calculate power spectrum of each window $|\mathrm{FFT}_k^f(x)_t|^2$, $k = 1, 2, 3$, $t = 1, \ldots, T$, $f \in \mathsf{LS}$.
4. Define $z_t^f := \frac{1}{3} \sum_{k=1}^{3} |\mathrm{FFT}_k^f(x)_t|^2$, $t = 1, \ldots, T$, $f \in \mathsf{LS}$, which is the average power spectrum of signal x.
5. **Output**: $\left\{ (z_t^f \mid f \in \mathsf{LS}) \mid t = 1, \ldots, T \right\} \subset \mathbb{R}^F$.

Notes:

- Following the suggestion from [9], a logarithmic frequency spacing is used to optimize the resolution for lower frequencies. LS stands for *log-spaced*.
- Frequency analysis is done using `ft_freqanalysis` function from the FieldTrip toolbox . It takes as an input a list of frequencies of interest, but the output list of frequencies might be different, e.g., for input list:

  ```
  [ 1.00, 1.10, 1.22, 1.34, ..., 98.70, 108.83, 120.00 ] Hz
  ```

  vector of output frequencies is:

  ```
  [ 1.00, 1.50, 2.00, 2.50, ..., 98.50, 109.00, 120.00 ] Hz
  ```

  and by LS we denote this output vector of frequencies. The difference is a result of the rounding and quantization performed by FieldTrip to match the desired frequency resolution. To control this behavior, one can increase the padding length (`cfg.pad` argument for `ft_freqanalysis` function used in ToFFi's `fourierAnalysis_Method1.m` file), so the frequency axis will be more granular, and thus rounding will not produce repeats that are removed by mentioned quantization.
- We use a multitaper approach with Slepian windows (Discrete Prolate Spheroidal Sequences, DPSS) [214] for spectra estimation as it is more robust to noise, increases the signal-to-noise ratio of high-frequency activity, and is advised when analyzing a single-trial (segmented) data [195].

Algorithm 2 *Spatial filtering with EVD approach*

1. **Input**: Symmetric data covariance matrix $R \in \mathbb{R}^{L\times L}$, and lead field matrix $H \in \mathbb{R}^{L\times(3M)}$.
2. We assume linear spatial filtering model, i.e.,

$$y = Hx + \eta, \tag{A.1}$$

   where $\eta \in \mathbb{R}^{L\times T}$ is a zero-mean random vector representing noise which is uncorrelated with the signal of interest $x \in \mathbb{R}^{(3M)\times T}$ which is also zero-mean, $y \in \mathbb{R}^{L\times T}$ is the observed (MEG or EEG) signal. In particular, it implies that $\mathbb{E}(y) = 0$.
3. Let H_i stands for ith column of matrix H, $i = 1, \ldots, 3M$. Define

$$h_m := [H_{3m-2}, H_{3m-1}, H_{3m}] \in \mathbb{R}^{L\times 3}, \qquad m = 1, \ldots, M, \tag{A.2}$$

   which is vector field generated by the brain tissue volume ρ at point m.
4. Since matrix R is symmetric, we can apply EVD decomposition and obtain factorization $h_m^\top R^{-1} h_m = U(m)\Sigma(m)U(m)^\top \in \mathbb{R}^{3\times 3}$, where $U(m) \in \mathbb{R}^{3\times 3}$, $\Sigma(m) \in \mathbb{R}^{3\times 3}$, $m = 1, \ldots, M$.
5. Let $\eta_m := U_l(m) \in \mathbb{R}^{3\times 1}$ be this column of matrix $U(m)$, for which $\Sigma_{ll}(m) \geq \Sigma_{jj}(m)$, $j = 1, 2, 3$ (column with the dominant eigenvalue), $m = 1, \ldots, M$.
6. Define $h_{m,\text{EVD}} := h_m \cdot \eta_m \in \mathbb{R}^{L\times 1}$ and

$$w_m := \left(h_{m,\text{EVD}}^\top R^{-1} h_{m,\text{EVD}}\right)^{-1} h_{m,\text{EVD}}^\top R^{-1} \in \mathbb{R}^{1\times L}, \tag{A.3}$$

   for $m = 1, \ldots, M$, is the spatial filter that reconstructs neural activity at mth source.
7. Put

$$W_{\text{LCMV,EVD}} := \begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix} \in \mathbb{R}^{M\times L} \tag{A.4}$$

8. **Output**: $W_{\text{LCMV,EVD}} \in \mathbb{R}^{M\times L}$ is the spatial filter that reconstructs neural activity.

Notes:

- The number of sensors L can vary from subject to subject.
- x is a source activity signal to be reconstructed. h_m is a lead field matrix that describes how the unit activity of the mth source propagates onto the array of (MEG/EEG) sensors. w_m is a spatial filter vector reconstructing mth source activity (x_m), according to the formula $x_m \approx \hat{x}_m = w_m y \in \mathbb{R}^T$.
- Presented source reconstruction method is known as LCMV beamforming [142].
- Matrix H has $3M$ columns. Every three columns correspond to the three canonical directions of a single source signal propagation.

Algorithm 3 *Data preprocessing*

1. **Input**:
   (a) Segmented sensor signal $\{y^s(n) \mid n \in \mathcal{N}_s\} \subset \mathbb{R}^{L_s \times T}$, where $\mathcal{N}_s \subset \mathbb{N}$ is a set of indices of time segments, and subject $s \in \mathcal{S}$. We assume that data is detrended, i.e., $\mathbb{E}\big(y^s(n)\big) = 0$, $n \in \mathcal{N}_s$, $s \in \mathcal{S}$.
   (b) Lead field matrices H^s, $s \in \mathcal{S}$.
   (c) $\big\{\mathcal{R}_1, \ldots, \mathcal{R}_p\big\}$ partition of set of sources into *regions of interest*, i.e., $\mathcal{R}_1 \cup \ldots \cup \mathcal{R}_p = \{1, \ldots, M\}$ and $\mathcal{R}_i \cap \mathcal{R}_j = \varnothing$, $i \neq j$.
2. Let $R^s := \frac{1}{N_s(T-1)} \sum_{n=1}^{N_s} \mathbb{E}\Big(y^s(n) y^s(n)^\top\Big)$ be the average data covariance matrix, where $N_s := |\mathcal{N}_s|$, $s \in \mathcal{S}$.
3. $W^s \in \mathbb{R}^{M \times L_s}$ are spatial filters specific for each subject $s \in \mathcal{S}$ obtained from Algorithm 2 using lead field matrices H^s and R^s be the covariance matrix estimate, where $s \in \mathcal{S}$.
4. Source reconstruction: let
$$q_m^s(n) := w_m^s y^s(n) \in \mathbb{R}^{1 \times T} \tag{A.5}$$
be the reconstructed signal of mth source, where $w_m^s \in \mathbb{R}^{1 \times L_s}$ be mth row of W^s, $m = 1, \ldots, M$, $n \in \mathcal{N}_s$, $s \in \mathcal{S}$.
5. Source power: let $e_m^s(n) := \mathcal{F}\big(q_m^s(n), \mathsf{LS}\big) \in \mathbb{R}^{1 \times F}$, be a vector of mth source power for each frequency $f = 1, \ldots, F$, where $m = 1, \ldots, M$, $n \in \mathcal{N}_s$, $s \in \mathcal{S}$, and $\mathcal{F}$ is an operator defined in Algorithm 1.
6. Power normalization: let $\varepsilon_f^s := \frac{1}{N_s M} \sum_{n=1}^{N_s} \sum_{m=1}^{M} e_m^s(n)_f \in \mathbb{R}$, where $s \in \mathcal{S}$, $f = 1, \ldots, F$. Therefore, $\varepsilon^s \in \mathbb{R}^{1 \times F}$. Think of this as of global average power spectrum.
7. Let $z_m^s(n) := \big[e_m^s(n)_f / \varepsilon_f^s\big]_{f=1,\ldots,F} \in \mathbb{R}^{1 \times F}$ be a source power spectrum normalized over whole brain, where $m = 1, \ldots, M$, $n \in \mathcal{N}_s$, $s \in \mathcal{S}$.
8. For $s \in \mathcal{S}$ and for $l = 1$ to p:
   (a) Let $x_l^s(n)_f := \frac{1}{|\mathcal{R}_l|} \sum_{m \in \mathcal{R}_l} z_m^s(n)_f \in \mathbb{R}$ be a power spectrum of lth region of interest for subject s, where $f = 1, \ldots, F$, $n \in \mathcal{N}_s$. Therefore, $x_l^s(n) \in \mathbb{R}^{1 \times F}$.
   (b) Let $\xi_l^s(n) := \frac{1}{F} \sum_{f=1}^{F} x_l^s(n)_f \in \mathbb{R}$.
   (c) Let $\mu_l^s := \frac{1}{N_s} \sum_{n=1}^{N_s} \xi_l^s(n)$ and $\sigma_l^s := \frac{1}{N_s - 1} \sqrt{\sum_{n=1}^{N_s} \big(\xi_l^s(n) - \mu_l^s\big)^2}$ be the mean and the variance of the energy of lth region of interest for subject s.
   (d) Z-score rejection criterion: let $\mathcal{N}_s' := \Big\{ n \in \mathcal{N}_s \quad | \quad |\frac{\xi_l^s(n) - \mu_l^s}{\sigma_l^s}| \leq \Gamma \Big\}$ is a set of indices of time segments for data with outliers removed.
9. **Output**: $\big\{x_l^s(n) \mid n \in \mathcal{N}_s'\big\} \subset \mathbb{R}^{1 \times F} = \mathbb{R}^F$, $l = 1, \ldots, p$, and $n \in \mathcal{N}_s'$.

Notes:

- In Step 1a, note that $\mathcal{N}_s$ indexes time segments and is not necessarily "continuous", e.g., $\mathcal{N}_s = \{1, 4, 5, 6, 10, 11\}$. Some segments might be missing due to possible external preprocessing routines applied to the dataset.
- Γ is the z-score threshold value for outlier rejection in Step 8d.

Algorithm 4 *k-means clustering algorithm with replication*

1. **Input**: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^F$, $k \in \mathbb{N}$ and $R \in \mathbb{N}$.
2. For each replication $r = 1$ to R:
   (a) Initialize(*) cluster centroids $\mu_1^r, \ldots, \mu_k^r \in \mathbb{R}^F$ randomly.
   (b) While the stopping criterion(**) has not been met:
      i. (Assignment to clusters) For $i = 1$ to n:

$$c_i^r := \operatorname{argmin}_{j \in \{1,\ldots,k\}} d\big(x_i, \mu_j^r\big), \tag{A.6}$$

      where $d\colon \mathbb{R}^F \times \mathbb{R}^F \to [0, +\infty)$ is the cosine distance(***).
      ii. (Clusters update) For $j = 1$ to k:

$$\mu_j^r := \frac{\sum_{i=1}^n \delta_{c_i^r, j} x_i}{\sum_{i=1}^n \delta_{c_i^r, j}}, \tag{A.7}$$

      where $\delta_{a,b}$ denotes the Kronecker delta.
3. Find the clustering that minimizes the objective function given by the total sum of distances

$$\mathcal{J}(r) := \sum_{j=1}^{k} \sum_{i=1}^{n} \delta_{c_i^r, j} d\big(x_i, \mu_j^r\big)^2. \tag{A.8}$$

4. For optimal $\overline{r} = \operatorname{argmin}_r \mathcal{J}(r)$ define $\mathcal{C}_j^{\overline{r}} := \big\{x \in X \mid j = \operatorname{argmin}_{h \in \{1,\ldots,k\}} d\big(x, \mu_h^{\overline{r}}\big)\big\}$, $j = 1, \ldots, k$. Therefore, each x_i, $i = 1, \ldots, n$ belongs to one and only one $\mathcal{C}_j^{\overline{r}}$.
5. **Output**: Centroids $\mu_1^{\overline{r}}, \ldots, \mu_k^{\overline{r}} \in \mathbb{R}^F$, family of sets $\{\mathcal{C}_1^{\overline{r}}, \ldots, \mathcal{C}_k^{\overline{r}}\}$, $\mathcal{C}_h^{\overline{r}} \subset \mathbb{R}^F$, $h = 1, \ldots, k$ indexes obtained clusters.

Notes:

- Here, R is the number of k-means replications, X is a set of normalized power spectra of single segments — the output of Algorithm 3.
- (*) MATLAB routine `kmeans` is used in its default configuration, which provides initial clusters using `kmeans++` algorithm (MATLAB Documentation: `https://www.mathworks.com/help/stats/kmeans.html#bueftl4-1`).
- (**) the default stopping criterion is used: current replication of k-means is finished when cluster assignment is the same as for the previous iteration in that replication OR algorithm finished 100 of iterations in current replicate.
- (***) it was suggested by [9], where authors claim that such a metric is more sensitive to shape and less to data amplitude.
- To learn about k-means read [260].

Algorithm 5 *Selection of optimal k for k-means clustering algorithm*

1. **Input**: $\left\{\left\{\{\mathcal{C}_j^{k,e}\}_{j=1}^{k}\right\}_{k=1}^{K}\right\}_{e=1}^{E}$ a set of k-means clusterings for $k = 1, \ldots, K$, calculated E times, i.e., $\bigcup_{j=1}^{k} \mathcal{C}_j^{k,e} = X$ and $\mathcal{C}_i^{k,e} \cap \mathcal{C}_j^{k,e} = \varnothing$, $i \neq j$ for each $k = 1, \ldots, K$.
2. For $e = 1$ to E iterations:
   (a) For $k = 1$ to K:
      i. For each data point $x \in \mathcal{C}_i^{k,e}$, let

$$a^{k,e}(x) := \frac{1}{|\mathcal{C}_i^{k,e}| - 1} \sum_{y \in \mathcal{C}_i^{k,e}, x \neq y} d(x, y),$$
$$b^{k,e}(x) := \min_{j \neq i} \frac{1}{|\mathcal{C}_j^{k,e}|} \sum_{y \in \mathcal{C}_j^{k,e}, x \neq y} d(x, y).$$

      ii. Define the *silhouette* of each data point $x \in \mathcal{C}_i^{k,e}$

$$s^{k,e}(x) := \begin{cases} \frac{b^{k,e}(x) - a^{k,e}(x)}{\max\{a^{k,e}(x), b^{k,e}(x)\}}, & \text{if} \quad |\mathcal{C}_i^{k,e}| > 1, \\ 0 & \text{otherwise.} \end{cases} \tag{A.9}$$

      iii. Define the average silhouette

$$\mathfrak{S}^{k,e} := \frac{\sum_{x \in X} s^{k,e}(x)}{|X|}. \tag{A.10}$$

3. Define $\overline{k} := \text{mode}_{k=1,\ldots,K; e=1,\ldots,E}\, \mathfrak{S}^{k,e}$.
4. **Output**: $\overline{k}$.

Notes:

- Here, optimal cluster evaluation using the Silhouette criterion is presented [261], however, other criteria listed in MATLAB Documentation (`https://www.mathworks.com/help/stats/evalclusters.html#shared-criterion`) can be used instead.
- Algorithm 5 running E times may return slightly different clusters, but practice suggests that the variance of results will be small. Nevertheless, the influence of this effect on the distribution of the data is taken into account by taking the mode (the most frequent) of obtained results.
- In 3, originally, the solution with maximum silhouette value was chosen (Appendix B, line 275 in Listing 10). ToFFi chooses the most frequent solution instead. The change prevents accidentally choosing some degenerated solution that might result from odd random k-means initiation.

Algorithm 6 *GMM clustering algorithm*

1. **Input**: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^F$, $k \in \mathbb{N}$ and initial $\mu_j \in \mathbb{R}^F$, $j = 1, \ldots, k$.
2. Define Gaussian component as

$$\phi(x \mid \mu, \Sigma) := \frac{1}{(2\pi)^{F/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right), \tag{A.11}$$

   where $\mu \in \mathbb{R}^F$, $\Sigma \in \mathbb{R}^{F \times F}$.
3. Let $q := 0$.
4. Choose component proportions $\alpha_j^{(q)} \in \mathbb{R}$ such that $\sum_{j=1}^k \alpha_j^{(q)} = 1$, diagonal $\Sigma_j^{(q)} \in \mathbb{R}^{F \times F}$, and set $\mu_j^{(q)} := \mu_j$, $j = 1, \ldots, k$, and compute the initial log-likelihood

$$\ell^{(q)} := \sum_{i=1}^n \log\left(\sum_{j=1}^k \alpha_j^{(q)} \phi\left(x_i \mid \mu_j^{(q)}, \Sigma_j^{(q)}\right)\right). \tag{A.12}$$

5. While stopping criterion(*) has not been met:
   (a) (**E-step**) For $j = 1, \ldots, k$ compute the posterior probability

$$w_{i,j}^{(q)} := \frac{\alpha_j^{(q)} \phi(x_i \mid \mu_j^{(q)}, \Sigma_j^{(q)})}{\sum_{l=1}^k \alpha_l^{(q)} \phi\big(x_i \mid \mu_l^{(q)}, \Sigma_l^{(q)}\big)}, \tag{A.13}$$

   $i = 1, \ldots, n$, and

$$n_j^{(q)} := \sum_{i=1}^n w_{i,j}^{(q)}. \tag{A.14}$$

   (b) (**M-step**) For $j = 1, \ldots, k$ compute new estimates for

$$\begin{aligned}
\text{component proportion:} \quad & \alpha_j^{(q+1)} := \frac{n_j^{(q)}}{n}, \\
\text{component expected value:} \quad & \mu_j^{(q+1)} := \frac{1}{n_j^{(q)}} \sum_{i=1}^n w_{i,j}^{(q)} x_i, \\
\text{component covariance matrix:} \quad & \Sigma_j^{(q+1)} := \frac{1}{n_j^{(q)}} \sum_{i=1}^n w_{i,j}^{(q)} \left(x_i - \mu_j^{(q+1)}\right)\left(x_i - \mu_j^{(q+1)}\right)^\top.
\end{aligned}$$

   Notice that $\Sigma_j^{(q)}$ for $q > 0$ does not need be diagonal.
   (c) Compute the new log-likelihood using Equation (A.12) with (q) replaced by $(q+1)$.
   (d) $q \leftarrow q + 1$.
6. **Output**:

$$\mathsf{GMM} := \big((\mu_1^{(q)}, \ldots, \mu_k^{(q)}), (\Sigma_1^{(q)}, \ldots, \Sigma_k^{(q)}), (\alpha_1^{(q)}, \ldots, \alpha_k^{(q)})\big),$$

   for some $q \in \mathbb{N}$ (for which algorithm has stopped).

Notes:

- GMM stands for Gaussian Mixture Models [262].
- GMM modeling follows k-means clustering, as in [9].
- X is a set of normalized power spectra of single segments — the output of Algorithm 3.
- Presented procedure is known in the literature as Expectation-Maximization (EM) algorithm [262].
- Regularization is done in its typical form. Before the EM algorithm start, a regularization parameter $\lambda \in \mathbb{R}^+$ is set. Then at the end of each iteration of the EM algorithm, λ is added to every diagonal element of the covariance matrix, i.e., $\Sigma_j^{(q+1)} := \Sigma_j^{(q+1)} + \lambda \mathbb{I}_F$, where $\mathbb{I}_F$ stands for the $F \times F$ identity matrix (we use diagonal loading to ensure well conditioning of covariance matrices). The EM algorithm is implemented in `fitgmdist` MATLAB function. It calls `gmdistribution.fit`, which calls `gmcluster`, which further calls `gmcluster_learn` function, where regularization is implemented (track `regVal` variable).
- (*) default stopping criterion implemented in `fitgmdist` function is used: algorithm finished 100 iterations OR termination tolerance for the log-likelihood function is reached, i.e., $\Delta\ell \geq 0 \quad \text{and} \quad \Delta\ell < |\ell^{(q)}| \cdot 10^{-6}$,
where $\Delta\ell := \ell^{(q)} - \ell^{(q-1)}$.

[*this space intentionally left blank*]

Algorithm 7 *Whole Spectral Fingerprinting — Part 1*

1. **Input**:
   (a) $V_s := \{y^s(n) \mid n \in \mathcal{N}_s\} \subset \mathbb{R}^{L_s \times T}$, where $\mathcal{N}_s \subset \mathbb{N}$ is a set of indices of signal segments, and $s \in \mathcal{S}$ a set of subjects,
   (b) Lead field matrices H^s, $s \in \mathcal{S}$.
   (c) $\{\mathcal{R}_1, \ldots, \mathcal{R}_p\}$ partition of set of sources into regions of interest, i.e., $\mathcal{R}_1 \cup \ldots \cup \mathcal{R}_p = \{1, \ldots, M\}$ and $\mathcal{R}_i \cap \mathcal{R}_j = \varnothing$, $i \neq j$.
   (d) $E_{\mathsf{IF}}, E_{\mathsf{SF}} \in \mathbb{N}$ parameters indicating the number of repeats in clustering evaluation.
2. After applying data preprocessing step (Algorithm 3) we obtain

$$\mathsf{X}_l^s := \{x_l^s(n) \mid n \in \mathcal{N}_s'\} \subset \mathbb{R}^F, \qquad l = 1, \ldots, p, \quad s \in \mathcal{S},$$
$$\mathsf{X}_l := \bigcup_{s \in \mathcal{S}} \mathsf{X}_l^s, \qquad l = 1, \ldots, p.$$

3. The number of clusters $\overline{k}_{s,l}$ can be chosen by one of the criteria below:
   (a) fixed: $\overline{k}_{s,l} := const$,
   (b) optimal: perform Algorithm 4 on $\mathsf{X}_l^s \subset \mathbb{R}^F$ with $R := 5$ for $k = 1, \ldots, K$, $l = 1, \ldots, p$. After repeating this operation $E := E_{\mathsf{IF}}$ times we obtain family of k-means clusterings $\big\{\big\{\{\mathcal{C}_j^{s,l,k,e}\}_{j=1}^k\big\}_{k=1}^K\big\}_{e=1}^E$. Then $\overline{k}_{s,l}$ is obtained from Algorithm 5.
4. On set $\mathsf{X}_l^s \subset \mathbb{R}^F$ apply k-means Algorithm 4 with R replications and $k := \overline{k}_{s,l}$ to obtain centroids $\mu_1^{s,l}, \ldots, \mu_k^{s,l} \in \mathbb{R}^F$, $l = 1, \ldots, p$, $s \in \mathcal{S}$.
5. Use data $\mathsf{X}_l^s \subset \mathbb{R}^F$, $l = 1, \ldots, p$, $s \in \mathcal{S}$, initial centroids $\mu_1^{s,l}, \ldots, \mu_k^{s,l} \in \mathbb{R}^F$, and apply Algorithm 6 to obtain models defined by their expectations (μ), covariances (Σ), and proportions (α):

$$\mathsf{GMM}_l^s := \big((\mu_1^{s,l}, \ldots, \mu_k^{s,l}), (\Sigma_1^{s,l}, \ldots, \Sigma_k^{s,l}), (\alpha_1^{s,l}, \ldots, \alpha_k^{s,l})\big), \tag{A.15}$$

   where $l = 1, \ldots, p$, $s \in \mathcal{S}$. GMM is fed with the centroids obtained by k-means in order to increase the speed of convergence of Algorithm 6. Note that the same notation is used to mark the initial and output centroids not to complicate subsequent symbols.
6. Compute the posterior probability for given $x \in \mathsf{X}_l^s$ to be an element of jth cluster as

$$w_j^{s,l}(x) := \frac{\alpha_j^{s,l}\phi(x \mid \mu_j^{s,l}, \Sigma_j^{s,l})}{\sum_{i=1}^k \alpha_i^{s,l}\phi\big(x \mid \mu_i^{s,l}, \Sigma_i^{s,l}\big)}, \quad j = 1, \ldots, k, \quad l = 1, \ldots, p, \quad s \in \mathcal{S}, \tag{A.16}$$

   where function ϕ is defined in Equation (A.11).
7. Define set of clusters by $\mathcal{C}_i^{s,l} := \{x \in \mathsf{X}_l^s \mid i = \operatorname{argmax}_j w_j^{s,l}(x)\}$, $i = 1, \ldots, k$, $l = 1, \ldots, p$, $s \in \mathcal{S}$.
8. Define so-called *centroid durations* $T_j^{s,l} := \frac{|\mathcal{C}_j^{s,l}|}{\sum_{i=1}^k |\mathcal{C}_i^{s,l}|}$, $j = 1, \ldots, k$, $l = 1, \ldots, p$, $s \in \mathcal{S}$.
9. Define **Individual Fingerprint** IF_l^s for specific region of interest and for a specific subject by the following

$$\mathsf{IF}_l^s := \big(\mu_h^{s,l}, \Sigma_h^{s,l}, \alpha_h^{s,l}, T_h^{s,l}\big)_{h=1}^k, \tag{A.17}$$

   where $l = 1, \ldots, p$, $s \in \mathcal{S}$, k is the number of clusters.

Algorithm 7 *Whole Spectral Fingerprinting — Part 2*

10. Define the *set of pooled clusters* by the following

$$\begin{aligned}\mathsf{PC}_l^s &:= \{\mu_h^{s,l}\}_{h=1}^k, \qquad l = 1,\dots,p, \quad s \in \mathcal{S},\\ \mathsf{PC}_l &:= \bigcup_{s\in\mathcal{S}} \mathsf{PC}_l^s, \qquad l = 1,\dots,p.\end{aligned}$$

11. Number of clusters $\overline{k}_l$ can be chosen by one of the criteria below:
    (a) fixed: $\overline{k}_l := const$,
    (b) optimal: perform k-means (Algorithm 4) on PC_l with $R := 5$ for $k = 1,\dots,K$, $l = 1,\dots,p$. After repeating this operation $E := E_{\mathsf{SF}}$ times we obtain family of k-means clusterings $\big\{\big\{\{\mathcal{C}_j^{l,k,e}\}_{j=1}^k\big\}_{k=1}^K\big\}_{e=1}^E$. Then $\overline{k}_l$ is obtained from Algorithm 5.
12. For each region of interest perform $\overline{k}_l$-means clustering (Algorithm 4) on PC_l, $l = 1,\dots,p$ and obtain optimal centroids $\varpi_1^l,\dots,\varpi_{\overline{k}_l}^l \in \mathbb{R}^F$.
13. For PC_l, $k := \overline{k}_l$ and initial centroids $\varpi_1^l,\dots,\varpi_{\overline{k}_l}^l \in \mathbb{R}^F$, $l = 1,\dots,p$, perform Algorithm 6 and obtain
$$\mathsf{GMM}_l := \big((\varpi_1^l,\dots,\varpi_{\overline{k}_l}^l),(\Sigma_1^l,\dots,\Sigma_{\overline{k}_l}^l),(\alpha_1^l,\dots,\alpha_{\overline{k}_l}^l)\big), \tag{A.18}$$
where $l = 1,\dots,p$. Note that here the same notation is used to mark the initial and output centroids not to complicate subsequent symbols.
14. Compute the posterior probability for given $x \in \mathsf{X}_l$ to be an element of jth cluster as
$$w_j^l(x) := \frac{\alpha_j^l \phi(x \mid \varpi_j^l, \Sigma_j^l)}{\sum_{i=1}^{\overline{k}_l} \alpha_i^l \phi\big(x \mid \varpi_i^l, \Sigma_i^l\big)}, \quad j = 1,\dots,\overline{k}_l, \quad l = 1,\dots,p. \tag{A.19}$$
15. Define the set of clusters by $\mathcal{C}_i^l := \{x \in \mathsf{X}_l \mid i = \operatorname{argmax}_j w_j^l(x)\}$, $i = 1,\dots,\overline{k}_l$, $l = 1,\dots,p$.
16. Define the set of subjects contributing to each cluster set by $\mathscr{S}(\mathcal{C}_i^l) := \{s \in \mathcal{S} \mid \mu_i^{s,l} \in \mathcal{C}_i^l\}$, $i = 1,\dots,\overline{k}_l$. Let $\mathsf{S}_i^l := |\mathscr{S}(\mathcal{C}_i^l)|$. Symbol $\mu_j^{s,l}$ represent output GMM centroids found in Step 5 of this Algorithm.
17. Let $J_i^{s,l} := \big\{j \in \{1,\dots,k_{IF}\} \mid \mu_j^{s,l} \in \mathcal{C}_i^l\big\}$ and $D_i^l := \dfrac{\sum_{s\in\mathcal{S}} \sum_{j\in J_i^{s,l}} T_j^{s,l}}{\mathsf{S}_i^l}$, $i = 1,\dots,\overline{k}_l$, $l = 1,\dots,p$. Definition of $T_j^{s,l}$ can be found in Step 8 of this Algorithm.
18. Define **Spectral Fingerprint** SF_l for specific region of interest by the following
$$\mathsf{SF}_l := \big(\varpi_i^l, \Sigma_i^l, \beta_i^l, D_i^l, \mathsf{S}_i^l\big)_{i\in I^l}, \qquad l = 1,\dots,p, \tag{A.20}$$
where $I^l := \big\{i \in \{1,\dots,\overline{k}_l\} \mid \mathsf{S}_i^l \geq \xi\big\}$ and $\beta_i^l := \frac{\alpha_i^l}{\sum_{i\in I^l}\alpha_i^l}$, $l = 1,\dots,p$.
19. **Output**: Individual fingerprint $\{\mathsf{IF}_l^s\}_{s\in\mathcal{S},l=1,\dots,p}$, spectral fingerprint $\{\mathsf{SF}_l\}_{l=1,\dots,p}$.

[*this space intentionally left blank*]

Notes:

- l denotes the number of regions, s indexes subjects, K indicates the maximum allowed number of clusters.
- The number of sensors L_s varies across subjects, as indicated by the subscript s.
- When defining spectral fingerprint, only clusters that are representative for the majority of participants are accepted (Step 18 in Algorithm 7), but the rest of the clusters are not removed from the MATLAB workspace and saved if needed. Parameter ξ indicates that at least that many subjects had to contribute to each cluster. I^l is a set of cluster indices for which the number of contributing subjects $\mathsf{S}_i^l \geq \xi$.
- Spectral fingerprint can be also viewed as a *signature*, i.e., a set of feature clusters defined by Rubner in [229]. For l-th region of interest:

$$SF_l = \{(m_1, t_1), (m_2, t_2), \ldots, (m_k, t_k)\} \tag{A.21}$$

  where $l = 1, \ldots, p = 100$ is a region index, $m_1, \ldots, m_k \in \mathbb{R}^F$ are spectral modes, $t_1, \ldots, t_k \in \mathbb{R}_+$ are mode durations.
- D_i^l represents group-level centroid duration computed in Step 17 in Algorithm 7, according to the method presented in [9]. Note that

$$0 < (t_1 + \ldots + t_k) \leq k \cdot 100\%$$

  Arbitrarily chosen first mode, m_1, results from clustering $j = 1, ..., J$ individual fingerprint modes with corresponding durations $\tau_1, \ldots, \tau_J \in [0; 100\%]$ belonging to N unique subjects. Then spectral mode duration is computed as follows:

$$t_1 = \frac{\sum_{j=1}^{J} \tau_j}{N}$$

  complying with [9]. Considering that $J \leq N$, it follows that $t_1 \leq 100\%$. Same reasoning applies to $t_2, \ldots, t_k$, and thus:

$$0 < (t_1 + \ldots + t_k) \leq k \cdot 100\%$$

  ■.

Pseudocodes for fingerprint analysis

Algorithm 8 *Original group-level identification (with bagging) — Part 1*

1. **Input**: Set of subjects $\mathcal{S}$, individual fingerprints $\{\mathsf{IF}_u^s\}_{s\in\mathcal{S},l=1,\dots,p}$, the number of iterations Z, the number of bags $B \in \mathbb{N}$, the number of elements in each bag $P \in \mathcal{N}$.
2. For $r = 1$ to Z:
   (a) Divide $\mathcal{S} = \mathcal{S}_r^{\mathcal{L}} \cup \mathcal{S}_r^{\mathcal{T}}$, such that $\mathcal{S}_r^{\mathcal{L}} \cap \mathcal{S}_r^{\mathcal{T}} = \varnothing$ and $|\mathcal{S}_r^{\mathcal{L}}| = |\mathcal{S}_r^{\mathcal{T}}| =: P$.
   (b) (**Learning**) For $b = 1$ to B:
      i. Let $\mathcal{B}_{b,r}^{\mathcal{L}} := [\![s_1^{b,r}, \dots, s_P^{b,r}]\!]$ be a *multiset* (so-called *bag*), where s_i are drawn from $\mathcal{S}_r^{\mathcal{L}}$ (with repeats), $i = 1, \dots, P$.
      ii. For $l = 1$ to p:
         A. Define the pooled clusters set

$$\mathsf{PCL}_l^{b,r} := \bigcup_{s\in\mathcal{B}_{b,r}^{\mathcal{L}}} \{\mu_h^{s,l}\}_{h=1}^{k_{IF}}, \qquad l = 1, \dots, p, \tag{A.22}$$

         where $(\mu_1^{s,l}, \dots, \mu_{k,IF}^{s,l}) \in \mathsf{IF}_u^s$.
         B. Choose $k := k_{\text{mode}}$, such that $k_{\text{mode}} = \text{argmax}_g \Lambda_G(g)$, where $\Lambda_G(g)$ denotes the multiplicity of element g in multiset $G = [\![|I^l|, l = 1, \dots, p]\!]$. I^l is definied in Algorithm 7 Step 18.
         C. Perform k-means clustering (Algorithm 4) on $\mathsf{PCL}_l^{b,r}$ and obtain centroids $\mu_1^{b,l,r}, \dots, \mu_k^{b,l,r} \in \mathbb{R}^F$.
         D. For data $\mathsf{PCL}_l^{b,r}$, initial centroids $\mu_1^{b,l,r}, \dots, \mu_k^{b,l,r} \in \mathbb{R}^F$, $l = 1, \dots, p$, perform Algorithm 6 and obtain

$$\mathsf{GMM}_l^b := \big((\mu_1^{b,l,r}, \dots, \mu_k^{b,l,r}), (\Sigma_1^{b,l,r}, \dots, \Sigma_k^{b,l,r}), (\alpha_1^{b,l,r}, \dots, \alpha_k^{b,l,r})\big),$$

      iii. We average GMM models by

$$\mu_j^{l,r} := \frac{1}{B}\sum_{b=1}^{B} \mu_j^{b,l,r}, \quad \Sigma_j^{l,r} := \frac{1}{B}\sum_{b=1}^{B} \Sigma_j^{b,l,r}, \quad \alpha_j^{l,r} := \frac{1}{B}\sum_{b=1}^{B} \alpha_j^{b,l,r}, \qquad j = 1, \dots, k.$$

      iv. Put $\text{PDF}_l^r(x) := \sum_{j=1}^{k} \alpha_j^{l,r} \phi\big(x \mid \mu_j^{l,r}, \Sigma_j^{l,r}\big)$, where ϕ is defined in Eq. A.11.

Notes:

- l indexes brain regions, p is the number of regions.

Algorithm 8 *Original group-level identification (with bagging) — Part 2*

1. (cont.)

   (c) (**Testing**) For $b = 1$ to B:

      i. Let $\mathcal{B}_{b,r}^{\mathcal{T}} := [\![s_1^{b,r}, \ldots, s_P^{b,r}]\!]$ is a multiset, where s_i are drawn from $\mathcal{S}_r^{\mathcal{T}}$ (with repeats), $i = 1, \ldots, P$.

      ii. For $l = 1$ to p:

         A. We define

$$\mathsf{PCT}_l^{b,r} := \bigcup_{s \in \mathcal{B}_{b,r}^{\mathcal{T}}} \{\mu_h^{s,l}\}_{h=1}^{k_{IF}}, \tag{A.23}$$

         B. For $v = 1$ to p:

            (1) Define negative log-likelihoods

$$\mathsf{NLL}_{l,v}^{b,r} := - \sum_{\mu \in \mathsf{PCT}_l^{b,r}} \log \big(\mathrm{PDF}_v^r(\mu) \big). \tag{A.24}$$

            where PDF_v^r is defined in Step iv.

   (d) Average log-likelihoods are given by

$$\mathsf{NLL}_{l,v}^{r} := \frac{1}{B} \sum_{b=1}^{B} \mathsf{NLL}_{l,v}^{b,r} \,. \tag{A.25}$$

2. **Output**: Sequence of matrices $\big[\mathsf{NLL}_{l,v}^r \big]_{l,v=1,\ldots,p} \in \mathbb{R}^{p \times p}$, where $r = 1, \ldots, Z$.

Notes:

- Bootstrap aggregating (*bagging*) is a technique that aims to improve training model estimates [190]. In comparison with cross-validation, bagging reduces the estimates' variance at the cost of increased bias [199]. Bagging was proposed in [9] as a beneficial strategy for alleviating the problem of the low number of subjects ($N = 22$) which could lead to high variance of resulting estimates.
- In Step 2.(b)ii.B., k is chosen as the mode of the optimal number of clusters, i.e., the mode of all values $\overline{k}_l$, $l = 1, \ldots, p$ obtained in Step 11 of Algorithm 7.
- General idea behind the algorithm shown in [9] is to normalize a power spectrum over the whole brain, cluster with respect to regions and subjects, and find spectral fingerprints that best describe the group-level activity for different brain regions. Bagging and ranking are other parts of the original pipeline. They are designed to provide a measure (so-called *mean rank*) of how accurately and unambiguously individual fingerprints can be recognized as prototypes of a particular brain region's activity. Mean rank calculation is presented in the next algorithm.

Algorithm 9 *Original group-level ranking*

1. **Input**: Sequence of matrices $\left[\mathsf{NLL}_{l,v}^{r}\right]_{l,v=1,\ldots,p}$, where $r = 1, \ldots, Z$, $Z \in \mathbb{N}$.
2. For $r = 1$ to Z:
   (a) Let $\mathsf{NLL}_{l}^{r} := \left[\mathsf{NLL}_{l,v}^{r}\right]_{v=1,\ldots,p}$ be a row of matrix of average negative log-likelihoods.
   (b) Let $\left[\mathsf{TRNLL}_{l,v}^{r}\right]_{v=1,\ldots,p} := \mathsf{tiedrank}(\mathsf{NLL}_{l}^{r}) \in \mathbb{R}^{p}$ be a row of the rank matrix.
3. Define auxiliary matrix $U \in \mathbb{R}^{p \times Z}$ according to one of the following cases:
   (a) If homologue brain areas in the other hemisphere are NOT considered as correct assignments, then $U_{i,j} := \mathsf{TRNLL}_{i,i}^{j}$, the diagonal of the rank matrix.
   (b) (Specific for AAL atlas with 116 areas) If homologue brain areas in the other hemisphere are considered as correct assignments, then we have

$$U_{i,j} := \begin{cases} \min\left\{\mathsf{TRNLL}_{i,i}^{j}, \mathsf{TRNLL}_{i,i+1}^{j}\right\}, & \text{for } i \equiv 1 \mod 2, \\ \min\left\{\mathsf{TRNLL}_{i,i}^{j}, \mathsf{TRNLL}_{i,i-1}^{j}\right\}, & \text{for } i \equiv 0 \mod 2, \end{cases} \tag{A.26}$$

   for $i = 1, \ldots, 108$, and $U_{i,j} := \mathsf{TRNLL}_{i,i}^{j}$, for $i = 109, \ldots, 116$.
4. Define ranks:
   (a) (MEAN-RANK) Reshape $U \in \mathbb{R}^{p \times Z}$ to $\overline{U} \in \mathbb{R}^{1 \times (p \cdot Z)}$, trim 10% of the lowest and 10% of the highest values. Compute the mean to obtain $\mu_{\text{MR}} \in \mathbb{R}$.
   (b) (MEAN-RANK-STD) Compute the standard deviation along rows of U. Afterwards, we compute mean of that to obtain $\mu_{\text{MRSTD}} \in \mathbb{R}$.
   (c) (MEAN-ROI-RANK) Trim 10% of lowest and 10% of highest values separately for each row of U. Compute the mean along rows to obtain $\mu_{\text{MRR}} \in \mathbb{R}^{p \times 1}$.
5. **Output**: MEAN-RANK $\mu_{\text{MR}} \in \mathbb{R}$, MEAN-RANK-STD $\mu_{\text{MRRSTD}} \in \mathbb{R}$, MEAN-ROI-RANK $\mu_{\text{MRR}} \in \mathbb{R}^{p \times 1}$.

Notes:

In Step 2b, MATLAB function `tiedrank` is used. In Step 3b: in the AAL atlas, brain areas and their homologues appear alternately, e.g., 1. ***Left*** *precentral gyrus*, 2. ***Right*** *precentral gyrus*, 3. ***Left*** *superior frontal gyrus*, 4. ***Right*** *superior frontal gyrus*, etc., up to ROI nr 108. Starting from ROI nr 109, brain areas have no homologues. MEAN-RANK represents the **whole dataset**, telling how hard it is to identify brain areas on average. A mean rank of 1 indicates that the correct region model was on average the most likely to fit the test area; a rank of 2 indicates the correct region model was the second most likely, and so on. The expected mean rank for identification purely by chance equals $p/2 + 0.5$. MEAN-RANK-STD quantifies the variance (stability) of the identification along all iterations $r = 1, \ldots, Z$. MEAN-ROI-RANK quantifies how accurately **given ROI** is identified from its spectral fingerprint.

Algorithm 10 *Modified group-level identification (with cross-validation)*

1. **Input**: Individual fingerprint $\{\mathsf{IF}_l^s\}_{s\in\mathcal{S},l=1,\dots,p}$. V.
2. Assign each subject $s \in \mathcal{S}$ randomly(*) to one of the sets $\{\mathcal{B}_1,\dots,\mathcal{B}_V\}$ called *folds*, i.e., $\mathcal{B}_i \cap \mathcal{B}_j = \varnothing$, if $i \neq j$, and $\bigcup_{i=1}^V \mathcal{B}_i = \mathcal{S}$.
3. For each fold $v = 1,\dots,V$:
   (a) Define the *set of pooled training clusters* by the following
   $$\mathsf{PC}_l^s := \{\mu_h^{s,l}\}_{h=1}^k, \qquad l = 1,\dots,p, \quad s \in \mathcal{S}\setminus\mathcal{B}_v,$$
   $$\mathsf{PC}_l^{-v} := \bigcup_{s\in\mathcal{S}\setminus\mathcal{B}_v} \mathsf{PC}_l^s, \qquad l = 1,\dots,p.$$
   with k being the total number of clusters pooled into PC_l^s set.
   (b) Repeat Steps 12. to 18. of Algorithm 7 with PC_l replaced by PC_l^{-v}. The number of clusters for Step 12. in Algorithm 7 can be chosen by one of the criteria below:
      i. $\overline{k}_l^v := const$,
      ii. $\overline{k}_l^v$ is *optimal*(**),
      iii. $\overline{k}_l^v := \mathrm{mode}_{s\in\mathcal{B}_v} \overline{k}_{s,l}$.
   (c) We obtain the spectral fingerprint $\{\mathsf{SF}_l^{-v}\}_{l=1,\dots,p}$.
   (d) For $l = 1,\dots,p$, and for $m = 1,\dots,p$ we calculate the log-likelihood of the model by
   $$\mathsf{NLL}_{l,m}^{s,v} := -\sum_{\mu\in\mathsf{PC}_l^s} \log\big(\mathrm{PDF}_m(\mu)\big), \qquad s \in \mathcal{B}_v, \tag{A.27}$$
   where
   $$\mathrm{PDF}_m(x) := \sum_{j=1}^k \alpha_j^m \phi\big(x \mid \mu_j^m, \Sigma_j^m\big),$$
   and $(\mu_j^m, \Sigma_j^m, \alpha_j^m, D_j^m, S_j^m)_{j\in I^m} = \mathsf{SF}_m^{-v}$, with $I^m := \big\{i \in \{1,\dots,\overline{k}_l^v\} \mid \mathsf{S}_j^m \geq \xi\big\}$.
4. **Output**: Sequence of matrices $\big[\,\mathsf{NLL}_{l,m}^{s,v}\,\big]_{l,m=1,\dots,p} \in \mathbb{R}^{p\times p}$, where $s \in \mathcal{B}_v$, $v = 1,\dots,V$.

Notes:

- s indexes subjects, l indexes brain regions, p is the number of regions.
- V is the number of folds, typically $V = 10$.
- In Step 3a, PC_l^{-v} is the set of pooled clusters excluding subjects from vth fold, which in turn will be used during validation (Step 3d; note $s \in \mathcal{B}_v$).
- (*) subjects should be distributed uniformly across all folds, such that folds are equally sized if possible.
- (**) i.e., equals the same value as was calculated in step 11b in Algorithm 7 (no recalculation needed).

Algorithm 11 *Modified group-level ranking*

1. **Input**: Sequence of matrices $\left[\,\mathsf{NLL}_{l,m}^{s,v}\,\right]_{l,m=1,\ldots,p}$, where $v = 1, \ldots, V$.
2. For $v = 1$ to V:
   (a) Let $\mathsf{NLL}_{l}^{s,v} := \left[\,\mathsf{NLL}_{l,m}^{s,v}\,\right]_{m=1,\ldots,p}$ be a row of the matrix of negative log-likelihoods.
   (b) Let $\left[\,\mathsf{TRNLL}_{l,m}^{s,v}\,\right]_{m=1,\ldots,p} := \mathsf{tiedrank}(\mathsf{NLL}_{l}^{s,v}) \in \mathbb{R}^p$ be a row of the rank matrix.
3. Define auxiliary matrices $R^v, A^v \in \mathbb{R}^{p\times|\mathcal{B}_v|}$ by $A_{i,s}^v = \delta_{1,R_{i,s}^v}$, where $R_{i,s}^v := \mathsf{TRNLL}_{i,i}^{s,v}$, the diagonal of the rank matrix, and $\delta_{a,b}$ is the Kronecker delta.
4. Define metrics:
   (a) (MEAN-ACCURACY) Calculate the mean of $A^v \in \mathbb{R}^{p\times|\mathcal{B}_v|}$ along rows to obtain $\mu_{\text{MA}}^v \in \mathbb{R}^p$. Define *mean accuracy* as
   $$\mu_{\text{MA}} := \frac{1}{V}\sum_{v=1}^{V}\mu_{\text{MA}}^v \in \mathbb{R}^p. \tag{A.28}$$
   (b) (MEAN-ACCURACY-STD) Define *standard deviation of accuracy* as
   $$\sigma_{\text{MA}} := \sqrt{\frac{1}{V-1}\sum_{v=1}^{V}(\mu_{\text{MA}}^v - \mu_{\text{MA}})^2} \in \mathbb{R}^p. \tag{A.29}$$
5. Define ranks:
   (a) (MEAN-RANKS) Calculate the mean of $R^v \in \mathbb{R}^{p\times|\mathcal{B}_v|}$ along rows to obtain $\mu_{\text{MR}}^v \in \mathbb{R}^p$. Define *mean ranks* as
   $$\mu_{\text{MR}} := \frac{1}{V}\sum_{v=1}^{V}\mu_{\text{MR}}^v \in \mathbb{R}^p. \tag{A.30}$$
   (b) (MEAN-RANKS-STD) Define *standard deviation of mean ranks* as
   $$\sigma_{\text{MR}} := \sqrt{\frac{1}{V-1}\sum_{v=1}^{V}(\mu_{\text{MR}}^v - \mu_{\text{MR}})^2} \in \mathbb{R}^p. \tag{A.31}$$
6. **Output**: MEAN-ACCURACY $\mu_{\text{MA}} \in \mathbb{R}$, MEAN-ACCURACY-STD $\sigma_{\text{MA}} \in \mathbb{R}$, and MEAN-RANKS $\mu_{\text{MR}} \in \mathbb{R}^p$, MEAN-RANKS-STD $\sigma_{\text{MR}} \in \mathbb{R}^p$.

Notes:

In Step 2b, MATLAB function `tiedrank` is used. μ_{MA}^v and μ_{MR}^v quantify the accuracy of identification on average across subjects for single fold v. MEAN-ACCURACY, and MEAN-RANKS quantify the accuracy of identification on average both across subjects and folds, for each ROI separately. MEAN-ACCURACY-STD and MEAN-RANKS-STD are standard deviations for the mentioned averages. Mean ranks have the same meaning as in Algorithm 9 and take values from interval $[1, p]$. Accuracy values range is $[0, 1]$. Ranking homologues similar to Step 3 in Algorithm 9 was not implemented in the current ToFFi Toolbox revision.

Algorithm 12 *Identification (single-subject level)*

1. **Input**: Power spectra X_l^s obtained from Algorithm 3:

$$\mathsf{X}_l^s := \{x_l^s(n) \mid n \in \mathcal{N}_s'\} \subset \mathbb{R}^F, \qquad l = 1, \ldots, p$$
$$\mathsf{X}^s := \bigcup_{l=1}^{p} \mathsf{X}_l^s,$$

   where s is a single subject chosen from the set of all subjects $\mathcal{S}$,
2. Assign each power spectrum $x \in \mathsf{X}^s$ randomly(*) to one of the sets $\{\mathcal{B}_1, \ldots, \mathcal{B}_V\}$ called *folds*, i.e., $\mathcal{B}_i \cap \mathcal{B}_j = \varnothing$, if $i \neq j$, and $\bigcup_{i=1}^{V} \mathcal{B}_i = \mathsf{X}^s$.
3. For each fold $v = 1, \ldots, V$:
   (a) Repeat steps from 2 to 9 of Algorithm 7 with X_l^s replaced by $\mathsf{X}_m^{-v} := \bigcup_{v' \neq v}(\mathcal{B}_{v'} \cap \mathsf{X}_m^s) = \mathsf{X}_m^s \setminus \mathcal{B}_v$.
   (b) We obtain spectral fingerprint $\{\mathsf{IF}_l^{-v}\}_{l=1,\ldots,p}$.
   (c) For $l = 1, \ldots, p$, and for $m = 1, \ldots, p$ we calculate the log-likelihood of the model by

$$\mathsf{INLL}_{l,m}^{s,v} := -\sum_{x \in \mathcal{B}_v \cap \mathsf{X}_l^s} \log\big(\mathrm{PDF}_m(x)\big),$$

   where

$$\mathrm{PDF}_m(x) := \sum_{j=1}^{k} \alpha_j^{s,m} \phi\big(x \mid \mu_j^{s,m}, \Sigma_j^{s,m}\big),$$

   and $\mathsf{IF}_l^{-v} := \big(\mu_h^{s,l}, \Sigma_h^{s,l}, \alpha_h^{s,l}, T_h^{s,l}\big)_{h=1}^{k}$, where $l = 1, \ldots, p$, k is the number of clusters.
4. **Output**: Sequence of matrices $\big[\,\mathsf{INLL}_{l,m}^{s,v}\,\big]_{l,m=1,\ldots,p} \in \mathbb{R}^{p \times p}$, where $v = 1, \ldots, V$.

Notes:

- s indexes subjects, l indexes brain regions, p is the number of regions.
- V is the number of folds, typically $V = 10$.
- (*) power spectra from different regions of interest should be distributed uniformly across all folds, and class proportions should be preserved. Consequently follows that folds are nearly equally sized.

[*this space intentionally left blank*]

Algorithm 13 *Individual-level ranking*

1. **Input**: Sequence of matrices $\left[\,\mathsf{INLL}_{l,m}^{s,v}\,\right]_{l,m=1,\ldots,p}$, where $v = 1, \ldots, V$, $V \in \mathbb{N}$.
2. For $v = 1$ to V:
   (a) Let $\mathsf{INLL}_{l}^{s,v} := \left[\,\mathsf{INLL}_{l,m}^{s,v}\,\right]_{m=1,\ldots,p}$ be a row of the matrix of negative log-likelihoods.
   (b) Let $\left[\,\mathsf{TRINLL}_{l,m}^{s,v}\,\right]_{m=1,\ldots,p} := \mathsf{tiedrank}(\mathsf{INLL}_{l}^{s,v}) \in \mathbb{R}^p$ be a row of the rank matrix.
3. Define auxiliary matrices $R^s, A^s \in \mathbb{R}^{p\times V}$ by $A^s_{i,v} = \delta_{1,R^s_{i,v}}$, where $R^s_{i,v} := \mathsf{TRINLL}_{i,i}^{s,v}$, the diagonal of the rank matrix, where $\delta_{a,b}$ is the Kronecker delta.
4. Define metrics:
   (a) (SUBJECT MEAN-ACCURACY) Calculate the mean of $A^s \in \mathbb{R}^{p\times V}$ along rows to obtain $\mu^{s,v}_{\mathrm{MA}} \in \mathbb{R}^p$. Define *subject mean accuracy* as

$$\mu^s_{\mathrm{MA}} := \frac{1}{V}\sum_{v=1}^{V} \mu^{s,v}_{\mathrm{MA}} \in \mathbb{R}^p. \tag{A.32}$$

   (b) (SUBJECT MEAN-ACCURACY-STD) Define *subject standard deviation of accuracy* as

$$\sigma^s_{\mathrm{MA}} := \sqrt{\frac{1}{V-1}\sum_{v=1}^{V}(\mu^{s,v}_{\mathrm{MA}} - \mu^s_{\mathrm{MA}})^2} \in \mathbb{R}^p. \tag{A.33}$$

5. Define ranks:
   (a) (SUBJECT MEAN-RANKS) Calculate the mean of $R^s \in \mathbb{R}^{p\times V}$ along rows to obtain $\mu^{s,v}_{\mathrm{MR}} \in \mathbb{R}^p$. Define *subject mean ranks* as

$$\mu^s_{\mathrm{MR}} := \frac{1}{V}\sum_{v=1}^{V} \mu^{s,v}_{\mathrm{MR}} \in \mathbb{R}^p. \tag{A.34}$$

   (b) (SUBJECT MEAN-RANKS-STD) Define *subject standard deviation of mean ranks* as

$$\sigma^s_{\mathrm{MR}} := \sqrt{\frac{1}{V-1}\sum_{v=1}^{V}(\mu^{s,v}_{\mathrm{MR}} - \mu^s_{\mathrm{MR}})^2} \in \mathbb{R}^p. \tag{A.35}$$

6. **Output**: SUBJECT MEAN-ACCURACY $\mu^s_{\mathrm{MA}} \in \mathbb{R}$, SUBJECT MEAN-ACCURACY-STD $\sigma^s_{\mathrm{MA}} \in \mathbb{R}$, and MEAN-RANKS $\mu^s_{\mathrm{MR}} \in \mathbb{R}^p$, SUBJECT MEAN-RANKS-STD $\sigma^s_{\mathrm{MR}} \in \mathbb{R}^p$.

Notes:

- μ^v_{MA} and μ^v_{MR} quantify the accuracy of identification on average across subjects for single fold v. Ranking homologues similar to Step 3 in Algorithm 9 was not implemented in the current ToFFi Toolbox revision.
- Mean ranks and accuracy have the same meaning as in Algorithm 11, but are defined on the single-subject level instead of the group-level.
- Mean ranks take values from interval $[1, p]$. Accuracy values range is $[0, 1]$.

Algorithm 14 *Network analysis*

1. **Input**: Set of subjects $\mathcal{S}$, individual fingerprints $\{\mathsf{IF}_l^s\}_{s\in\mathcal{S},l=1,\ldots,p}$, spectral fingerprints $\{\mathsf{SF}_l\}_{l=1,\ldots,p}$.

2. Define

$$\begin{aligned}\mathsf{PC}_l^s &:= \{\mu_h^{s,l}\}_{h=1}^k, \qquad l=1,\ldots,p, \quad s\in\mathcal{S},\\ \mathsf{PC}_l &:= \bigcup_{s\in\mathcal{S}} \mathsf{PC}_u^s, \qquad l=1,\ldots,p.\end{aligned}$$

   with k being the total number of clusters pooled into PC_l^s set, and

$$\mathsf{NL}_{l,v} := -\sum_{\mu\in\mathsf{PC}_l} \log\big(\mathrm{PDF}^v(\mu)\big), \quad l,v=1,\ldots,p, \tag{A.36}$$

   where $\mathrm{PDF}^v(x) := \sum_{j=1}^k \alpha_j^v \phi\big(x \mid \mu_j^v, \Sigma_j^v\big)$ is the probability density function of the Gaussian mixture given by spectral fingerprint $\mathsf{SF}_v = \big(\mu_i^v, \Sigma_i^v, \alpha_i^v, D_i^v, \mathsf{S}_i^v\big)_{i\in I^v}$, $l=1,\ldots,p$, defined in Equation (A.20).

3. Let NL_l be lth row of matrix $[\mathsf{NL}_{l,v}]_{l,v=1,\ldots,p}$. Compute the cosine distance of rows as follows

$$D_{i,j} := d\big(\mathsf{NL}_i, \mathsf{NL}_j\big) = 1 - \frac{\mathsf{NL}_i \cdot \mathsf{NL}_j}{\|\mathsf{NL}_i\|\|\mathsf{NL}_j\|}, \quad i,j=1,\ldots,p. \tag{A.37}$$

4. Obtain clusters $\{\mathcal{C}_h\}_{h\in H}$ by performing the hierarchical clustering using distance matrix $[D_{i,j}]_{i,j=1,\ldots,p}$ with UPGMA (unweighted pair group method with arithmetic mean) as an algorithm for computing the distance between clusters, where H is a set of indices of clusters for given truncation height ι.

5. **Output**: Clusters $\{\mathcal{C}_h\}_{h\in H}$.

Notes:

- s indexes subjects, l indexes brain regions, p is the number of regions.
- In this hierarchical clustering procedure, a binary tree is formed first by progressively joining data into bigger and bigger clusters, then truncation at height ι means that the clustering with $|\{\mathcal{C}_h\}_{h\in H}| = \iota$ is accepted as the final solution [219].

[*this space intentionally left blank*]

Table A.2. List of typical algorithm parameters for the analysis of the Human Connectome Project MEG dataset (Chapter 5).

Parameter description	Symbol	Default value
Common parameters		
Number of frequencies of interest	F	42
Number of activity sources	M	5798
Number of time points in single time segment	T	509
Number of subjects	$\|S\|$	89
Number of brain areas	p	100
Single source voxel volume	ρ	8 mm^3
Frequencies of interest	LS	(below the table*)
Data preprocessing — Alg. 3		
z-score threshold for artifact rejection	Γ	2.5
Whole Spectral Fingerprinting — p. 188		
Regularization amount for GMM clustering algorithm	λ	0.012
Maximum value of k in optimal selection algorithm	K	15
Number of k-means replications	R	10
Criterion for setting k for Individual Fingerprints (Step 3)	—	optimal
Number of iterations in optimal selection algorithm for Individual Fingerprints	E_{IF}	250
Criterion for setting k for Spectral Fingerprints (Step 11)	—	optimal
Number of iterations in optimal selection algorithm for Spectral Fingerprints	E_{SF}	600
Minimum number of subjects required in SF for given cluster to be accepted	ξ	1
Identification (group-level) — p. 194		
Number of folds	V	10
Criterion for setting k (Step 3b)	—	mode
Maximum value of k in optimal selection algorithm	K	15
Number of k-means replications	R	10
Regularization amount for GMM clustering algorithm	λ	0.012
Minimum number of subjects required in SF for given cluster to be accepted	ξ	64
Identification (individual level) — p. 196		
Number of folds	V	10
Criterion for setting k (Step 3a)	—	optimal
Number of iterations in optimal selection algorithm for Spectral Fingerprints	E_{IF}	250
Maximum value of k in optimal selection algorithm	K	15
Number of k-means replications	R	10
Regularization amount for GMM clustering algorithm	λ	0.012
Network Analysis — p. 198		
Tree truncation height	ι	20

```
* - LS = [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0,
          8.0, 8.5, 9.5, 10.5, 11.5, 12.5, 14.0, 15.5, 17.0, 18.5, 20.5,
          23.0, 25.0, 27.5, 30.5, 33.5, 37.0, 41.0, 45.0, 50.0, 55.0, 60.5,
          67.0, 73.5, 81.0, 89.5, 98.5, 109.0, 120.0] Hz
```

B Spectral Fingerprinting — beta version script

The following code was published by dr A. Keitel and prof. J. Gross alongside their journal article [9].

Listing 10: Beta version script of Spectral Fingerprinting by Keitel & Gross, 2016.

```
%% SPECTRAL FINGERPRINTS
% Version 1.0 beta

% Copyright (c) Anne Keitel & Joachim Gross, 2016
% Centre for Cognitive Neuroimaging (CCNi)
% Institute of Neuroscience and Psychology
% University of Glasgow
% ------------------------------------------------------------------------
% This script guides you through the process of clustering your data to get
% 'spectral profiles' for atlas-defined brain areas. You can find the script
% and data to work with here: http://meg.psy.gla.ac.uk/
% Note that the clustering process can yield slighmichotly different results
% each time you create 1st- and 2nd-level GM models.
% ------------------------------------------------------------------------
% Please note that this is a beta version and needs to be tested.
% The script runs on MATLAB 2013 and up. Things might be different with
% earlier versions.
% ------------------------------------------------------------------------
% If you use any of the following code, please cite:
% Keitel A  & Gross J (2016). Individual human brain areas
% can be identified from their characteristic spectral activation
% fingerprints. PLoS Biology, 14(6).
% ------------------------------------------------------------------------
% This program is free software; you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation; either version 3 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with this program; if not, write to the Free Software
% Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA

% License can be found online: http://meg.psy.gla.ac.uk/

% ------------------------------------------------------------------------
% Last update: 03/07/2016
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DIRECTORIES, FIELDTRIP INITIALISATION, TEMPLATES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

HomeFolder = '';                                        % folder with matlab
    scripts and data
cd(HomeFolder)
```

```
48
49 FieldtripPath = '';                                          % Fieldtrip
50 addpath(FieldtripPath);
51 ft_defaults                                                  % initialise Fieldtrip
52
53 % templates for atlas (AAL atlas) and 10-mm source model
54 AtlasDir = [FieldtripPath 'template\atlas\aal\ROI_MNI_V4.nii'];
55 SourceMod = [HomeFolder 'templategrid10mm.mat'];
56 % Note that this is a template from an earlier fieldtrip version, these
57 % templates change. Depending on your template/spatial resolution, the
58 % interpolation with the atlas will be different.
59
60 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61 % INTERPOLATION
62 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
63 % load atlas and sourcemodel and interpolate both
64
65 atlas = ft_read_atlas(AtlasDir);
66 atlas = ft_convert_units(atlas, 'cm');
67
68 load(SourceMod);    % [sourcemodel]
69
70 cfg = [];
71 cfg.interpmethod    = 'nearest';
72 cfg.parameter       = 'tissue';
73 sourceatlas = ft_sourceinterpolate(cfg, atlas, sourcemodel);
74
75 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
76 % SOURCE PROJECTION AND 1ST-LEVEL CLUSTERING
77 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
78 % information / parameters:
79
80 goodsub = 1:22;                          % vector with to-be-analysed subjects
81 nsub = length(goodsub);
82
83 nroi = numel(atlas.tissuelabel);         % number of ROIs in atlas
84 goodroi = [1:94 96:116];                 % all interpolated ROIs (look at sourceatlas.
       tissue)
85
86 nclust = 10;                             % number of individual clusters for 1-st-level
        models
87 % 10 works for us (with 400 - 500 individual trials).
88 % Depending on your data you might need to change it.
89
90 logfreqs = logspace(0,log10(120),50);    % logarithmic frequency resolution
91 nfreq = 42;                              % number of analysed frequencies after
       fieldtrip takes out the redundant ones
92 % (check numel(freq.freq) after FFT)
93
94 regu = .012;                             % regularisation for GM models (possible to go
        higher, e.g. 0.05)
95
96 opts = statset('Display','final');       % 'off' for no information display
97
98 indivname = 'indivclust';                % designated name for individual cluster files
```

```
99
100 %% start loop: source projection and first-level clustering
101
102 for isub = goodsub,
103 fprintf('\nAnalysing sub %d .. \n', isub);
104
105 % load clean, segmented data
106 load(sprintf('%sdata_clean_%d',HomeFolder,isub));     % [data]
107
108 % FFT
109 cfg=[];
110 cfg.method      = 'mtmfft';
111 cfg.output      = 'fourier';
112 cfg.taper       = 'dpss';
113 cfg.tapsmofrq   = 2;
114 cfg.pad         = 2;
115 cfg.keeptrials  = 'yes';
116 cfg.foi         = logfreqs;
117 freq = ft_freqanalysis(cfg,data);
118
119 ff = freq.freq;                                  % save frequency resolution later
120
121 % load previously computed LCMV filter
122 load(sprintf('%sLCMV_%d',HomeFolder,isub));     % [filt]
123 ninsvox = size(filt,1);                          % number of voxels inside brain
124
125 % facilitate data handling by numerically increasing values
126 % (can be different for each MEG system; look at data)
127 freq.fourierspctrm = freq.fourierspctrm*1e14;
128
129 sourcef = zeros(length(freq.cumsumcnt),ninsvox,nfreq);    % create source matrix (
    trials x voxels x freq)
130
131 % SOURCE PROJECTION; apply weights to fourier spectra and compute power
132 m = 1;
133 for k = 1:length(freq.cumtapcnt),
134 for k2 = 1:freq.cumtapcnt(1),
135 sourcef(k,:,:) = squeeze(sourcef(k,:,:))+abs(filt*squeeze(freq.fourierspctrm(m,:,:)))
    .^2;
136 m = m+1;
137 end
138 end
139
140 % NORMALISE ACTIVITY (we use ratio normalisation here)
141 avgact = squeeze(mean(mean(sourcef,1),2));      % average activity of whole brain
142 tmp = repmat(avgact',[ninsvox 1]);
143 for k = 1:length(freq.cumsumcnt),
144 sourcef(k,:,:) = (squeeze(sourcef(k,:,:)))./tmp;
145 end
146
147 roicluster = zeros(nclust,nfreq,nroi);          % create matrix for individual cluster
148 id = cell(nroi,1);                               % create matrix for cluster indices
149 pt = zeros(nroi,nclust);                         % create matrix for cluster durations
150
151 % CLUSTERING
```

```
% average each atlas ROI and cluster power spectra (kmeans and GM models)
for iroi=goodroi

% find all voxels belonging to a ROI
roivoxel = find(sourceatlas.tissue==iroi);

% match the voxel indices between the sourcemodel (which includes inside
% and outside voxels) with our voxel indices (which only has inside voxels)
[~,insideroivoxel] = intersect(sourcemodel.inside,roivoxel);

roimean = squeeze(mean(sourcef(:,insideroivoxel,:),2));
%------------------------------
olier = zscore(mean(roimean,2));            % clean data in frequency domain
fi = find(olier>2.5);                       % by removing outlier
roimean(fi,:) = [];                         % (optional)
%------------------------------

roimean = roimean-1;   % make values below/above zero (instead of one)

% k-means clustering
[idd,rr] = kmeans(roimean,nclust,'Options',opts,'Distance','cosine','Replicates',10);

% Gaussian Mixture clustering; uses k-means cluster as starting point
gm = gmdistribution.fit(roimean,nclust,'Start',idd,'Options',opts,'Regularize',regu);
roicluster(:,:,iroi) = gm.mu;           % 1-st level cluster
id{iroi} = cluster(gm,roimean);         % index cluster

%-----------------------------------------------
for iclust = 1:nclust                                   % get the amount of trials
    belonging
pt(iroi,iclust) = length(find(id{iroi}==iclust));   % to each cluster (expressed in %)
pt(iroi,iclust) = pt(iroi,iclust)/length(id{iroi})*100;
end
%-----------------------------------------------

end

save(sprintf('%s%s_nclust%d_sub%d',HomeFolder,indivname,nclust,isub), 'roicluster','ff
    ','avgact','pt','id');
end

% SAVES:
% 'roicluster'  - GM cluster (10 per participant)
% 'ff'          - used frequencies (nfreq should be length(ff) )
% 'pp'          - average activity of all voxels
% 'pt'          - percentage of trials belonging to each cluster
% 'id'          - index of trials belonging to a cluster

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OPTIMAL NUMBER OF CLUSTERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% load first-level clusters and store in one matrix
allsubs = zeros(nclust,nfreq,nroi,nsub);
for isub = goodsub
```

```
filename = sprintf('%s%s_nclust%d_sub%d',HomeFolder,indivname,nclust,isub);
load(filename,'roicluster','ff');
allsubs(:,:,:,isub) = roicluster;
end
% optional: remove all empty entries, i.e. if you have missing subjects
% e.g., allsubs(:,:,:,badsub)=[];

% rearrange data
allrois = cell(nroi,1);
for iroi = 1:nroi,
allrois{iroi}(:,:,:) = squeeze(allsubs(:,:,iroi,:));
end
clear allsubs

% build sub index to keep track of subs
subindex = [];
for isub = 1:nsub,
subindex = [subindex ones(1,nclust)*isub];
end
subindex=subindex';

% parameters etc.
clustnum = 1:15;                % list of number of clusters to evaluate
niter = 1000;                   % number of iterations for optimal cluster analysis

optname = sprintf('%soptclust',HomeFolder);

%% start loop: optimal number of clusters
% Depending on the number of iterations and rois, this can take a very long
% time (e.g., 1000 iterations for 115 ROIs takes ~12 hours, depending on your machine)
% Results from optimal cluster analysis are provided (load 'nclustopt' in next step)

allnclust = NaN(nroi,niter);    % create matrix for optimal cluster numbers
allsilhval = NaN(nroi,niter);   % create matrix for Silhoutte values

for iiter=1:niter,
for iroi = goodroi,

% rearrange data
roidata = allrois{iroi};
roidata = permute(roidata,[1 3 2]);
roidata = reshape(roidata,nclust*nsub,nfreq);
tmp = roidata;

% evaluate cluster solutions
E = evalclusters(tmp,'kmeans','silhouette','klist',clustnum,'Distance','cosine');
allnclust(iroi,iiter) = E.OptimalK;                     % store optimal number for
    each iteration
allsilhval(iroi,iiter) = E.CriterionValues(E.OptimalK); % store Silhouette value for
    each iteration
end
end

save(optname,'allnclust','allsilhval');
% Save optimal no of clusters for each iteration plus silhouette values.
```

```
% Cosine distance criterium is important (others don't capture shape of
% amplitudes so well).

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GROUP CLUSTERING FOR FINAL GM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Restart MATLAB for this step, otherwise it might report an error when
% trying to perform the gmdistribution function in the 2nd-level clustering loop

% build 1st-level cluster matrix etc. (run lines 81 - 92, plus lines 203 - 231)

% load (your own) optimal cluster data
load(optname);                              % [allnclust allsilhval]

% get optimal cluster numbers from Silhoutte values
nclustopt = NaN(nroi,1);                    % create matrix for optimal cluster
    numbers
[~,iclustmaxid] = max(allsilhval,[],2);
for iroi = goodroi,
nclustopt(iroi) = allnclust(iroi,iclustmaxid(iroi));
end;

% % (if you work with our data, load vector with optimal cluster numbers here)
% load('nclustopt');                          % [nclustopt]

% parameters etc.
nsubcl = cell(nroi,1);                      % create matrix for participant
    contributions
allidx = zeros(length(subindex),nroi);      % create matrix for cluster indices

opts = statset('MaxIter',1000);             % maximum number of iterations

finalname = sprintf('%sGMmodel_allsubs',HomeFolder);

%% start loop: second-level clustering

for iroi = goodroi,

% rearrange data (all subject data in 1st dimension, frequencies in 2nd dimension)
roidata = allrois{iroi};
roidata = permute(roidata(:,:,:),[1 3 2]);
roidata = reshape(roidata,nclust*nsub,nfreq);

optn=nclustopt(iroi);

% k-means with optimal number of clusters
[idd,rr] = kmeans(roidata,optn,'Options',opts,'Distance','cosine','Replicates',10);

% Gaussian Mixture clustering; uses k-means cluster as starting point
gm{iroi} = gmdistribution.fit(roidata,optn,'Start',idd,'Options',opts,'Regularize',
    regu);
idx = cluster(gm{iroi},roidata);
allidx(:,iroi) = idx;               % indices of cluster in 1st-level models

```

```
%-----------------------------------
for k = 1:optn,                       % number of subjects contributing to each cluster
fi = find(idx==k);                 % (important if you only want to show cluster of
    majority of participants)
nsubcl{iroi}(k) = length(unique(subindex(fi)));
end
%-----------------------------------
end

save(finalname,'gm','allidx','nsubcl','ff');
% SAVES:
% 'gm'          - GM cluster results (gm.mu are the final clusters)
% 'allIDX'      - index of trials belonging to a cluster
% 'nsubcl'      - number of subjects contributing to each cluster
% 'ff'          - used frequencies

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TO PLOT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% We've used a modified version of the mseb.m function
% (multiple shaded error bars) to plot modes and SEM
% http://www.mathworks.com/matlabcentral/fileexchange/47950-mseb-x-y-errbar-lineprops-
    transparent-

% A simple plot could look like this:
roi = 1;                                    % ROI you want to plot

figure, h = plot(ff,gm{roi}.mu,'LineWidth',2);
title(sprintf('Raw spectral modes in %s',atlas.tissuelabel{roi}),'Interpreter','none',
    'FontSize',12,'FontWeight','bold');
xlabel('Frequencies (Hz)','FontSize',12);
ylabel('Normalised power','FontSize',12);
```

C ToFFi Toolbox — repository

ToFFi Toolbox consists of over 250 scripts, and for this reason, the whole code was not attached to this dissertation, however, the reader can freely access the code on Github with the links provided below.

Find ToFFi Toolbox on GitHub - `https://github.com/micholeodon/ToFFi_Toolbox`

ToFFi Documentation - `https://github.com/micholeodon/ToFFi_Toolbox/blob/master/ToFFi_Toolbox-20211013/docs/ToFFi_Manual.pdf`

Installation guide

NOTE: Recommended versions of the software

Recommended versions of the software:

- MATLAB - R2021a or newer,
- Fieldtrip - revision 20210816 or newer.

The following toolboxes need to be installed alongside MATLAB:

- Signal Processing Toolbox,
- Statistics and Machine Learning Toolbox,
- Parallel Computing Toolbox.

Getting the software

1. Install MATLAB,
2. Download the Fieldtrip Toolbox:
    - use `https://www.fieldtriptoolbox.org/download/` or
    - download it directly using any FTP client to enter `ftp://ftp.fieldtriptoolbox.org/pub/fieldtrip/`)
3. Setup Fieldtrip:
    - `https://www.fieldtriptoolbox.org/faq/should_i_add_fieldtrip_with_all_subdirectories_to_my_matlab_path/`
4. Download the ToFFi repository from Github:
    - `https://github.com/micholeodon/ToFFi_Toolbox`
5. Extract downloaded repository if it is compressed within a `.zip` file,
6. The ToFFi toolbox is ready to use.

WARNING! MATLAB Search paths

1. **Do not** manually add any MATLAB search paths related to the ToFFi Toolbox content using `addpath` command or GUI, as it may confuse the toolbox.

2. Before running the toolbox routines, remove any MATLAB search paths to old Fieldtrip versions you do not plan to use. Otherwise, Fieldtrip will behave confusingly and could produce unexpected errors.

Getting the input data

To get the data set that was used in the toolbox demonstration (Section 4.9 of the dissertation), one should read sections 5.2—5.4 in ToFFi Documentation.

When using other data sets, please remember that input data should comply with the specification found in chapter 5 of the ToFFi Documentation.

References

[1] G. Buzsaki, *Rhythms of the Brain*, 1st ed. Oxford ; New York: Oxford University Press, Jun. 2011.

[2] H.-J. Park and K. Friston, "Structural and Functional Brain Networks: From Connections to Cognition," *Science*, vol. 342, no. 6158, p. 1238411, Nov. 2013. doi: 10.1126/science.1238411

[3] S. H. Koslow and M. F. Huerta, Eds., *Neuroinformatics: An Overview of the Human Brain Project*. New York: Psychology Press, Oct. 2019.

[4] B. Kryzhanovsky, W. Dunin-Barkowski, V. Redko, and Y. Tiumentsev, Eds., *Advances in Neural Computation, Machine Learning, and Cognitive Research IV: Selected Papers from the XXII International Conference on Neuroinformatics, ... in Computational Intelligence Book 925)*, 1st ed. Springer, Oct. 2020.

[5] P. Érdi, B. S. Bhattacharya, and A. L. Cochran (eds.), *Computational Neurology and Psychiatry*, 1st ed., ser. Springer Series in Bio-/Neuroinformatics 6. Springer International Publishing, 2017.

[6] M. DiLuca and J. Olesen, "The Cost of Brain Diseases: A Burden or a Challenge?" *Neuron*, vol. 82, no. 6, pp. 1205–1208, Jun. 2014. doi: 10.1016/j.neuron.2014.05.044

[7] N. Yahata, K. Kasai, and M. Kawato, "Computational neuroscience approach to biomarkers and treatments for mental disorders," *Psychiatry and Clinical Neurosciences*, vol. 71, no. 4, pp. 215–237, 2017. doi: 10.1111/pcn.12502

[8] T. H. Donner and M. Siegel, "A framework for local cortical oscillation patterns," *Trends in Cognitive Sciences*, vol. 15, no. 5, pp. 191–199, May 2011. doi: 10.1016/j.tics.2011.03.007

[9] A. Keitel and J. Gross, "Individual human brain areas can be identified from their characteristic spectral activation fingerprints," *PLoS Biol*, vol. 14, no. 6, p. e1002498, 2016. doi: 10.1371/journal.pbio.1002498

[10] M. K. Komorowski, K. Rykaczewski, T. Piotrowski, K. Jurewicz, J. Wojciechowski, A. Keitel, J. Dreszer, and W. Duch, "ToFFi – Toolbox for Frequency-based Fingerprinting of Brain Signals," *arXiv:2110.09919 [cs, q-bio]*, Oct. 2021.

[11] C. Lubinus, J. Orpella, A. Keitel, H. Gudi-Mindermann, A. K. Engel, B. Roeder, and J. M. Rimmele, "Data-Driven Classification of Spectral Profiles Reveals

Brain Region-Specific Plasticity in Blindness," *Cerebral Cortex*, vol. 31, no. 5, pp. 2505–2522, May 2021. doi: 10.1093/cercor/bhaa370

[12] M. K. Komorowski, M. Aghabeig, J. Nikadon, T. Piotrowski, J. Dreszer, B. Bałaj, M. Lewandowska, J. Wojciechowski, N. Pawlaczyk, M. Szmytke, A. Cichocki, and W. Duch, "Multi-level Explanations in Neuroscience II: EEG Spectral Fingerprints and Tensor Decompositions for Understanding Brain Activity - Initial Results," *Acta Physica Polonica B*, vol. 49, p. 2011, 2018. doi: 10.5506/APhysPolB.49.2011

[13] D. S. Margulies, S. S. Ghosh, A. Goulas, M. Falkiewicz, J. M. Huntenburg, G. Langs, G. Bezgin, S. B. Eickhoff, F. X. Castellanos, M. Petrides, E. Jefferies, and J. Smallwood, "Situating the default-mode network along a principal gradient of macroscale cortical organization," *Proceedings of the National Academy of Sciences*, vol. 113, no. 44, pp. 12 574–12 579, Nov. 2016. doi: 10.1073/pnas.1608282113

[14] G. Buzsáki and B. O. Watson, "Brain rhythms and neural syntax: Implications for efficient coding of cognitive content and neuropsychiatric disease," *Dialogues in Clinical Neuroscience*, vol. 14, no. 4, pp. 345–367, Dec. 2012.

[15] D. Papo, "Time scales in cognitive neuroscience," *Frontiers in Physiology*, vol. 4, p. 86, Apr. 2013. doi: 10.3389/fphys.2013.00086

[16] C. Lanczos, *Discourse on Fourier Series*. Philadelphia: SIAM-Society for Industrial and Applied Mathematics, Sep. 2016.

[17] J. Szabatin, *Podstawy Teorii Sygnałów*, 4th ed. Warsaw: Wydawnictwa Komunikacji i Lacznosci WKL, Jan. 2003.

[18] A. P. Burgess, "On the interpretation of synchronization in EEG hyperscanning studies: A cautionary note," *Frontiers in Human Neuroscience*, vol. 7, 2013. doi: 10.3389/fnhum.2013.00881

[19] M. Nedergaard and A. Verkhratsky, "Artifact versus reality—How astrocytes contribute to synaptic events," *Glia*, vol. 60, no. 7, pp. 1013–1023, 2012. doi: 10.1002/glia.22288

[20] R. Martín, R. Bajo-Grañeras, R. Moratalla, G. Perea, and A. Araque, "Circuit-specific signaling in astrocyte-neuron networks in basal ganglia pathways," *Science*, Aug. 2015. doi: 10.1126/science.aaa7945

[21] P. Morquette, D. Verdier, A. Kadala, J. Féthière, A. G. Philippe, R. Robitaille, and A. Kolta, "An astrocyte-dependent mechanism for neuronal rhythmogenesis," *Nature Neuroscience*, vol. 18, no. 6, pp. 844–854, Jun. 2015. doi: 10.1038/nn.4013

[22] J. A. Filosa, A. D. Bonev, S. V. Straub, A. L. Meredith, M. K. Wilkerson, R. W. Aldrich, and M. T. Nelson, "Local potassium signaling couples neuronal activity to vasodilation in the brain," *Nature Neuroscience*, vol. 9, no. 11, pp. 1397–1403, Nov. 2006. doi: 10.1038/nn1779

[23] T. Takano, G.-F. Tian, W. Peng, N. Lou, W. Libionka, X. Han, and M. Nedergaard, "Astrocyte-mediated control of cerebral blood flow," *Nature Neuroscience*, vol. 9, no. 2, pp. 260–267, Feb. 2006. doi: 10.1038/nn1623

[24] G. C. Brown and J. J. Neher, "Microglial phagocytosis of live neurons," *Nature Reviews Neuroscience*, vol. 15, no. 4, pp. 209–216, Apr. 2014. doi: 10.1038/nrn3710

[25] Y. Zhan, R. C. Paolicelli, F. Sforazzini, L. Weinhard, G. Bolasco, F. Pagani, A. L. Vyssotski, A. Bifone, A. Gozzi, D. Ragozzino, and C. T. Gross, "Deficient neuron-microglia signaling results in impaired functional brain connectivity and social behavior," *Nature Neuroscience*, vol. 17, no. 3, pp. 400–406, Mar. 2014. doi: 10.1038/nn.3641

[26] Y. Lee, B. M. Morrison, Y. Li, S. Lengacher, M. H. Farah, P. N. Hoffman, Y. Liu, A. Tsingalia, L. Jin, P.-W. Zhang, L. Pellerin, P. J. Magistretti, and J. D. Rothstein, "Oligodendroglia metabolically support axons and contribute to neurodegeneration," *Nature*, vol. 487, no. 7408, pp. 443–448, Jul. 2012. doi: 10.1038/nature11314

[27] L. Pinto and M. Götz, "Radial glial cell heterogeneity—The source of diverse progeny in the CNS," *Progress in Neurobiology*, vol. 83, no. 1, pp. 2–23, Sep. 2007. doi: 10.1016/j.pneurobio.2007.02.010

[28] R. W. Williams and K. Herrup, "The Control of Neuron Number," *Annual Review of Neuroscience*, vol. 11, no. 1, pp. 423–453, 1988. doi: 10.1146/annurev.ne.11.030188.002231

[29] J. W. Kalat, *Biological Psychology*, 13th ed. Boston, MA: Cengage Learning, Jan. 2018.

[30] A. Hess and J. Z. Young, "The nodes of Ranvier," *Proceedings of the Royal Society of London. Series B - Biological Sciences*, vol. 140, no. 900, pp. 301–320, Nov. 1952. doi: 10.1098/rspb.1952.0063

[31] R. Fitzhugh, "Computation of Impulse Initiation and Saltatory Conduction in a Myelinated Nerve Fiber," *Biophysical Journal*, vol. 2, no. 1, pp. 11–21, Jan. 1962. doi: 10.1016/S0006-3495(62)86837-4

[32] M. N. Rasband and E. Peles, "Mechanisms of node of Ranvier assembly," *Nature Reviews Neuroscience*, vol. 22, no. 1, pp. 7–20, Jan. 2021. doi: 10.1038/s41583-020-00406-8

[33] R. Hari and A. Puce, *MEG-EEG Primer*. Oxford University Press, Mar. 2017.

[34] R. S. Smith and Z. J. Koles, "Myelinated nerve fibers: Computed effect of myelin thickness on conduction velocity," *American Journal of Physiology-Legacy Content*, Nov. 1970. doi: 10.1152/ajplegacy.1970.219.5.1256

[35] Y. Tang, J. R. Nyengaard, D. M. De Groot, and H. J. G. Gundersen, "Total regional and global number of synapses in the human brain neocortex," *Synapse*, vol. 41, no. 3, pp. 258–273, 2001. doi: 10.1002/syn.1083

[36] C. Sherrington, *The Integrative Action of the Nervous System*. CUP Archive, 1952.

[37] H. B. Enderton, *A Mathematical Introduction to Logic*, 2nd ed. San Diego, Calif: Harcourt/Academic Press, 2001.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning Series, F. Bach, Ed. Cambridge, MA, USA: MIT Press, Nov. 2016.

[39] J. LeDoux, *Synaptic Self: How Our Brains Become Who We Are*, 0th ed. New York, NY: Penguin Books, Jan. 2003.

[40] M. Steriade, D. A. McCormick, and T. J. Sejnowski, "Thalamocortical Oscillations in the Sleeping and Aroused Brain," *Science*, Oct. 1993. doi: 10.1126/science.8235588

[41] F. Ferrarelli, S. Sarasso, Y. Guller, B. A. Riedner, M. J. Peterson, M. Bellesi, M. Massimini, B. R. Postle, and G. Tononi, "Reduced natural oscillatory frequency of frontal thalamocortical circuits in schizophrenia," *Archives of General Psychiatry*, vol. 69, no. 8, pp. 766–774, Aug. 2012. doi: 10.1001/archgenpsychiatry.2012.147

[42] F. A. Middleton and P. L. Strick, "Basal ganglia and cerebellar loops: Motor and cognitive circuits," *Brain Research Reviews*, vol. 31, no. 2, pp. 236–250, Mar. 2000. doi: 10.1016/S0165-0173(99)00040-5

[43] B. U. Westner, S. S. Dalal, A. Gramfort, V. Litvak, J. C. Mosher, R. Oostenveld, and J.-M. Schoffelen, "A unified view on beamformers for M/EEG source reconstruction," *NeuroImage*, vol. 246, p. 118789, Feb. 2022. doi: 10.1016/j.neuroimage.2021.118789

[44] Y. Komura, R. Tamura, T. Uwano, H. Nishijo, K. Kaga, and T. Ono, "Retrospective and prospective coding for predicted reward in the sensory thalamus," *Nature*, vol. 412, no. 6846, pp. 546–549, Aug. 2001. doi: 10.1038/35087595

[45] V. Braitenberg, "Thoughts on the cerebral cortex," *Journal of Theoretical Biology*, vol. 46, no. 2, pp. 421–447, Aug. 1974. doi: 10.1016/0022-5193(74)90007-1

[46] O. Sporns, *Networks of the Brain*, 1st ed. Cambridge, Mass: The MIT Press, Oct. 2010.

[47] K. E. Garcia, C. D. Kroenke, and P. V. Bayly, "Mechanics of cortical folding: Stress, growth and stability," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 373, no. 1759, p. 20170321, Nov. 2018. doi: 10.1098/rstb.2017.0321

[48] R. Nieuwenhuys, "The neocortex. An overview of its evolutionary development, structural organization and synaptology," *Anatomy and Embryology*, vol. 190, no. 4, pp. 307–337, Oct. 1994. doi: 10.1007/BF00187291

[49] K. Zilles, "Brodmann: A pioneer of human brain mapping—his impact on concepts of cortical organization," *Brain*, vol. 141, no. 11, pp. 3262–3278, Nov. 2018. doi: 10.1093/brain/awy273

[50] M. Strotzer, "One Century of Brain Mapping Using Brodmann Areas*," *Clinical Neuroradiology*, vol. 19, no. 3, pp. 179–186, Aug. 2009. doi: 10.1007/s00062-009-9002-3

[51] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of Neural Science*. New York: McGraw-Hill Education Ltd., Dec. 2012.

[52] S. S. P. DSc, Ed., *Gray's Anatomy: The Anatomical Basis of Clinical Practice*, 41st ed. New York: Elsevier, Oct. 2015.

[53] A. Chatterjee and M. H. Southwood, "Cortical blindness and visual imagery," *Neurology*, vol. 45, no. 12, pp. 2189–2195, Dec. 1995. doi: 10.1212/WNL.45.12.2189

[54] G. Goldenberg, W. Mu¨llbacher, and A. Nowak, "Imagery without perception—A case study of anosognosia for cortical blindness," *Neuropsychologia*, vol. 33, no. 11, pp. 1373–1382, Nov. 1995. doi: 10.1016/0028-3932(95)00070-J

[55] H. Klüver and P. C. Bucy, "Preliminary analysis of functions of the temporal lobes in monkeys. 1939," *The Journal of Neuropsychiatry and Clinical Neurosciences*, vol. 9, no. 4, pp. 606–620, 1997. doi: 10.1176/jnp.9.4.606

[56] D. J. Lanska, "The Klüver-Bucy Syndrome," *Neurologic-Psychiatric Syndromes in Focus - Part I*, vol. 41, pp. 77–89, 2018. doi: 10.1159/000475721

[57] E. M. Hubbard, M. Piazza, P. Pinel, and S. Dehaene, "Interactions between number and space in parietal cortex," *Nature Reviews Neuroscience*, vol. 6, no. 6, pp. 435–448, Jun. 2005. doi: 10.1038/nrn1684

[58] A. de la Vega, L. J. Chang, M. T. Banich, T. D. Wager, and T. Yarkoni, "Large-Scale Meta-Analysis of Human Medial Frontal Cortex Reveals Tripartite Functional Organization," *Journal of Neuroscience*, vol. 36, no. 24, pp. 6553–6562, Jun. 2016. doi: 10.1523/JNEUROSCI.4402-15.2016

[59] P. Bach-y-Rita, "Brain plasticity as a basis for recovery of function in humans," *Neuropsychologia*, vol. 28, no. 6, pp. 547–554, Jan. 1990. doi: 10.1016/0028-3932(90)90033-K

[60] M. A. Castro-Alamancos, L. M. Garcia-Segura, and J. Borrell, "Transfer of Function to a Specific Area of the Cortex After Induced Recovery from Brain Damage," *European Journal of Neuroscience*, vol. 4, no. 9, pp. 853–863, 1992. doi: 10.1111/j.1460-9568.1992.tb00195.x

[61] J. Konorski, *Conditioned Reflexes and Neuron Organization*, ser. Conditioned Reflexes and Neuron Organization. New York, NY, US: Cambridge University Press, 1948.

[62] R. J. Ilmoniemi and J. Sarvas, *Brain Signals: Physics and Mathematics of MEG and EEG*, 1st ed. Cambridge, MA: The MIT Press, May 2019.

[63] D. A. Fleisch, *A Student's Guide to Maxwell's Equations*. Cambridge, UK ; New York: Cambridge University Press, 2008.

[64] G. Nolte, "The magnetic lead field theorem in the quasi-static approximation and its use for magnetoencephalography forward calculation in realistic volume conductors," *Physics in Medicine and Biology*, vol. 48, no. 22, pp. 3637–3652, Nov. 2003. doi: 10.1088/0031-9155/48/22/002

[65] Q. Liu, M. Ganzetti, N. Wenderoth, and D. Mantini, "Detecting Large-Scale Brain Networks Using EEG: Impact of Electrode Density, Head Modeling and Source Localization," *Frontiers in Neuroinformatics*, vol. 12, 2018. doi: 10.3389/fninf.2018.00004

[66] M. Stenroos, A. Hunold, and J. Haueisen, "Comparison of three-shell and simplified volume conductor models in magnetoencephalography," *NeuroImage*, vol. 94, pp. 337–348, Jul. 2014. doi: 10.1016/j.neuroimage.2014.01.006

[67] E. D. Adrian and B. H. C. Matthews, "The Berger rhythm: Potential changes from the occipital lobes in man," *Brain: A Journal of Neurology*, vol. 57, pp. 355–385, 1934. doi: 10.1093/brain/57.4.355

[68] M. Steriade, P. Gloor, R. R. Llinás, F. H. Lopes da Silva, and M. M. Mesulam, "Basic mechanisms of cerebral rhythmic activities," *Electroencephalography and Clinical Neurophysiology*, vol. 76, no. 6, pp. 481–508, Dec. 1990. doi: 10.1016/0013-4694(90)90001-Z

[69] M. M. Steriade and R. W. McCarley, *Brainstem Control of Wakefulness and Sleep*. Springer Science & Business Media, Mar. 2013.

[70] G. Buzsáki, N. Logothetis, and W. Singer, "Scaling Brain Size, Keeping Timing: Evolutionary Preservation of Brain Rhythms," *Neuron*, vol. 80, no. 3, pp. 751–764, Oct. 2013. doi: 10.1016/j.neuron.2013.10.002

[71] G. M. Di Liberto, G. Marion, and S. A. Shamma, "Accurate Decoding of Imagined and Heard Melodies," *Frontiers in Neuroscience*, vol. 15, 2021.

[72] S. Haegens, H. Cousijn, G. Wallis, P. J. Harrison, and A. C. Nobre, "Inter- and intra-individual variability in alpha peak frequency," *NeuroImage*, vol. 92, pp. 46–55, May 2014. doi: 10.1016/j.neuroimage.2014.01.049

[73] M. Rosanova, A. Casali, V. Bellina, F. Resta, M. Mariotti, and M. Massimini, "Natural Frequencies of Human Corticothalamic Circuits," *Journal of Neuroscience*, vol. 29, no. 24, pp. 7679–7685, Jun. 2009. doi: 10.1523/JNEUROSCI.0445-09.2009

[74] E. Niedermeyer and F. H. L. da Silva, *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 2005.

[75] G. Pfurtscheller and F. L. Da Silva, "Event-related EEG/MEG synchronization and desynchronization: Basic principles," *Clinical neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999.

[76] G. G. Knyazev, J. Y. Slobodskoj-Plusnin, A. V. Bocharov, and L. V. Pylkova, "The default mode network and EEG alpha oscillations: An independent component analysis," *Brain Research*, vol. 1402, pp. 67–79, Jul. 2011. doi: 10.1016/j.brainres.2011.05.052

[77] S. Palva and J. M. Palva, "New vistas for α-frequency band oscillations," *Trends in Neurosciences*, vol. 30, no. 4, pp. 150–158, Apr. 2007. doi: 10.1016/j.tins.2007.02.001

[78] W. Klimesch, "Alpha-band oscillations, attention, and controlled access to stored information," *Trends in Cognitive Sciences*, vol. 16, no. 12, pp. 606–617, Dec. 2012. doi: 10.1016/j.tics.2012.10.007

[79] A. Capilla, L. Arana, M. García-Huéscar, M. Melcón, J. Gross, and P. Campo, "The natural frequencies of the resting human brain: An MEG-based atlas," p. 2021.11.17.468973, Nov. 2021.

[80] W. N. Kuhlman, "Functional topography of the human mu rhythm," *Electroencephalography and Clinical Neurophysiology*, vol. 44, no. 1, pp. 83–93, Jan. 1978. doi: 10.1016/0013-4694(78)90107-4

[81] R. Salmelin and R. Hari, "Spatiotemporal characteristics of sensorimotor neuromagnetic rhythms related to thumb movement," *Neuroscience*, vol. 60, no. 2, pp. 537–550, May 1994. doi: 10.1016/0306-4522(94)90263-1

[82] H. Jasper and W. Penfield, "Electrocorticograms in man: Effect of voluntary movement upon the electrical activity of the precentral gyrus," *Archiv für Psychiatrie und Nervenkrankheiten*, vol. 183, no. 1, pp. 163–174, Jan. 1949. doi: 10.1007/BF01062488

[83] J. Tiihonen, M. Kajola, and R. Hari, "Magnetic mu rhythm in man," *Neuroscience*, vol. 32, no. 3, pp. 793–800, Jan. 1989. doi: 10.1016/0306-4522(89)90299-6

[84] R. Salmelin, M. Hámáaláinen, M. Kajola, and R. Hari, "Functional Segregation of Movement-Related Rhythmic Activity in the Human Brain," *NeuroImage*, vol. 2, no. 4, pp. 237–243, Dec. 1995. doi: 10.1006/nimg.1995.1031

[85] J. Tiihonen, R. Hari, M. Kajola, J. Karhu, S. Ahlfors, and S. Tissari, "Magnetoencephalographic 10-Hz rhythm from the human auditory cortex," *Neuroscience Letters*, vol. 129, no. 2, pp. 303–305, Aug. 1991. doi: 10.1016/0304-3940(91)90486-D

[86] L. Lehtelä, R. Salmelin, and R. Hari, "Evidence for reactive magnetic 10-Hz rhythm in the human auditory cortex," *Neuroscience Letters*, vol. 222, no. 2, pp. 111–114, Jan. 1997. doi: 10.1016/S0304-3940(97)13361-4

[87] J. Cannon, M. M. McCarthy, S. Lee, J. Lee, C. Börgers, M. A. Whittington, and N. Kopell, "Neurosystems: Brain rhythms and cognitive processing," *European Journal of Neuroscience*, vol. 39, no. 5, pp. 705–719, 2014. doi: 10.1111/ejn.12453

[88] M. E. Hasselmo and C. E. Stern, "Theta rhythm and the encoding and retrieval of space and time," *NeuroImage*, vol. 85, pp. 656–666, Jan. 2014. doi: 10.1016/j.neuroimage.2013.06.022

[89] J. Kamiński, A. Brzezicka, and A. Wróbel, "Short-term memory capacity (7 ± 2) predicted by theta to gamma cycle length ratio," *Neurobiology of Learning and Memory*, vol. 95, no. 1, pp. 19–23, Jan. 2011. doi: 10.1016/j.nlm.2010.10.001

[90] L.-T. Hsieh and C. Ranganath, "Frontal midline theta oscillations during working memory maintenance and episodic encoding and retrieval," *NeuroImage*, vol. 85, pp. 721–729, Jan. 2014. doi: 10.1016/j.neuroimage.2013.08.003

[91] M. S. Mellem, S. Wohltjen, S. J. Gotts, A. S. Ghuman, and A. Martin, "Intrinsic frequency biases and profiles across human cortex," *Journal of Neurophysiology*, vol. 118, no. 5, pp. 2853–2864, Nov. 2017. doi: 10.1152/jn.00061.2017

[92] K. Mahjoory, J.-M. Schoffelen, A. Keitel, and J. Gross, "The frequency gradient of human resting-state brain oscillations follows cortical hierarchies," *eLife*, vol. 9, p. e53715, Aug. 2020. doi: 10.7554/eLife.53715

[93] S. Karakaş, "A review of theta oscillation and its functional correlates," *International Journal of Psychophysiology*, vol. 157, pp. 82–99, Nov. 2020. doi: 10.1016/j.ijpsycho.2020.04.008

[94] J. E. Lisman and O. Jensen, "The Theta-Gamma Neural Code," *Neuron*, vol. 77, no. 6, pp. 1002–1016, Mar. 2013. doi: 10.1016/j.neuron.2013.03.007

[95] F. Roux and P. J. Uhlhaas, "Working memory and neural oscillations: Alpha–gamma versus theta–gamma codes for distinct WM information?" *Trends in Cognitive Sciences*, vol. 18, no. 1, pp. 16–25, Jan. 2014. doi: 10.1016/j.tics.2013.10.010

[96] P. J. Uhlhaas, G. Pipa, S. Neuenschwander, M. Wibral, and W. Singer, "A new look at gamma? High- (>60 Hz) γ-band activity in cortical networks: Function, mechanisms and impairment," *Progress in Biophysics and Molecular Biology*, vol. 105, no. 1, pp. 14–28, Mar. 2011. doi: 10.1016/j.pbiomolbio.2010.10.004

[97] R. Srinivasan, P. L. Nunez, D. M. Tucker, R. B. Silberstein, and P. J. Cadusch, "Spatial sampling and filtering of EEG with spline Laplacians to estimate cortical potentials," *Brain Topography*, vol. 8, no. 4, pp. 355–366, Jun. 1996. doi: 10.1007/BF01186911

[98] N. Pouratian and S. A. Sheth, Eds., *Stereotactic and Functional Neurosurgery: Principles and Applications*, 1st ed. Springer, Feb. 2020.

[99] P. Fries, D. Nikolić, and W. Singer, "The gamma cycle," *Trends in Neurosciences*, vol. 30, no. 7, pp. 309–316, Jul. 2007. doi: 10.1016/j.tins.2007.05.005

[100] C. S. Herrmann, I. Fründ, and D. Lenz, "Human gamma-band activity: A review on cognitive and behavioral correlates and network models," *Neuroscience & Biobehavioral Reviews*, vol. 34, no. 7, pp. 981–992, Jun. 2010. doi: 10.1016/j.neubiorev.2009.09.001

[101] M. A. Whittington, M. O. Cunningham, F. E. LeBeau, C. Racca, and R. D. Traub, "Multiple origins of the cortical gamma rhythm," *Developmental Neurobiology*, vol. 71, no. 1, pp. 92–106, 2011. doi: 10.1002/dneu.20814

[102] W. Sedley and M. Cunningham, "Do cortical gamma oscillations promote or suppress perception? An under-asked question with an over-assumed answer," *Frontiers in Human Neuroscience*, vol. 7, 2013.

[103] G. Buzsáki and X.-J. Wang, "Mechanisms of Gamma Oscillations," *Annual Review of Neuroscience*, vol. 35, no. 1, pp. 203–225, 2012. doi: 10.1146/annurev-neuro-062111-150444

[104] S. W. Hughes, M. L. Lőrincz, H. R. Parri, and V. Crunelli, "Chapter 10 - Infraslow (<0.1Hz) oscillations in thalamic relay nuclei: Basic mechanisms and significance to health and disease states," in *Progress in Brain Research*, ser. Slow Brain Oscillations of Sleep, Resting State and Vigilance, E. J. W. Van Someren, Y. D. Van Der Werf, P. R. Roelfsema, H. D. Mansvelder, and F. H. Lopes Da Silva, Eds. Elsevier, Jan. 2011, vol. 193, pp. 145–162.

[105] P. Achermann and A. A. Borbély, "Low-frequency (<1Hz) oscillations in the human sleep electroencephalogram," *Neuroscience*, vol. 81, no. 1, pp. 213–222, Aug. 1997. doi: 10.1016/S0306-4522(97)00186-3

[106] B. Morillon, L. H. Arnal, C. E. Schroeder, and A. Keitel, "Prominence of delta oscillatory rhythms in the motor cortex and their relevance for auditory and speech perception," *Neuroscience & Biobehavioral Reviews*, vol. 107, pp. 136–142, Dec. 2019. doi: 10.1016/j.neubiorev.2019.09.012

[107] R. T. Canolty and R. T. Knight, "The functional role of cross-frequency coupling," *Trends in Cognitive Sciences*, vol. 14, no. 11, pp. 506–515, Nov. 2010. doi: 10.1016/j.tics.2010.09.001

[108] J. Aru, J. Aru, V. Priesemann, M. Wibral, L. Lana, G. Pipa, W. Singer, and R. Vicente, "Untangling cross-frequency coupling in neuroscience," *Current Opinion in Neurobiology*, vol. 31, pp. 51–61, Apr. 2015. doi: 10.1016/j.conb.2014.08.002

[109] A. Hyafil, A.-L. Giraud, L. Fontolan, and B. Gutkin, "Neural Cross-Frequency Coupling: Connecting Architectures, Mechanisms, and Functions," *Trends in Neurosciences*, vol. 38, no. 11, pp. 725–740, Nov. 2015. doi: 10.1016/j.tins.2015.09.001

[110] S. J. Luck and E. S. Kappenman, Eds., *The Oxford Handbook of Event-Related Potential Components*, ser. Oxford Library of Psychology. Oxford: University Press, 2012.

[111] C. Neuper, M. Wörtz, and G. Pfurtscheller, "ERD/ERS patterns reflecting sensorimotor activation and deactivation," *Progress in Brain Research*, vol. 159, pp. 211–222, 2006. doi: 10.1016/S0079-6123(06)59014-4

[112] G. Buzsáki and A. Draguhn, "Neuronal Oscillations in Cortical Networks," *Science*, vol. 304, no. 5679, pp. 1926–1929, Jun. 2004. doi: 10.1126/science.1099745

[113] M. Gerster, G. Waterstraat, V. Litvak, K. Lehnertz, A. Schnitzler, E. Florin, G. Curio, and V. Nikulin, "Separating Neural Oscillations from Aperiodic 1/f

Activity: Challenges and Recommendations," *Neuroinformatics*, Apr. 2022. doi: 10.1007/s12021-022-09581-8

[114] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes," *Nature Reviews Neuroscience*, vol. 13, no. 6, pp. 407–420, Jun. 2012. doi: 10.1038/nrn3241

[115] C. C. Finlay, S. Maus, C. D. Beggan, T. N. Bondar, A. Chambodut, T. A. Chernova, A. Chulliat, V. P. Golovkov, B. Hamilton, M. Hamoudi, R. Holme, G. Hulot, W. Kuang, B. Langlais, V. Lesur, F. J. Lowes, H. Lühr, S. Macmillan, M. Mandea, S. McLean, C. Manoj, M. Menvielle, I. Michaelis, N. Olsen, J. Rauberg, M. Rother, T. J. Sabaka, A. Tangborn, L. Tøffner-Clausen, E. Thébault, A. W. P. Thomson, I. Wardinski, Z. Wei, and T. I. Zvereva, "International Geomagnetic Reference Field: The eleventh generation," *Geophysical Journal International*, vol. 183, no. 3, pp. 1216–1230, Dec. 2010. doi: 10.1111/j.1365-246X.2010.04804.x

[116] B. D. Josephson, "Possible new effects in superconductive tunnelling," *Physics Letters*, vol. 1, no. 7, pp. 251–253, Jul. 1962. doi: 10.1016/0031-9163(62)91369-0

[117] B. A. Glowacki, "Niobium aluminide as a source of high-current superconductors," *Intermetallics*, vol. 7, no. 2, pp. 117–140, Feb. 1999. doi: 10.1016/S0966-9795(98)00084-3

[118] J. R. Hopkins and D. K. Finnemore, "Surface superconductivity in niobium and niobium-tantalum alloys," *Physical Review B*, vol. 9, no. 1, pp. 108–114, Jan. 1974. doi: 10.1103/PhysRevB.9.108

[119] M. Van Camp, O. de Viron, A. Watlet, B. Meurers, O. Francis, and C. Caudron, "Geophysics From Terrestrial Time-Variable Gravity Measurements," *Reviews of Geophysics*, vol. 55, no. 4, pp. 938–992, 2017. doi: 10.1002/2017RG000566

[120] M. Van Camp, O. Francis, and T. Lecocq, "Recording Belgium's Gravitational History," *Eos*, vol. 98, Dec. 2017. doi: 10.1029/2017EO089743

[121] P. Hansen, M. Kringelbach, and R. Salmelin, Eds., *MEG: An Introduction to Methods*. New York: Oxford University Press, 2010.

[122] S. W. Statham and G. H. Joseph, "Magnetic alloy," US Patent US1 582 353A, Apr., 1926.

[123] R. Hari, S. Baillet, G. Barnes, R. Burgess, N. Forss, J. Gross, M. Hämäläinen, O. Jensen, R. Kakigi, F. Mauguière, N. Nakasato, A. Puce, G.-L. Romani, A. Schnitzler, and S. Taulu, "IFCN-endorsed practical guidelines for clinical magnetoencephalography (MEG)," *Clinical Neurophysiology*, vol. 129, no. 8, pp. 1720–1747, Aug. 2018. doi: 10.1016/j.clinph.2018.03.042

[124] J. Gross, S. Baillet, G. R. Barnes, R. N. Henson, A. Hillebrand, O. Jensen, K. Jerbi, V. Litvak, B. Maess, R. Oostenveld, L. Parkkonen, J. R. Taylor, V. van Wassenhove, M. Wibral, and J.-M. Schoffelen, "Good practice for conducting and reporting MEG research," *NeuroImage*, vol. 65, pp. 349–363, Jan. 2013. doi: 10.1016/j.neuroimage.2012.10.001

[125] E. Boto, S. S. Meyer, V. Shah, O. Alem, S. Knappe, P. Kruger, T. M. Fromhold, M. Lim, P. M. Glover, P. G. Morris, R. Bowtell, G. R. Barnes, and M. J. Brookes, "A new generation of magnetoencephalography: Room temperature measurements using optically-pumped magnetometers," *NeuroImage*, vol. 149, pp. 404–414, Apr. 2017. doi: 10.1016/j.neuroimage.2017.01.034

[126] N. Koshev, A. Butorina, E. Skidchenko, A. Kuzmichev, A. Ossadtchi, M. Ostras, M. Fedorov, and P. Vetoshko, "Evolution of MEG: A first MEG-feasible fluxgate magnetometer," *Human Brain Mapping*, vol. 42, no. 15, pp. 4844–4856, 2021. doi: 10.1002/hbm.25582

[127] R. A. Poldrack, J. A. Mumford, and T. E. Nichols, *Handbook of Functional MRI Data Analysis*. Cambridge New York Melbourne Madrid: Cambridge University Press, 2011.

[128] B. M. Dale, M. A. Brown, and R. C. Semelka, *MRI: Basic Principles and Applications*, 5th ed. Chichester, West Sussex ; Hoboken, NJ: Wiley-Blackwell, 2015.

[129] J. N. Acharya and V. J. Acharya, "Overview of EEG Montages and Principles of Localization," *Journal of Clinical Neurophysiology: Official Publication of the American Electroencephalographic Society*, vol. 36, no. 5, pp. 325–329, Sep. 2019. doi: 10.1097/WNP.0000000000000538

[130] J. T. Cacioppo, L. G. Tassinary, and G. Berntson, Eds., *Handbook of Psychophysiology*, 3rd ed. Cambridge: Cambridge University Press, 2007.

[131] S. Ogawa, T. M. Lee, A. R. Kay, and D. W. Tank, "Brain magnetic resonance imaging with contrast dependent on blood oxygenation." *Proceedings of the*

National Academy of Sciences, vol. 87, no. 24, pp. 9868–9872, Dec. 1990. doi: 10.1073/pnas.87.24.9868

[132] H. H. Mitchell, T. S. Hamilton, F. R. Steggerda, and H. W. Bean, "The Chemical Composition of the Adult Human Body and its Bearing on the Biochemistry of Growth," *Journal of Biological Chemistry*, vol. 158, no. 3, pp. 625–637, May 1945. doi: 10.1016/S0021-9258(19)51339-4

[133] L. Landini, V. Positano, and M. Santarelli, Eds., *Advanced Image Processing in Magnetic Resonance Imaging*. Boca Raton: CRC Press, Jan. 2017.

[134] G. B. Chavhan, P. S. Babyn, B. Thomas, M. M. Shroff, and E. M. Haacke, "Principles, Techniques, and Applications of T2*-based MR Imaging and Its Special Applications," *RadioGraphics*, vol. 29, no. 5, pp. 1433–1449, Sep. 2009. doi: 10.1148/rg.295095034

[135] E. Sareen, S. Zahar, D. V. D. Ville, A. Gupta, A. Griffa, and E. Amico, "Exploring MEG brain fingerprints: Evaluation, pitfalls, and interpretations," *NeuroImage*, vol. 240, p. 118331, Oct. 2021. doi: 10.1016/j.neuroimage.2021.118331

[136] E. S. Finn, X. Shen, D. Scheinost, M. D. Rosenberg, J. Huang, M. M. Chun, X. Papademetris, and R. T. Constable, "Functional connectome fingerprinting: Identifying individuals using patterns of brain connectivity," *Nature Neuroscience*, vol. 18, no. 11, pp. 1664–1671, Oct. 2015. doi: 10.1038/nn.4135

[137] O. Miranda-Dominguez, B. D. Mills, S. D. Carpenter, K. A. Grant, C. D. Kroenke, J. T. Nigg, and D. A. Fair, "Connectotyping: Model Based Fingerprinting of the Functional Connectome," *PLOS ONE*, vol. 9, no. 11, p. e111048, Nov. 2014. doi: 10.1371/journal.pone.0111048

[138] L. A. Farwell, "Brain fingerprinting: A comprehensive tutorial review of detection of concealed information with event-related brain potentials," *Cognitive Neurodynamics*, vol. 6, no. 2, pp. 115–154, Apr. 2012. doi: 10.1007/s11571-012-9192-2

[139] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, 1st ed. New York: Wiley-Interscience, May 2001.

[140] M. Pal, R. Roy, J. Basu, and M. S. Bepari, "Blind source separation: A review and analysis," in *2013 International Conference Oriental COCOSDA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, Nov. 2013. doi: 10.1109/ICSDA.2013.6709849 pp. 1–5.

[141] B. D. van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, pp. 4–24, Apr. 1988.

[142] B. Van Veen, W. Van Drongelen, M. Yuchtman, and A. Suzuki, "Localization of brain electrical activity via linearly constrained minimum variance spatial filtering," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 9, pp. 867–880, Sep. 1997. doi: 10.1109/10.623056

[143] J. Gross, J. Kujala, M. Hamalainen, L. Timmermann, A. Schnitzler, and R. Salmelin, "Dynamic imaging of coherent sources: Studying neural interactions in the human brain," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 2, pp. 694–699, Jan. 2001. doi: 10.1073/pnas.98.2.694

[144] T. Piotrowski, J. Nikadon, and A. Moiseev, "Localization of brain activity from EEG/MEG using MV-PURE framework," *Biomedical Signal Processing and Control*, vol. 64, p. 102243, Feb. 2021. doi: 10.1016/j.bspc.2020.102243

[145] T. Piotrowski, J. Nikadon, and D. Gutiérrez, "MV-PURE Spatial Filters With Application to EEG/MEG Source Reconstruction," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 553–567, Feb. 2019. doi: 10.1109/TSP.2018.2883851

[146] K. Rykaczewski, J. Nikadon, W. Duch, and T. Piotrowski, "supFunSim: Spatial Filtering Toolbox for EEG," *Neuroinformatics*, vol. 19, no. 1, pp. 107–125, Jan. 2021. doi: 10.1007/s12021-020-09464-w

[147] M. Siegel, T. H. Donner, R. Oostenveld, P. Fries, and A. K. Engel, "Neuronal Synchronization along the Dorsal Visual Pathway Reflects the Focus of Spatial Attention," *Neuron*, vol. 60, no. 4, pp. 709–719, Nov. 2008. doi: 10.1016/j.neuron.2008.09.010

[148] H. van Dijk, J.-M. Schoffelen, R. Oostenveld, and O. Jensen, "Prestimulus Oscillatory Activity in the Alpha Band Predicts Visual Discrimination Ability," *Journal of Neuroscience*, vol. 28, no. 8, pp. 1816–1823, Feb. 2008. doi: 10.1523/JNEUROSCI.1853-07.2008

[149] L. Elshoff, M. Muthuraman, A. R. Anwar, G. Deuschl, U. Stephani, J. Raethjen, and M. Siniatchkin, "Dynamic Imaging of Coherent Sources Reveals Different Network Connectivity Underlying the Generation and Perpetuation of Epileptic Seizures," *PLOS ONE*, vol. 8, no. 10, p. e78422, Oct. 2013. doi: 10.1371/journal.pone.0078422

[150] W. Singer, "Cortical dynamics revisited," *Trends in Cognitive Sciences*, vol. 17, no. 12, pp. 616–626, Dec. 2013. doi: 10.1016/j.tics.2013.09.006

[151] B. Biswal, F. Z. Yetkin, V. M. Haughton, and J. S. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo-planar mri," *Magnetic Resonance in Medicine*, vol. 34, no. 4, pp. 537–541, 1995. doi: 10.1002/mrm.1910340409

[152] J. S. Damoiseaux, S. a. R. B. Rombouts, F. Barkhof, P. Scheltens, C. J. Stam, S. M. Smith, and C. F. Beckmann, "Consistent resting-state networks across healthy subjects," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 37, pp. 13 848–13 853, Sep. 2006. doi: 10.1073/pnas.0601417103

[153] B. T. T. Yeo, F. M. Krienen, J. Sepulcre, M. R. Sabuncu, D. Lashkari, M. Hollinshead, J. L. Roffman, J. W. Smoller, L. Zöllei, J. R. Polimeni, B. Fischl, H. Liu, and R. L. Buckner, "The organization of the human cerebral cortex estimated by intrinsic functional connectivity," *Journal of Neurophysiology*, vol. 106, no. 3, pp. 1125–1165, Sep. 2011. doi: 10.1152/jn.00338.2011

[154] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar, and S. E. Petersen, "Functional network organization of the human brain," *Neuron*, vol. 72, no. 4, pp. 665–678, Nov. 2011. doi: 10.1016/j.neuron.2011.09.006

[155] R. D. Pascual-Marqui, D. Lehmann, M. Koukkou, K. Kochi, P. Anderer, B. Saletu, H. Tanaka, K. Hirata, E. R. John, L. Prichep, R. Biscay-Lirio, and T. Kinoshita, "Assessing interactions in the brain with exact low-resolution electromagnetic tomography," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 369, no. 1952, pp. 3768–3784, Oct. 2011. doi: 10.1098/rsta.2011.0081

[156] H. Yuan, L. Ding, M. Zhu, V. Zotev, R. Phillips, and J. Bodurka, "Reconstructing Large-Scale Brain Resting-State Networks from High-Resolution EEG: Spatial and Temporal Comparisons with fMRI," *Brain Connectivity*, vol. 6, no. 2, pp. 122–135, Sep. 2015. doi: 10.1089/brain.2014.0336

[157] Q. Liu, S. Farahibozorg, C. Porcaro, N. Wenderoth, and D. Mantini, "Detecting large-scale networks in the human brain using high-density electroencephalography," *Human Brain Mapping*, vol. 38, no. 9, pp. 4631–4643, 2017. doi: 10.1002/hbm.23688

[158] M. S. Hämäläinen and R. J. Ilmoniemi, "Interpreting magnetic fields of the brain: Minimum norm estimates," *Medical & Biological Engineering & Computing*, vol. 32, no. 1, pp. 35–42, Jan. 1994. doi: 10.1007/BF02512476

[159] M. J. Brookes, M. Woolrich, H. Luckhoo, D. Price, J. R. Hale, M. C. Stephenson, G. R. Barnes, S. M. Smith, and P. G. Morris, "Investigating the electrophysiological basis of resting state networks using magnetoencephalography," *Proceedings of the National Academy of Sciences*, vol. 108, no. 40, pp. 16 783–16 788, Oct. 2011. doi: 10.1073/pnas.1112685108

[160] F. de Pasquale, S. Della Penna, A. Z. Snyder, L. Marzetti, V. Pizzella, G. L. Romani, and M. Corbetta, "A Cortical Core for Dynamic Integration of Functional Networks in the Resting Human Brain," *Neuron*, vol. 74, no. 4, pp. 753–764, May 2012. doi: 10.1016/j.neuron.2012.03.031

[161] A. Fornito, A. Zalesky, and E. Bullmore, *Fundamentals of Brain Network Analysis*. Amsterdam ; Boston: Academic Press, Mar. 2016.

[162] C. W. Lynn and D. S. Bassett, "The physics of brain network structure, function and control," *Nature Reviews Physics*, vol. 1, no. 5, pp. 318–332, May 2019. doi: 10.1038/s42254-019-0040-8

[163] J. T. Baker, D. G. Dillon, L. M. Patrick, J. L. Roffman, R. O. Brady, D. A. Pizzagalli, D. Öngür, and A. J. Holmes, "Functional connectomics of affective and psychotic pathology," *Proceedings of the National Academy of Sciences*, vol. 116, no. 18, pp. 9050–9059, Apr. 2019. doi: 10.1073/pnas.1820780116

[164] M. F. Glasser, T. S. Coalson, E. C. Robinson, C. D. Hacker, J. Harwell, E. Yacoub, K. Ugurbil, J. Andersson, C. F. Beckmann, M. Jenkinson, S. M. Smith, and D. C. Van Essen, "A multi-modal parcellation of human cerebral cortex," *Nature*, vol. 536, no. 7615, pp. 171–178, Aug. 2016. doi: 10.1038/nature18933

[165] T. Ito, K. R. Kulkarni, D. H. Schultz, R. D. Mill, R. H. Chen, L. I. Solomyak, and M. W. Cole, "Cognitive task information is transferred between brain regions via resting-state network topology," *Nature Communications*, vol. 8, no. 1, p. 1027, Oct. 2017. doi: 10.1038/s41467-017-01000-w

[166] J. F. Hipp, D. J. Hawellek, M. Corbetta, M. Siegel, and A. K. Engel, "Large-scale cortical correlation structure of spontaneous oscillatory activity," *Nature Neuroscience*, vol. 15, no. 6, pp. 884–890, Jun. 2012. doi: 10.1038/nn.3101

[167] M. van Vliet, M. Liljeström, S. Aro, R. Salmelin, and J. Kujala, "Analysis of Functional Connectivity and Oscillatory Power Using DICS: From Raw MEG Data to Group-Level Statistics in Python," *Frontiers in Neuroscience*, vol. 12, 2018. doi: 10.3389/fnins.2018.00586

[168] M. Siegel, T. H. Donner, and A. K. Engel, "Spectral fingerprints of large-scale neuronal interactions," *Nature Reviews Neuroscience*, vol. 13, no. 2, pp. 121–134, Feb. 2012. doi: 10.1038/nrn3137

[169] M. Bola and B. A. Sabel, "Dynamic reorganization of brain functional networks during cognition," *NeuroImage*, vol. 114, pp. 398–413, Jul. 2015. doi: 10.1016/j.neuroimage.2015.03.057

[170] J.-P. Lachaux, E. Rodriguez, J. Martinerie, F. J. Varela *et al.*, "Measuring phase synchrony in brain signals," *Human brain mapping*, vol. 8, no. 4, pp. 194–208, 1999.

[171] K. Kumar, M. Toews, L. Chauvin, O. Colliot, and C. Desrosiers, "Multi-modal brain fingerprinting: A manifold approximation based framework," *NeuroImage*, vol. 183, pp. 212–226, Dec. 2018. doi: 10.1016/j.neuroimage.2018.08.006

[172] F. M. Krienen, B. T. T. Yeo, and R. L. Buckner, "Reconfigurable task-dependent functional coupling modes cluster around a core functional architecture," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1653, p. 20130526, Sep. 2014. doi: 10.1098/rstb.2013.0526

[173] R. Ciric, J. S. Nomi, L. Q. Uddin, and A. B. Satpute, "Contextual connectivity: A framework for understanding the intrinsic dynamic architecture of large-scale functional brain networks," *Scientific Reports*, vol. 7, no. 1, p. 6537, Jul. 2017. doi: 10.1038/s41598-017-06866-w

[174] K. Sekihara and S. S. Nagarajan, *Adaptive Spatial Filters for Electromagnetic Brain Imaging*, ser. Series in Biomedical Engineering. Berlin Heidelberg: Springer-Verlag, 2008.

[175] K. Amunts, A. Schleicher, and K. Zilles, "Cytoarchitecture of the cerebral cortex–more than localization," *NeuroImage*, vol. 37, no. 4, pp. 1061–1065; discussion 1066–1068, Oct. 2007. doi: 10.1016/j.neuroimage.2007.02.037

[176] A. Schnitzler and J. Gross, "Normal and pathological oscillatory communication in the brain," *Nature Reviews. Neuroscience*, vol. 6, no. 4, pp. 285–296, Apr. 2005. doi: 10.1038/nrn1650

[177] P. J. Uhlhaas and W. Singer, "Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology," *Neuron*, vol. 52, no. 1, pp. 155–168, Oct. 2006. doi: 10.1016/j.neuron.2006.09.020

[178] ——, "Neuronal dynamics and neuropsychiatric disorders: Toward a translational paradigm for dysfunctional large-scale networks," *Neuron*, vol. 75, no. 6, pp. 963–980, Sep. 2012. doi: 10.1016/j.neuron.2012.09.004

[179] F. de Pasquale, S. Della Penna, A. Z. Snyder, C. Lewis, D. Mantini, L. Marzetti, P. Belardinelli, L. Ciancetta, V. Pizzella, G. L. Romani, and M. Corbetta, "Temporal dynamics of spontaneous MEG activity in brain networks," *Proceedings of the National Academy of Sciences*, vol. 107, no. 13, pp. 6040–6045, Mar. 2010. doi: 10.1073/pnas.0913863107

[180] A. Hillebrand, G. R. Barnes, J. L. Bosboom, H. W. Berendse, and C. J. Stam, "Frequency-dependent functional connectivity within resting-state networks: An atlas-based MEG beamformer solution," *NeuroImage*, vol. 59, no. 4, pp. 3909–3921, Feb. 2012. doi: 10.1016/j.neuroimage.2011.11.005

[181] D. Mantini, M. G. Perrucci, C. Del Gratta, G. L. Romani, and M. Corbetta, "Electrophysiological signatures of resting state networks in the human brain," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 32, pp. 13 170–13 175, Aug. 2007. doi: 10.1073/pnas.0700668104

[182] E. Florin and S. Baillet, "The brain's resting-state activity is shaped by synchronized cross-frequency coupling of neural oscillations," *NeuroImage*, vol. 111, pp. 26–35, May 2015. doi: 10.1016/j.neuroimage.2015.01.054

[183] R. Scheeringa, K. M. Petersson, A. Kleinschmidt, O. Jensen, and M. C. Bastiaansen, "EEG Alpha Power Modulation of fMRI Resting-State Connectivity," *Brain Connectivity*, vol. 2, no. 5, pp. 254–264, Oct. 2012. doi: 10.1089/brain.2012.0088

[184] A. V. Sazonov, C. K. Ho, J. W. M. Bergmans, J. B. A. M. Arends, P. A. M. Griep, E. A. Verbitskiy, P. J. M. Cluitmans, and P. A. J. M. Boon, "An investigation of the phase locking index for measuring of interdependency of cortical source signals recorded in the EEG," *Biological Cybernetics*, vol. 100, no. 2, pp. 129–146, Feb. 2009. doi: 10.1007/s00422-008-0283-4

[185] F. Mormann, K. Lehnertz, P. David, and C. E. Elger, "Mean phase coherence as a measure for phase synchronization and its application to the EEG of epilepsy

patients," *Physica D: Nonlinear Phenomena*, vol. 144, no. 3-4, pp. 358–369, Oct. 2000. doi: 10.1016/S0167-2789(00)00087-7

[186] M. E. Raichle, "The restless brain: How intrinsic activity organizes brain function," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 370, no. 1668, p. 20140172, May 2015. doi: 10.1098/rstb.2014.0172

[187] ——, "The restless brain," *Brain Connectivity*, vol. 1, no. 1, pp. 3–12, 2011. doi: 10.1089/brain.2011.0019

[188] G. Deco, V. K. Jirsa, and A. R. McIntosh, "Emerging concepts for the dynamical organization of resting-state activity in the brain," *Nature Reviews Neuroscience*, vol. 12, no. 1, pp. 43–56, Jan. 2011. doi: 10.1038/nrn2961

[189] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain," *NeuroImage*, vol. 15, no. 1, pp. 273–289, Jan. 2002. doi: 10.1006/nimg.2001.0978

[190] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996. doi: 10.1007/BF00058655

[191] B. Hutcheon and Y. Yarom, "Resonance, oscillation and the intrinsic frequency preferences of neurons," *Trends in Neurosciences*, vol. 23, no. 5, pp. 216–222, May 2000. doi: 10.1016/s0166-2236(00)01547-2

[192] H. Gudi-Mindermann, J. M. Rimmele, G. Nolte, P. Bruns, A. K. Engel, and B. Röder, "Working memory training in congenitally blind individuals results in an integration of occipital cortex in functional networks," *Behavioural Brain Research*, vol. 348, pp. 31–41, Aug. 2018. doi: 10.1016/j.bbr.2018.04.002

[193] J. M. Rimmele, H. Gudi-Mindermann, G. Nolte, B. Röder, and A. K. Engel, "Working memory training integrates visual cortex into beta-band networks in congenitally blind individuals," *NeuroImage*, vol. 194, pp. 259–271, Jul. 2019. doi: 10.1016/j.neuroimage.2019.03.003

[194] F. Karayanidis, M. C. Keuken, A. Wong, J. L. Rennie, G. de Hollander, P. S. Cooper, W. Ross Fulham, R. Lenroot, M. Parsons, N. Phillips, P. T. Michie, and B. U. Forstmann, "The Age-ility Project (Phase 1): Structural and functional

imaging and electrophysiological data repository," *NeuroImage*, vol. 124, no. Pt B, pp. 1137–1142, Jan. 2016. doi: 10.1016/j.neuroimage.2015.04.047

[195] M. X. Cohen, *Analyzing Neural Time Series Data: Theory and Practice*, ser. Issues in Clinical and Cognitive Neuropsychology. Cambridge, Massachusetts: The MIT Press, 2014.

[196] R. Oostenveld, P. Fries, E. Maris, and J.-M. Schoffelen, "FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data," *Computational Intelligence and Neuroscience*, vol. 2011, p. e156869, Dec. 2010. doi: 10.1155/2011/156869

[197] A. Delorme and S. Makeig, "EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9–21, Mar. 2004. doi: 10.1016/j.jneumeth.2003.10.009

[198] P. Sanz Leon, S. A. Knock, M. M. Woodman, L. Domide, J. Mersmann, A. R. McIntosh, and V. Jirsa, "The Virtual Brain: A simulator of primate brain network dynamics," *Frontiers in Neuroinformatics*, vol. 7, 2013. doi: 10.3389/fninf.2013.00010

[199] A. C. Davison and P. Hall, "On the bias and variability of bootstrap and cross-validation estimates of error rate in discrimination problems," *Biometrika*, vol. 79, no. 2, pp. 279–284, Jun. 1992. doi: 10.1093/biomet/79.2.279

[200] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, and K. Ugurbil, "The WU-Minn Human Connectome Project: An Overview," *NeuroImage*, vol. 80, pp. 62–79, Oct. 2013. doi: 10.1016/j.neuroimage.2013.05.041

[201] N. Chonacky and D. Winch, "Reviews of Maple, Mathematica, and Matlab: Coming Soon to a Publication Near You," *Computing in Science Engineering*, vol. 7, no. 2, pp. 9–10, Mar. 2005. doi: 10.1109/MCSE.2005.39

[202] M. X. Cohen, *MATLAB for Brain and Cognitive Scientists*. Cambridge, Massachusetts: MIT Press, 12 maja 2017.

[203] P. Wallisch, M. E. Lusignan, M. D. Benayoun, T. I. Baker, A. S. Dickey, and N. G. Hatsopoulos, *MATLAB for Neuroscientists: An Introduction to Scientific Computing in MATLAB*, 2nd ed. Amsterdam: Academic Press, Dec. 2013.

[204] K. J. Blinowska-Cieślak and J. Żygierewicz, *Practical Biomedical Signal Analysis Using Matlab*, ser. Series in Medical Physics and Biomedical Engineering. Boca Raton [etc.]: CRC Press / Taylor & Francis Group, 2012.

[205] S. Heitmann, M. J. Aburn, and M. Breakspear, "The Brain Dynamics Toolbox for Matlab," *Neurocomputing*, vol. 315, pp. 82–88, Nov. 2018. doi: 10.1016/j.neucom.2018.06.026

[206] W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols, Eds., *Statistical Parametric Mapping: The Analysis of Functional Brain Images*, 1st ed. Amsterdam ; Boston: Academic Press, Nov. 2006.

[207] F. Tadel, S. Baillet, J. C. Mosher, D. Pantazis, and R. M. Leahy, "Brainstorm: A user-friendly application for MEG/EEG analysis," *Computational Intelligence and Neuroscience*, vol. 2011, p. 879716, 2011. doi: 10.1155/2011/879716

[208] "IEEE Standard for Floating-Point Arithmetic," *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, Jul. 2019. doi: 10.1109/IEEESTD.2019.8766229

[209] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Computing Surveys*, vol. 23, no. 1, pp. 5–48, Mar. 1991. doi: 10.1145/103162.103163

[210] C. Newham, *Learning the Bash Shell: Unix Shell Programming*, 3rd ed. Beijing ; Sesbastopol, Calif.: O'Reilly Media, Apr. 2005.

[211] D. Harel and Y. Feldman, *Algorithmics: The Spirit of Computing*, 3rd ed. Harlow, Essex, England ; New York: Addison-Wesley, Jun. 2004.

[212] M. F. Glasser, S. N. Sotiropoulos, J. A. Wilson, T. S. Coalson, B. Fischl, J. L. Andersson, J. Xu, S. Jbabdi, M. Webster, J. R. Polimeni, D. C. Van Essen, and M. Jenkinson, "The minimal preprocessing pipelines for the Human Connectome Project," *NeuroImage*, vol. 80, pp. 105–124, Oct. 2013. doi: 10.1016/j.neuroimage.2013.04.127

[213] D. Marcus, J. Harwell, T. Olsen, M. Hodge, M. Glasser, F. Prior, M. Jenkinson, T. Laumann, S. Curtiss, and D. Van Essen, "Informatics and Data Mining Tools and Strategies for the Human Connectome Project," *Frontiers in Neuroinformatics*, vol. 5, 2011. doi: 10.3389/fninf.2011.00004

[214] D. Slepian, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty—V: The Discrete Case," *Bell System Technical Journal*, vol. 57, no. 5, pp. 1371–1430, 1978. doi: 10.1002/j.1538-7305.1978.tb02104.x

[215] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed. Upper Saddle River, N.J: Pearson, Apr. 2006.

[216] D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications*. Cambridge: Cambridge University Press, 1993.

[217] A. Schaefer, R. Kong, E. M. Gordon, T. O. Laumann, X.-N. Zuo, A. J. Holmes, S. B. Eickhoff, and B. T. T. Yeo, "Local-Global Parcellation of the Human Cerebral Cortex from Intrinsic Functional Connectivity MRI," *Cerebral Cortex (New York, N.Y.: 1991)*, vol. 28, no. 9, pp. 3095–3114, Sep. 2018. doi: 10.1093/cercor/bhx179

[218] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, M. S. Albert, and R. J. Killiany, "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest," *NeuroImage*, vol. 31, no. 3, pp. 968–980, Jul. 2006. doi: 10.1016/j.neuroimage.2006.01.021

[219] M. Weiß and M. Göker, "Chapter 12 - Molecular Phylogenetic Reconstruction," in *The Yeasts (Fifth Edition)*, C. P. Kurtzman, J. W. Fell, and T. Boekhout, Eds. London: Elsevier, Jan. 2011, pp. 159–174.

[220] D. T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining: 4*. Hoboken: Wiley, 11 lipca 2014.

[221] D. H. Greene and D. E. Knuth, *Mathematics for the Analysis of Algorithms*, 3rd ed. Boston: Birkhäuser, Oct. 2007.

[222] T. H. Cormen, R. L. Rivest, C. E. Leiserson, and C. Stein, *Introduction to Algorithms, 3rd Edition*, 3rd ed. Cambridge, Mass: MIT Press, Sep. 2009.

[223] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. Cambridge, UK ; New York: Cambridge University Press, Sep. 2007.

[224] N. Chapin, "Flowcharting With the ANSI Standard: A Tutorial," *ACM Computing Surveys*, vol. 2, no. 2, pp. 119–146, Jun. 1970. doi: 10.1145/356566.356570

[225] J. F. Claerbout and M. Karrenbach, "Electronic documents give reproducible research a new meaning," in *SEG Technical Program Expanded Abstracts 1992*, ser. SEG Technical Program Expanded Abstracts. Society of Exploration Geophysicists, Jan. 1992, pp. 601–604.

[226] S. N. Goodman, D. Fanelli, and J. P. A. Ioannidis, "What does research reproducibility mean?" *Science Translational Medicine*, vol. 8, no. 341, pp. 341ps12–341ps12, Jun. 2016. doi: 10.1126/scitranslmed.aaf5027

[227] J. Samogin, Q. Liu, M. Marino, N. Wenderoth, and D. Mantini, "Shared and connection-specific intrinsic interactions in the default mode network," *NeuroImage*, vol. 200, pp. 474–481, Oct. 2019. doi: 10.1016/j.neuroimage.2019.07.007

[228] C. D. Hacker, A. Z. Snyder, M. Pahwa, M. Corbetta, and E. C. Leuthardt, "Frequency-specific electrophysiologic correlates of resting state fMRI networks," *NeuroImage*, vol. 149, pp. 446–457, Apr. 2017. doi: 10.1016/j.neuroimage.2017.01.054

[229] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, Nov. 2000. doi: 10.1023/A:1026543900054

[230] M. Demirtaş, J. B. Burt, M. Helmer, J. L. Ji, B. D. Adkinson, M. F. Glasser, D. C. Van Essen, S. N. Sotiropoulos, A. Anticevic, and J. D. Murray, "Hierarchical Heterogeneity across Human Cortex Shapes Large-Scale Neural Dynamics," *Neuron*, vol. 101, no. 6, pp. 1181–1194.e13, Mar. 2019. doi: 10.1016/j.neuron.2019.01.017

[231] T. Xu, K.-H. Nenning, E. Schwartz, S.-J. Hong, J. T. Vogelstein, A. Goulas, D. A. Fair, C. E. Schroeder, D. S. Margulies, J. Smallwood, M. P. Milham, and G. Langs, "Cross-species functional alignment reveals evolutionary hierarchy within the connectome," *NeuroImage*, vol. 223, p. 117346, Dec. 2020. doi: 10.1016/j.neuroimage.2020.117346

[232] R. A. Bethlehem, C. Paquola, J. Seidlitz, L. Ronan, B. Bernhardt, C.-C. Consortium, and K. A. Tsvetanov, "Dispersion of functional gradients across the adult lifespan," *NeuroImage*, vol. 222, p. 117299, Nov. 2020. doi: 10.1016/j.neuroimage.2020.117299

[233] B.-y. Park, R. Vos de Wael, C. Paquola, S. Larivière, O. Benkarim, J. Royer, S. Tavakol, R. R. Cruces, Q. Li, S. L. Valk, D. S. Margulies, B. Mišić, D. Bzdok,

J. Smallwood, and B. C. Bernhardt, "Signal diffusion along connectome gradients and inter-hub routing differentially contribute to dynamic human brain function," *NeuroImage*, vol. 224, p. 117429, Jan. 2021. doi: 10.1016/j.neuroimage.2020.117429

[234] P. Friedrich, S. J. Forkel, and M. Thiebaut de Schotten, "Mapping the principal gradient onto the corpus callosum," *NeuroImage*, vol. 223, p. 117317, Dec. 2020. doi: 10.1016/j.neuroimage.2020.117317

[235] N. Cross, C. Paquola, F. B. Pomares, A. A. Perrault, A. Jegou, A. Nguyen, U. Aydin, B. C. Bernhardt, C. Grova, and T. T. Dang-Vu, "Cortical gradients of functional connectivity are robust to state-dependent changes following sleep deprivation," *NeuroImage*, vol. 226, p. 117547, Feb. 2021. doi: 10.1016/j.neuroimage.2020.117547

[236] P. Haueis, "Multiscale modeling of cortical gradients: The role of mesoscale circuits for linking macro- and microscale gradients of cortical organization and hierarchical information processing," *NeuroImage*, vol. 232, p. 117846, May 2021. doi: 10.1016/j.neuroimage.2021.117846

[237] R. Viviani, L. Dommes, J. E. Bosch, and K. Labek, "Segregation, connectivity, and gradients of deactivation in neural correlates of evidence in social decision making," *NeuroImage*, vol. 223, p. 117339, Dec. 2020. doi: 10.1016/j.neuroimage.2020.117339

[238] J. Zhang, O. Abiose, Y. Katsumi, A. Touroutoglou, B. C. Dickerson, and L. F. Barrett, "Intrinsic Functional Connectivity is Organized as Three Interdependent Gradients," *Scientific Reports*, vol. 9, no. 1, p. 15976, Nov. 2019. doi: 10.1038/s41598-019-51793-7

[239] F. M. Krienen, B. T. T. Yeo, T. Ge, R. L. Buckner, and C. C. Sherwood, "Transcriptional profiles of supragranular-enriched genes associate with corticocortical network architecture in the human brain," *Proceedings of the National Academy of Sciences*, vol. 113, no. 4, pp. E469–E478, Jan. 2016. doi: 10.1073/pnas.1510903113

[240] K. Wagstyl, L. Ronan, I. M. Goodyer, and P. C. Fletcher, "Cortical thickness gradients in structural hierarchies," *NeuroImage*, vol. 111, pp. 241–250, May 2015. doi: 10.1016/j.neuroimage.2015.02.036

[241] M. F. Glasser and D. C. Van Essen, "Mapping human cortical areas in vivo based on myelin content as revealed by T1- and T2-weighted MRI," *The Journal of*

Neuroscience: The Official Journal of the Society for Neuroscience, vol. 31, no. 32, pp. 11 597–11 616, Aug. 2011. doi: 10.1523/JNEUROSCI.2180-11.2011

[242] L. J. Larson-Prior, R. Oostenveld, S. Della Penna, G. Michalareas, F. Prior, A. Babajani-Feremi, J. M. Schoffelen, L. Marzetti, F. de Pasquale, F. Di Pompeo, J. Stout, M. Woolrich, Q. Luo, R. Bucholz, P. Fries, V. Pizzella, G. L. Romani, M. Corbetta, and A. Z. Snyder, "Adding dynamics to the Human Connectome Project with MEG," *NeuroImage*, vol. 80, pp. 190–201, Oct. 2013. doi: 10.1016/j.neuroimage.2013.05.056

[243] D. Slepian and H. O. Pollak, "Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty — I," *Bell System Technical Journal*, vol. 40, no. 1, pp. 43–63, 1961. doi: 10.1002/j.1538-7305.1961.tb03976.x

[244] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, Jul. 2006. doi: 10.1016/j.acha.2006.04.006

[245] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7431, May 2005. doi: 10.1073/pnas.0500334102

[246] S. T. Rachev, "The Monge–Kantorovich Mass Transference Problem and Its Stochastic Applications," *Theory of Probability & Its Applications*, vol. 29, no. 4, pp. 647–676, Jan. 1985. doi: 10.1137/1129093

[247] F. L. Hitchcock, "The Distribution of a Product from Several Sources to Numerous Localities," *Journal of Mathematics and Physics*, vol. 20, no. 1-4, pp. 224–230, 1941. doi: 10.1002/sapm1941201224

[248] L. V. Kantorovich, "Mathematical Methods of Organizing and Planning Production," *Management Science*, vol. 6, no. 4, pp. 366–422, Jul. 1960. doi: 10.1287/mnsc.6.4.366

[249] G. Langs, P. Golland, and S. S. Ghosh, "Predicting Activation Across Individuals with Resting-State Functional Connectivity Based Multi-Atlas Label Fusion," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-24571-3_38 pp. 313–320.

[250] G. Langs, A. Sweet, D. Lashkari, Y. Tie, L. Rigolo, A. J. Golby, and P. Golland, "Decoupling function and anatomy in atlases of functional connectivity patterns: Language mapping in tumor patients," *NeuroImage*, vol. 103, pp. 462–475, Dec. 2014. doi: 10.1016/j.neuroimage.2014.08.029

[251] G. Langs, B. H. Menze, D. Lashkari, and P. Golland, "Detecting stable distributed patterns of brain activation using Gini contrast," *NeuroImage*, vol. 56, no. 2, pp. 497–507, May 2011. doi: 10.1016/j.neuroimage.2010.07.074

[252] G. Langs, D. Lashkari, A. Sweet, Y. Tie, L. Rigolo, A. J. Golby, and P. Golland, "Learning an Atlas of a Cognitive Process in Its Functional Geometry," in *Information Processing in Medical Imaging*, ser. Lecture Notes in Computer Science, G. Székely and H. K. Hahn, Eds. Berlin, Heidelberg: Springer, 2011. doi: 10.1007/978-3-642-22092-0_12 pp. 135–146.

[253] G. Langs, P. Golland, Y. Tie, L. Rigolo, and A. J. Golby, "Functional Geometry Alignment and Localization of Brain Areas," *Advances in neural information processing systems*, vol. 1, pp. 1225–1233, 2010.

[254] B. Fischl, "FreeSurfer," *NeuroImage*, vol. 62, no. 2, pp. 774–781, Aug. 2012. doi: 10.1016/j.neuroimage.2012.01.021

[255] B. Mckeown, W. H. Strawson, H.-T. Wang, T. Karapanagiotidis, R. Vos de Wael, O. Benkarim, A. Turnbull, D. Margulies, E. Jefferies, C. McCall, B. Bernhardt, and J. Smallwood, "The relationship between individual variation in macroscale functional gradients and distinct aspects of ongoing thought," *NeuroImage*, vol. 220, p. 117072, Oct. 2020. doi: 10.1016/j.neuroimage.2020.117072

[256] J. A. Brissenden and D. C. Somers, "Cortico–cerebellar networks for visual attention and working memory," *Current Opinion in Psychology*, vol. 29, pp. 239–247, Oct. 2019. doi: 10.1016/j.copsyc.2019.05.003

[257] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 19:1–19:21, Jul. 2017. doi: 10.1145/3068335

[258] N. Jankowski, "Revdbscan and Flexscan—O(nlog n) Clustering Algorithms," in *Neural Information Processing*, ser. Communications in Computer and Information Science, T. Mantoro, M. Lee, M. A. Ayu, K. W. Wong, and A. N. Hidayanto, Eds. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-92307-5_75 pp. 642–650.

[259] C. M. Michel and T. Koenig, "EEG microstates as a tool for studying the temporal dynamics of whole-brain neuronal networks: A review," *NeuroImage*, vol. 180, pp. 577–593, Oct. 2018. doi: 10.1016/j.neuroimage.2017.11.062

[260] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5.1, pp. 281–298, Jan. 1967.

[261] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987. doi: 10.1016/0377-0427(87)90125-7

[262] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake, "Finite Mixture Models," *Annual Review of Statistics and Its Application*, vol. 6, no. 1, pp. 355–378, 2019. doi: 10.1146/annurev-statistics-031017-100325