École Polytechnique Fédérale de Lausanne

Centre Hospitalier Universitaire Vaudois

Master Thesis

Investigating Brain Parcellation with Resting State fMRI at Subject and Group Levels

Robin Junod

Dr. Michel Akselrod
Supervisor

Prof. Dimitri Van De Ville
Thesis Advisor

6th July 2025

Abstract

Brain parcellation is a fundamental technique in neuroimaging, used to reduce complex brain data into meaningful regions. While most parcellation efforts focus on group-level analyses, individualized (subject-level) parcellation can capture person-specific variations in connectivity patterns, an aspect still underrepresented in the literature. In this work, I present a surface-based parcellation pipeline that uses resting-state functional MRI (rs-fMRI) data to produce parcels at both the group and subject levels.

The method relies on computing a similarity matrix, its gradient maps, and segmenting cortical surfaces through a watershed algorithm. I validate the approach against a null model of random parcellation and demonstrate notable gains in parcel homogeneity and dice coefficients (p-value < 0.001). Visual inspections confirm consistent boundary detection across hemispheres and multiple datasets. Notably, subject-level parcellations preserve finer-scale connectivity details when compared to group-level results, suggesting a potential path for more personalized analyses.

All code is available (open-source) in Python, facilitating adoption and reproducibility. While the results are constrained by limited computational resources, this pipeline provides a practical template for future work on individualized parcellation and network analyses in neuroimaging.

Contents

1 Introduction **4**
1.1 Context . . . 4
1.2 Origins . . . 4
1.3 Motivation . . . 5

2 Litterature review **6**
2.1 Hard parcellation . . . 6
2.2 Soft Parcellation . . . 8

3 Methods **9**
3.1 Data Acquisition and Preprocessing . . . 9
3.1.1 Dataset . . . 9
3.1.2 Minimal preprocessing and noise reduction . . . 10
3.2 Surface-Based Parcellation (Main Method) . . . 10
3.2.1 Surface Extraction . . . 10
3.2.2 Similarity Matrix Computation . . . 11
3.2.3 Gradient computation . . . 12
3.2.4 Watershed by flooding (boundary detection) . . . 12
3.2.5 Pipelines . . . 13
3.3 Parcel Evaluation . . . 14
3.3.1 Homogeneity . . . 14
3.3.2 Dice coefficient . . . 14
3.3.3 Null model . . . 15
3.3.4 Graph Theory . . . 15
3.4 Volume-Based Parcellation (Exploratory Attempt) . . . 15
3.4.1 Motivation . . . 15
3.4.2 Implementation . . . 16
3.4.3 Limitations . . . 16

4 Results **17**
4.1 Visual inspection . . . 17
4.1.1 Boundary Maps . . . 17
4.1.2 Parcellation Maps . . . 19
4.2 Parcel Evaluation . . . 20
4.2.1 Homogeneity . . . 20
4.2.2 Homogeneity, Subject-level Vs Group-level parcellation . . . 21
4.2.3 Dice coefficient metric . . . 22
4.2.4 Graph theory metrics . . . 23
4.3 Tuning of the model . . . 24
4.3.1 Similarity Matrix . . . 24

4.3.2 Role of Smoothing 24

5 Discussion **25**

5.1 Limitations 26

5.2 Future Directions and Applications 27

5.3 Supplementary material 27

5.4 Acknowledgment 27

6 Annex **30**

6.1 Pseudo codes 30

CHAPTER 1

INTRODUCTION

1.1 CONTEXT

Understanding the intricate organization of the human brain has been a central pursuit in neuroscience. Brain parcellation is the process of dividing the brain into distinct regions or parcels, each assumed to exhibit relatively homogeneous structural or functional characteristics. It is a fundamental step in many neuroimaging analyses, as it allows researchers to reduce the complexity of high-dimensional data and interpret the results in terms of localized or network-level patterns.

1.2 ORIGINS

Parcellation finds its roots in early 20th-century work by Korbinian Brodmann [1], a German neuroanatomist who meticulously examined the microstructure (cytoarchitecture) of the cerebral cortex. His influential studies culminated in the identification of 52 cortical regions, now known as Brodmann areas. They are based on differences in the cellular organization and layering of gray matter in the brain. Brodmann's systematic mapping was one of the first attempts to delineate the cortex into functionally distinct territories, which researchers still refer to, a century later.

Modern parcellation methods now integrate a wide range of neuroimaging modalities (e.g., MRI, fMRI, diffusion imaging) and computational techniques (e.g., clustering, graph theory, machine learning) to subdivide the brain more precisely. These newer approaches add layers of information about connectivity, function, and organization that were unattainable in Brodmann time.

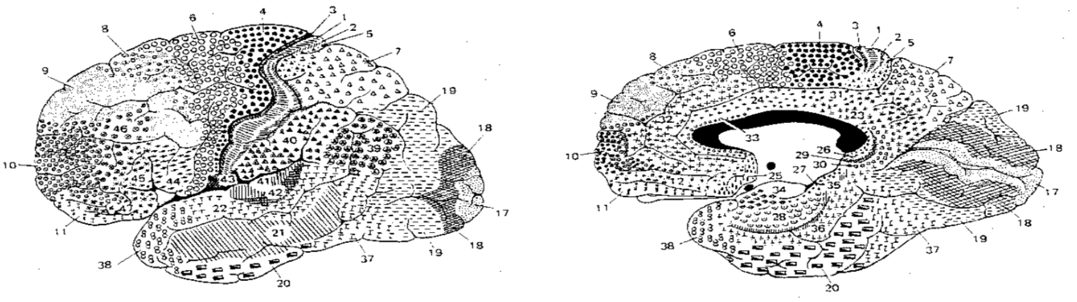


FIGURE 1.1
Lateral and Medial view of the Brodmann areas [1]

Broadly speaking, contemporary parcellations can be classified into two main categories:

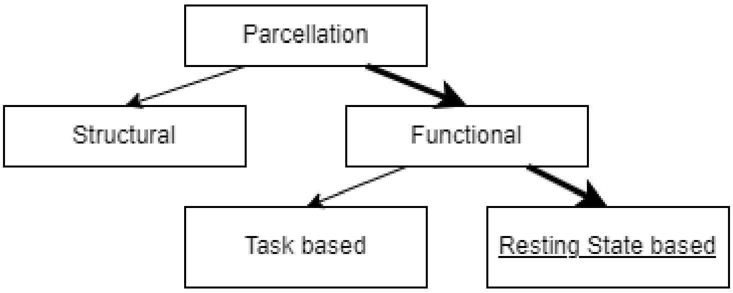
- Structural Parcellations: Rely on anatomical features such as cortical folding patterns or microstructural differences (Brodmann).
- Functional Parcellations: Use measurements of brain activity, often from resting-state or task-based fMRI, to cluster regions showing similar temporal dynamics.

1.3 MOTIVATION

Traditional brain parcellation methods have predominantly relied on task-based functional magnetic resonance imaging **(t-fMRI)**. In these approaches, participants engage in specific tasks designed to activate particular brain regions, enabling researchers to delineate functional areas based. While effective, t-fMRI requires active participation, which can be challenging for certain populations, including individuals with cognitive impairments or severe disabilities.

In contrast, resting-state fMRI **(rs-fMRI)** measures spontaneous brain activity when an individual is not engaged in any explicit task. This modality has emerged as a powerful alternative for brain parcellation due to several key advantages. Rs-fMRI is **more affordable and easier to perform**, eliminating the need for complex task creation. Moreover, it is **suitable for a broader range of individuals**, including those who may find task-based protocols difficult to perform. Finally, rs-fMRI **datasets are more abundant** than task-based fMRI datasets, with repositories such as the Human Connectome Project (HCP) providing extensive collections of rs-fMRI data.

In this study, we focus on parcellating the brain using resting-state fMRI data. Specifically, the goal is to create brain parcels by analyzing functional connectivity patterns observed during rest (RSFC). This involves examining how different brain regions interact and communicate in the absence of external task demands.



Chapter 2

Litterature review

This thesis focuses on parcellation derived from resting-state fMRI data. Consequently, I will not address structural or task-based parcellation methods. Instead, this section highlights the different approaches to **functional brain parcellation with resting state**.

Two principal approaches can be identified for the functional parcellation of the brain:

1. **Hard parcellations**, which segment the brain into non-overlapping regions, assigning each region to a single network.
2. **Soft parcellations**, which model the brain's organization as overlapping or continuous gradients, reflecting gradual transitions between functional areas.

In the sections below, I start by exploring hard parcellation methods, followed by soft parcellation approaches.

2.1 Hard parcellation

This approaches assume that the brain can be divided into distinct, **non-overlapping regions**, each associated with a single dominant network or function. In the context of resting-state fMRI (rs-fMRI), these methods typically rely on group-level functional connectivity data, identifying sharp boundaries between parcels based on the temporal correlation of neuronal activity. The advantage of hard parcellation lies in its relative simplicity: each parcel is treated as a functionally homogeneous unit, which simplifies interpretation and network analyses.

Historically, several influential parcellation atlases have emerged, each providing a unique window into large-scale functional organization. For instance, Power et al. (2011) [2] developed a widely cited parcellation by clustering rs-fMRI data into distinct nodes. Its success spurred further work, as researchers recognized that discrete nodes could facilitate functional network analyses, offering a relatively straightforward way to study brain-wide connectivity patterns. Around the same time, Yeo et al. (2011) [3] presented another parcellation, identifying 7- and 17-network divisions of the cortex. Their atlas captured large-scale functional networks—such as the default mode network, frontoparietal control networks, and sensory-motor networks—that have been extensively replicated and refined in subsequent research.

Building on these efforts, Wig et al. (2014) [4] introduced a data-driven approach explicitly designed to improve cross-individual alignment. Their method emphasized finding consistent functional boundaries that could, in principle, generalize across participants, while still respecting individual variability. Gordon et al. (2016) [5] further refined this methodology, ultimately creating an atlas of functionally coherent regions suitable for large-scale resting-state studies. Notably, both Wig and Gordon employed a combination of minimal preprocessing steps, surface reconstruction with FreeSurfer, and rs-fMRI data projection onto cortical surface meshes. Such a pipeline, largely standardized in the neuroimaging community, begins by cleaning the BOLD signals (e.g., through motion correction, slice timing correction, noise regression, and normalization). The T1-weighted structural MRI is then used to reconstruct the individual's cortical surface, enabling a more anatomically faithful representation of the brain than simple volumetric space.

The next and arguably most critical step in these pipelines involves computing the **similarity matrix**, where each vertex's time series on the cortical surface is correlated with every other voxel or vertex[1]. The resulting vertex-by-vertex matrix quantifies the degree of functional similarity across the cortex. After that, researchers typically calculate the **spatial gradient magnitude** of the similarity map to highlight boundaries and transitions between regions of similar connectivity.

From this gradient, a **boundary-detection** algorithm is employed to assign each vertex to a parcel. While Wig et al. used a non-maximal suppression-based approach to detect edges, Gordon et al. utilized a watershed-by-flooding algorithm to define parcels in the gradient space. Both strategies aim to identify the gradients boundary (area where the gradient is very high), ensuring that parcels form around local minima or maxima of connectivity change. This discrete segmentation yields parcels that encapsulate stable area within the cortex.

Beyond these well-cited works, other prominent atlases have emerged. Schaefer et al. (2018) [6] proposed a "local-global" approach for subdividing the cortex at multiple granularities, facilitating analyses across different scales. Kong et al. (2019)[7] demonstrated that individual-specific parcellations could capture meaningful variability in cognition, personality, and emotion, prompting broader interest in how best to accommodate person-specific functional maps within a group-level framework.

Overall, hard parcellation methods are popular because they produce robust, interpretable results that map well onto known functional circuits, and they offer a clear path for examining network interactions in health and disease. Their parcels can be easily integrated with common graph-theoretic measures (e.g., degree centrality or modularity), making them suitable for large-scale studies of functional organization. However, a recognized limitation is the potential loss of information about gradual transitions. The demarcation of rigid borders may overlook subtle gradients that, in reality, often span contiguous regions. As a result, there has been parallel growth in "soft" or continuous parcellation approaches, aiming to better capture these complex patterns.

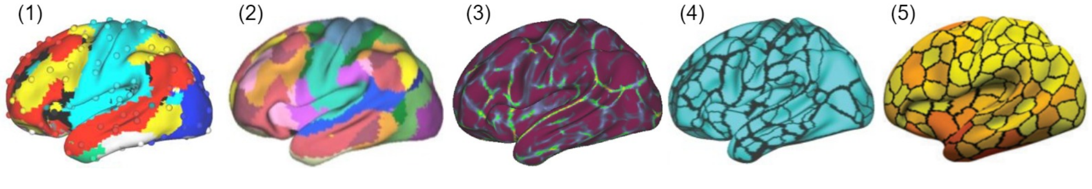


FIGURE 2.1
(1) Power et al. [2], (2) Yeo et al. [3], (3) Wig et al. [4], (4) Gordon et al. [5], (5) Schaeffer et al. [6]

[1] A vertex is a node on a surface mesh, while a voxel is a three-dimensional cube.

2.2 SOFT PARCELLATION

This family of methods posits that the brain should not be segmented into non-overlapping units. Instead, functional organization may shift gradually on the cortical surface, allowing single areas to serve multiple overlapping networks.

For instance, Allen et al. (2011) [8] applied Group ICA to resting-state fMRI data from over 600 participants, revealing a rich set of reproducible networks whose spatial maps showed substantial overlap. This finding underscores that distinct cognitive processes can converge within the same cortical territories.

Similarly, Haak et al. (2018) [9] employed resting-state functional connectivity, correlation measures, and graph-theoretical techniques to detect spatial eigenmodes. This allows to show how connectivity patterns transition smoothly across the cortex. Their method computes the Laplacian eigenvectors to identify gradual variations in several network modes.

These studies highlight the utility of soft parcellation approaches, which capture the nature of neural networks and reveal that regions often participate in multiple networks.

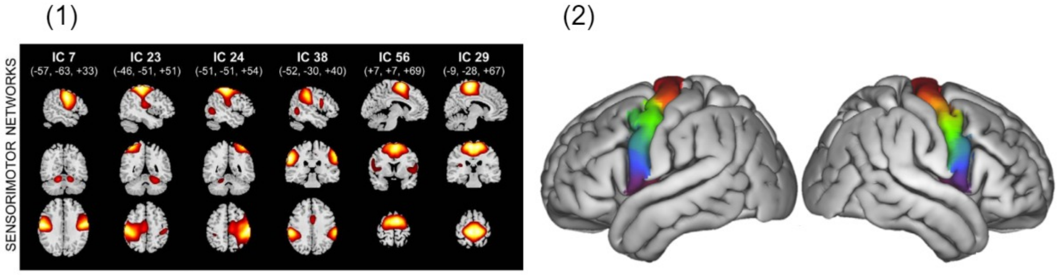


FIGURE 2.2
(1) Allen et al. [8], (2) Haak et al. [9]

CHAPTER 3

METHODS

In this section, I describe the brain parcellation method using resting-state fMRI data. First, I tried a volumetric approach based on 3D voxels, but the results were unsatisfactory. I then switched to a surface-based approach by using FreeSurfer to extract the cortical mesh. Inspired by the Gordon and Wig method [4, 5], I developed a parcellation pipeline that produced more promising results.

3.1 DATA ACQUISITION AND PREPROCESSING

3.1.1 DATASET

For this study, I used two functional MRI datasets —PPSFACE18 and PSFACE20— originally collected in this study [10]. These datasets consist of high-resolution fMRI scans from healthy, right-handed participants (see Table 3.1 for details).

	PPSFACE18	**PPSFACE20**
Total subjects	18	20
Female subjects	9	13
Mean age	24.5	28.45
Minimum age	18	22
Maximum age	41	48

TABLE 3.1

All MRI data were acquired using a Siemens Prisma **3T scanner** equipped with a 32-channel receiver/transmitter head coil. Functional images were acquired using a gradient echo-planar imaging (EPI) sequence to cover the entire brain with the following acquisition parameters: a repetition time **TR of 1000** ms, echo time TE of 32 ms, slice thickness of 2 mm, and 66 axial slices. The in-plane resolution was set to 2×2 mm^2, and a multi-slice acceleration factor of 6.

Each participant underwent four functional runs corresponding to the experimental task, with 380 volumes per run. Additionally, **two resting-state runs** were acquired, each lasting 5 minutes, one at the beginning and one at the end of the session. We will refere them as -Run1- and -Run2- respectively.

For anatomical reference, T1-weighted structural images were obtained using an MPRAGE sequence with a resolution of 1×1×1 mm^3, TR of 2000 ms, TE of 2.25 ms, and a flip angle of 8°. These structural MRI scans were processed with FreeSurfer's 'recon-all' tool [11], which automatically identifies different brain tissues and builds detailed cortical surface models.

3.1.2 MINIMAL PREPROCESSING AND NOISE REDUCTION

The preprocessing of functional MRI images was conducted using SPM12 (Wellcome Department of Cognitive Neurology, London, UK) [12] and the CONN toolbox [13].

Minimal preprocessing steps included motion correction, where rigid-body transformations were applied to realign each brain volume to the first volume and correct for head movement. Slice-timing correction was performed to account for temporal differences in slice acquisition order, followed by spatial smoothing with a 3 mm full width at half maximum (FWHM) Gaussian kernel, reducing high-frequency noise while preserving anatomical details. Functional images were then co-registered across runs to ensure spatial consistency. Finally a normalization step into the MNI152 space was performed.

Then, other preprocessing steps were performed using the CONN toolbox, which implements a robust pipeline for denoising and **removal of artifacts**. This included band-pass filtering (0.008–0.09 Hz) to remove low-frequency drift and high-frequency noise, as well as nuisance regression incorporating the average BOLD signal from white matter and cerebrospinal fluid (CSF) and the six estimated motion parameters. These steps effectively minimized physiological and motion-related artifacts, enhancing the reliability of the functional connectivity analysis.

By performing these preprocessing steps, the goal is to retain the most meaningful aspects of the BOLD signal while ensuring a standardized and reproducible pipeline for subsequent analyses.

3.2 SURFACE-BASED PARCELLATION (MAIN METHOD)

Since the results from volume-based parcellation were not satisfactory (see 3.4.3), I opted for a surface-based parcellation. This approach offers several advantages. First, it focuses specifically on the gray matter, minimizing contributions from other brain components such as cerebrospinal fluid and white matter.

Another key advantage is that surface-based methods preserve topological neighborhood relationships. In the brain's folded cortical structure, neighboring voxels in volumetric space do not always correspond to actual functional or neuronal neighbors. However, with surface-based extraction, adjacent vertices on the cortical surface are more likely to represent true anatomical and functional neighbors. Additionally, surface-based parcellation has been widely used in the literature and has demonstrated strong performance in previous studies.

The primary drawback of this method is the computational cost, as surface extraction requires extensive processing using FreeSurfer.

3.2.1 SURFACE EXTRACTION

After preprocessing the fMRI data with SPM [12] and removing artifacts using CONN [13], each participant's high-resolution anatomical MRI were processed with FreeSurfer's recon-all command [11]. This allows to accurately reconstruct the cortical surface, delineating the pial and white matter boundaries. These reconstructed surfaces were then used as targets for projecting the preprocessed fMRI data onto the subject's cortical sheet, thereby enabling surface-based analyses of functional connectivity.

Finally, these projected fMRI data are normalized into the fsaverage6 (an atlas from FreeSurfer) which will give us at the end, for each subject, 40,962 ($\sim 40k$) vertices per hemisphere.

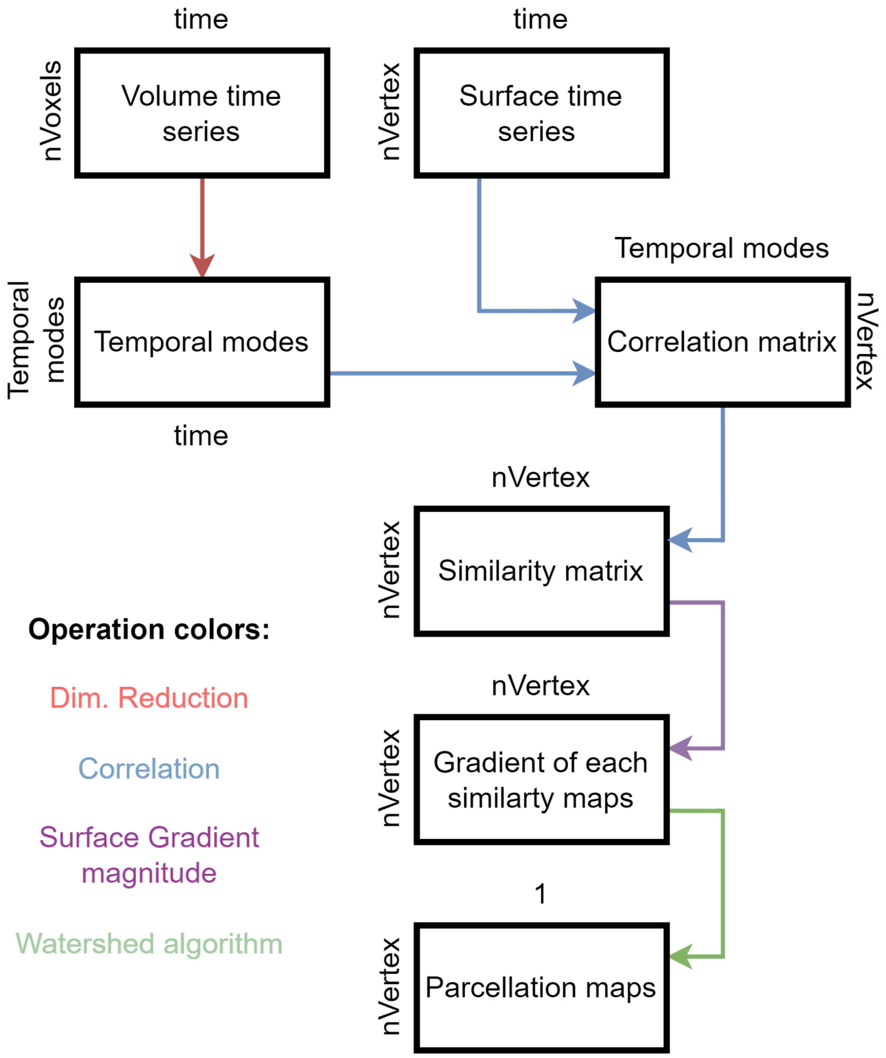


FIGURE 3.1
Overview of the pipeline

3.2.2 SIMILARITY MATRIX COMPUTATION

The similatry matrix is propabely the most critical steps in the parcellation pipeline because it quantifies the functional connectivity between surface vertices by measuring how correlated their time series are. In this implementation, measures were taken to manage computational and memory constraints while still capturing meaningful connectivity information.

Due to computational limitations, it was impossible to compute a huge correlation matrix as in [5] [4] paper. To address these constraints, I used PCA-based dimensionality reduction to extract the most relevant time courses. ICA was also experimented, but the results were less favorable.

PCA FOR DIMENSION REDUCTION OF VOXELS

The first step is to normalize the fMRI time series and applying Principal Component Analysis (PCA) to extract n dominant temporal modes from all voxel time series (only the first 17 components were kept). This dimensionality reduction proved essential for handling large datasets, as it enabled to avoid performing direct correlations on every single voxel time series, a step that would have been prohibitively time consuming with the computational resources available.

VERTEX-WISE CORRELATION

The PCA-derived temporal modes were correlated with each vertex's time series on the cortical surface using Pearson correlation. This produced an intermediate matrix with dimensions **nVertices** × **nTemporalModes**.

Next, an autocorrelation step is performed, comparing every vertex to every other vertex. This resulted in a square similarity matrix of size : **nVertices**×**nVertices**.

This matrix represents how similar the BOLD time series are across all pairs of vertices on the cortical surface.

SMOOTHING OF SIMILATRX MATRIX

To enhance spatial continuity and reduce noise, I applied smoothing to the similarity matrix. Specifically, 5 iterations of first neighborhood averaging were made for each column. The number of smoothing iterations acts as a hyperparameter that can be adjusted to tune the final number of parcels.

3.2.3 GRADIENT COMPUTATION

The goal of this step is to identify regions of rapid change in functional connectivity by computing the gradient magnitude of similarity values. Each column of the similarity matrix represents a similarity map, indicating how a given vertex is functionally related to all other vertices. Since there are 40,962 vertices per hemisphere, I generated 40,962 gradient maps per hemisphere.

To calculate the gradient magnitude at each vertex, I use the surface mesh extracted by FreeSurfer, which defines the neighborhood structure of the cortical surface. For each vertex v, the gradient is computed as the average root squared difference between its similarity value and those of its neighboring vertices:

$$G_i(v) = 1/|N(v)|\sqrt{\sum_{u \in N(v)} (S_v(v) - S_v(u))^2} \tag{3.1}$$

where:

- $G(v)$ is the gradient magnitude at vertex v for the similatry map i,
- $N(v)$ represents the set of neighboring vertices of v (as defined by the FreeSurfer mesh),
- $S_v(v)$ is the similarity value of vertex v in its similarity map,
- $S_v(u)$ is the similarity value of a neighboring vertex u,
- $|N(v)|$ is the number of neighboring vertices.

The pseudocode for this part is in the Annex 4.

SMOOTHING OF GRADIENT

Similar to the procedure applied to the similarity matrix, here an additional smoothing step is performed using the same iterative algorithm for 10 iterations.

3.2.4 WATERSHED BY FLOODING (BOUNDARY DETECTION)

The watershed by flooding algorithm is a computationally efficient method commonly used for image segmentation. It identifies regions ("basins") by simulating the process of water gradually filling valleys in an image.

For the parcellation, the watershed algorithm was adapted to work on the cortical surface mesh (a graph-based representation of the cortex). The gradient maps serve as a topographical representation, where regions of low gradient values correspond to functionally homogeneous areas. The watershed algorithm iteratively assigns vertices to parcels, ensuring that the segmentation follows natural connectivity boundaries.

A step-by-step pseudocode for the modified watershed algorithm is provided in the Annex 2.

The watershed algorithm is applied to each of the 40k gradient maps to detect local connectivity boundaries, generating a 40k × 40k boundary matrix.These individual boundary maps are summed to create a single boundary map (also referred to in the literature as an edge map), which has dimensions 40k × 1.

Finally, the watershed by flooding algorithm is used, one more time, on the bounardy map in order to generate the final parcellation.

One key advantage of the watershed algorithm is that it produces discrete, numbered parcels, with each vertex assigned to a unique parcel—unlike gradient maps, which only provide a continuous representation.

3.2.5 PIPELINES

The original goal was subject-level parcellation, but most studies in the literature use group-level parcellation [4, 5, 14, 15]. The following part presents both approaches.

SUBJECT-LEVEL PARCELS

1. Data Preprocessing
2. Computation of the Similarity Matrix (per hemisphere)
3. Smoothing of the Similarity Matrix (40k × 40k)
4. Gradient Computation (40k × 40k)
5. Boundary Computation (40k × 40k)
6. Sum of Boundary (40k × 1)
7. Smoothing of the Boundary Sum (40k × 1)
8. Watershed by Flooding (final parcellation, 40k × 1)

GROUP-LEVEL PARCELS

1. Data Preprocessing
2. Computation of the Similarity Matrix (per subject and hemisphere)
3. Smoothing of the Similarity Matrix (40k × 40k)
4. Averaging of Similarity Matrices Across Subjects
5. Gradient Computation on the Group-Averaged Similarity Matrix (40k × 40k)
6. Boundary Computation (40k × 40k)
7. Sum of Boundary (40k × 1)
8. Smoothing of the Boundary Sum (40k × 1)
9. Watershed by Flooding (final parcellation, 40k × 1)

3.3 Parcel Evaluation

3.3.1 Homogeneity

The above procedure creates parcels by identifying strong boundaries where connectivity patterns differ greatly between neighboring cortical regions. However, a well-defined parcel should not only be distinct from its neighbors, but also maintain a single, consistent connectivity pattern inside the parcel. This means its connectivity should be homogeneous throughout the parcel.

I used the Craddock et al. [16] homogeneity metric to measure within-cluster functional similarity. It groups vertices into parcels and calculates the average pairwise correlation of their time series within each parcel, giving each parcel a homogeneity value between 0 and 1. The overall parcellation homogeneity is then defined as the average of these parcel's homogeneity.

Let C be a cluster with n vertex. Denote the time series for vertex i by

$$\mathbf{x}_i = \big(x_i(1), x_i(2), \ldots, x_i(T)\big),$$

where T is the number of time points.

The Pearson correlation coefficient between vertices i and j is given by:

$$r(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{t=1}^{T} \big(x_i(t) - \bar{x}_i\big)\big(x_j(t) - \bar{x}_j\big)}{\sqrt{\sum_{t=1}^{T} \big(x_i(t) - \bar{x}_i\big)^2} \sqrt{\sum_{t=1}^{T} \big(x_j(t) - \bar{x}_j\big)^2}},$$

where $\bar{x}_i$ and $\bar{x}_j$ are the means of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Higher within-parcel correlation means higher homogeneity.

The cluster homogeneity, $H(C)$, is then defined as the mean of these correlation values across all vertex pairs in C:

$$H(C) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} r(\mathbf{x}_i, \mathbf{x}_j).$$

For k parcels/clusters the average homogeneity for the parcellation would be :

$$AverageHomogeneity = \frac{1}{k} \sum_{i=1}^{k} H(C_i)$$

3.3.2 Dice coefficient

The other well-used method in the literature to compare different parcellations is the dice coefficient.

It can be mathematically defined as:

$$\text{Dice}(A, B) = \frac{2\,|A \cap B|}{|A| + |B|},$$

where A and B are sets, $|A|$ and $|B|$ denote the sizes of these sets, and $|A \cap B|$ is the size of their intersection.

In this case, I used the dice coefficient due to its popularity among parcellation and image segmentation studies [5, 14, 15]. It provides a straightforward, interpretable metric for comparing how well two parcellations overlap in terms of their respective boundaries or assigned regions. I used it to compare parcellations from different runs and across different datasets. The higher the dice coefficient, the better the overlap.

3.3.3 NULL MODEL

A null model is used to evaluate the performance of the parcellation. In this case, the null model randomly places n seeds, which then expand in a manner similar to the 'watershed by flooding' algorithm, where each parcel grows until it meets an adjacent parcel.

It is crucial that the null model allows us to specify the number of parcels because, as pointed out by [5], both the number of parcels and the size of the parcel significantly affect metrics such as homogeneity and the dice coefficient. Although this null model differs from that of Gordon et al. [5], it still preserves the essential idea of having the same number of parcels, which facilitates fair comparisons between methods.

For details of the implementation, the pseudo-code can be found in the Annex 1.

3.3.4 GRAPH THEORY

Using the parcellations derived from the group analysis, I applied graph-theoretic measures—including clustering coefficient, shortest path length, modularity, and small-world properties—to characterize network organization.

For each participant, I averaged the time series within each parcel and computed pairwise correlations across all parcels, generating an $n_{\text{parcels}} \times n_{\text{parcels}}$ correlation matrix.

To focus on the strongest connections, only the top 20% of correlations were retained and converted into a undirected and unweighted graph. I then calculated the mentioned metrics on each participant's graph and summarized them at the group level by computing the mean and standard deviation across participants.

3.4 VOLUME-BASED PARCELLATION (EXPLORATORY ATTEMPT)

3.4.1 MOTIVATION

This first approach aimed to apply a gradient-based resting-state functional connectivity (RSFC) algorithm directly in three-dimensional voxel space, inspired by Wig's suggestion that such methods could naturally extend beyond surfaces.

> “ *The gradient-based strategy for finding RSFC pattern transitions can naturally be extended into 3-dimensions for this purpose, though we do not present such an approach here.* ”
>
> Wig et al. [4]

Although theoretically sound, this method did not yield satisfactory results, underscoring the difficulty of performing voxelwise parcellations without projecting onto a cortical surface (see limitations 3.4.3).

3.4.2 IMPLEMENTATION

The pipeline for the 3D voxel-based parcellation followed a similar structure to the surface-based approach, drawing inspiration from the methods outlined in Gordon et al. (2016) [5] and Haak et al. (2018) [9]. Additionally, the similarity matrix computation was based on the strategy used in the Congrads paper, but instead of applying it to the entire cortex, I focused on a single cortical region (e.g., the primary somatosensory cortex).

To manage computational complexity, I restrained the similarity matrix calculation to a specific region of interest(ROI), significantly reducing the number of voxels involved.

Next, I computed the gradient magnitude for each voxel in the ROI. This step involved comparing each voxel's connectivity pattern with its spatial neighbors.

To perform parcellation, I extended the watershed by flooding algorithm, which is traditionally applied in 2D image segmentation, to a 3D voxel-based space.

The goal of this approach was to create a fully voxel-wise parcellation, relying solely on BOLD response similarity without requiring projection onto a cortical surface. However, because the method operated in 3D space, ensuring biologically meaningful and functionally coherent parcellations proved to be highly challenging.

3.4.3 LIMITATIONS

Despite its theoretical feasibility, the watershed algorithm did not perform well in 3D space, making it unsuitable for practical application. Although the algorithm produced some parcellation results, the parcels were very fragmented and noisy.

Key limitations included:

- Excessive Noise in 3D Parcels: The lack of topological constraints led to disconnected and highly irregular parcellation patterns.
- Lack of Cortical Folding Awareness: The gradient computation was purely spatial and did not take into account the pial surface, meaning it segmented voxels based purely on Euclidean distance rather than true functional neighborhoods.
- Computational Constraints: While limiting the analysis to a single cortical region reduced the size of the similarity matrix, applying watershed segmentation in 3D remained computationally intensive and inefficient.

Given these limitations, the approach did not yield sufficiently robust or interpretable results. The implementation code is available on GitHub.

CHAPTER 4

RESULTS

4.1 VISUAL INSPECTION

A simple visual inspection serves as a straightforward initial check of the results.

4.1.1 BOUNDARY MAPS

Boundary maps highlight areas where connectivity patterns change abruptly. To generate them, the watershed-by-flooding algorithm is applied to the 40k gradient maps and then, the resulting boundaries are summed. The outcome is a boundary (or edge) map, with values representing the density of detected edges.

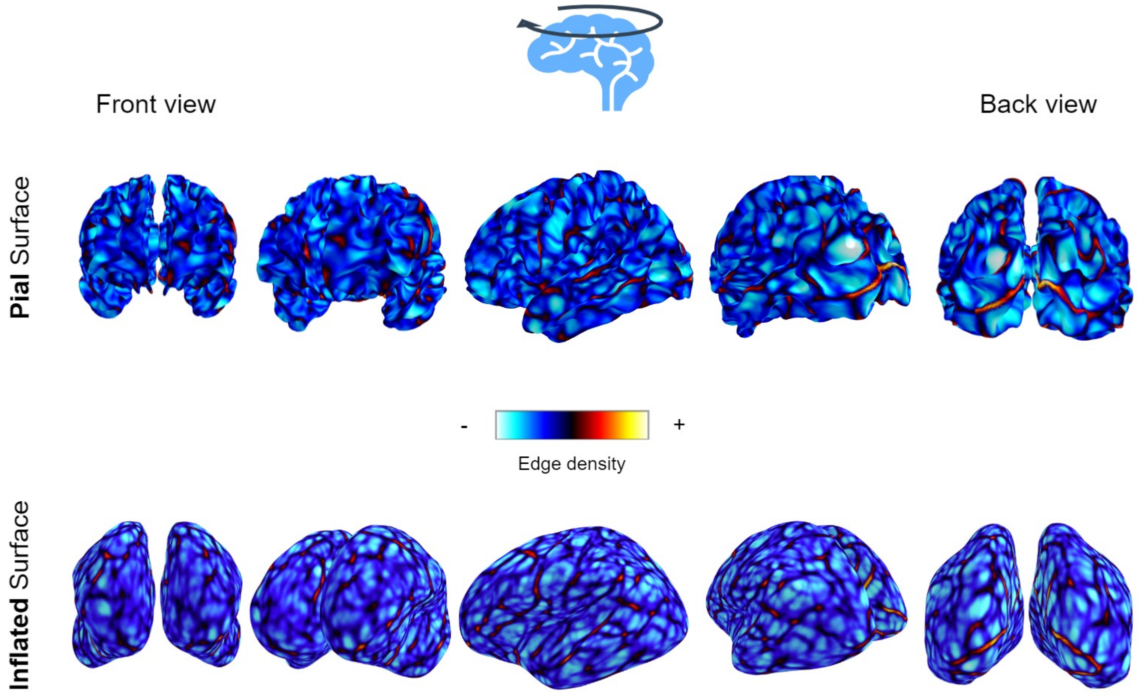
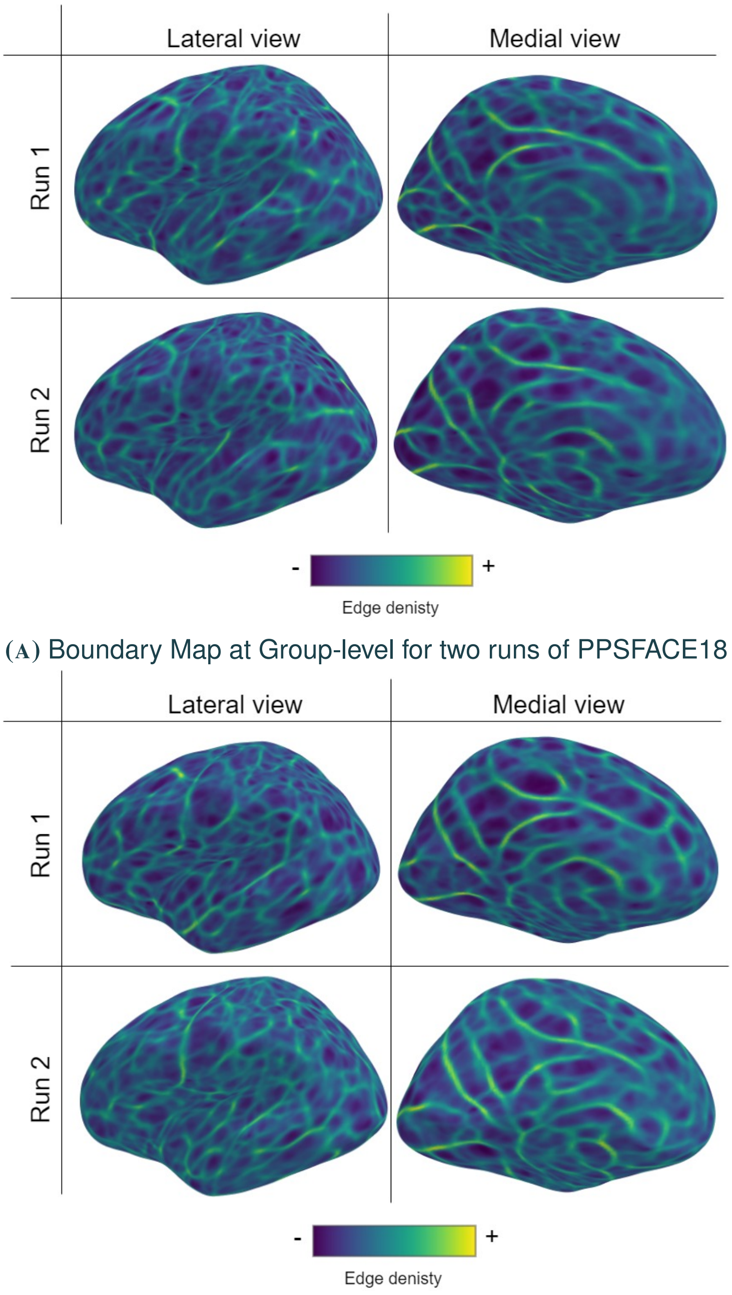


FIGURE 4.1
Whole-brain boundary map showing noticeable symmetry in boundary distribution between the left and right hemispheres (from dataset PPSFACE18). The top figure is the boundary map projected into the cortical pial and the lower figure is projected into an inflated surface. Inflated surface are often used for better visualization of the brain even though they don't represent an exact anatomical brain.

ACROSS RUNS AND DATASETS Here, I present the boundary maps obtained with my method 4.2. Two different datasets were processed, each with two runs, and some patterns appear consistent both across runs and across datasets.



(A) Boundary Map at Group-level for two runs of PPSFACE18

(B) Boundary Map at Group-level for two runs of PPSFACE20

FIGURE 4.2

4.1.2 PARCELLATION MAPS

To obtain the parcellation maps, the watershed algorithm is applied to the boundary map. Each local minimum serves as a seed, expanding based on the boundary map's topology.

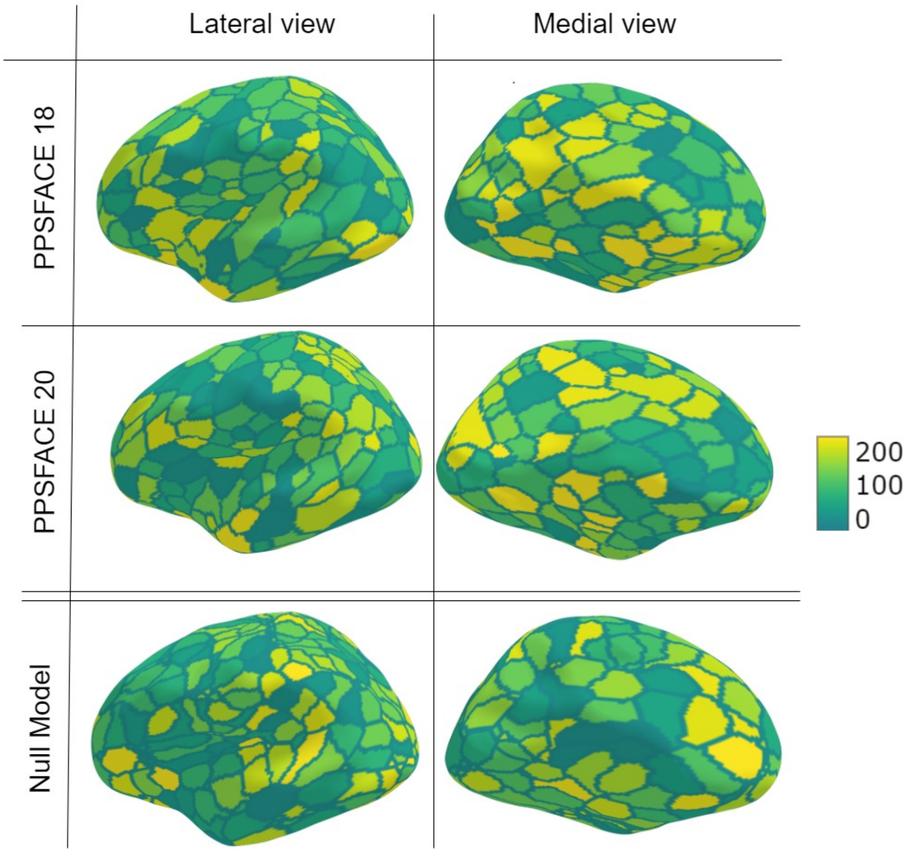


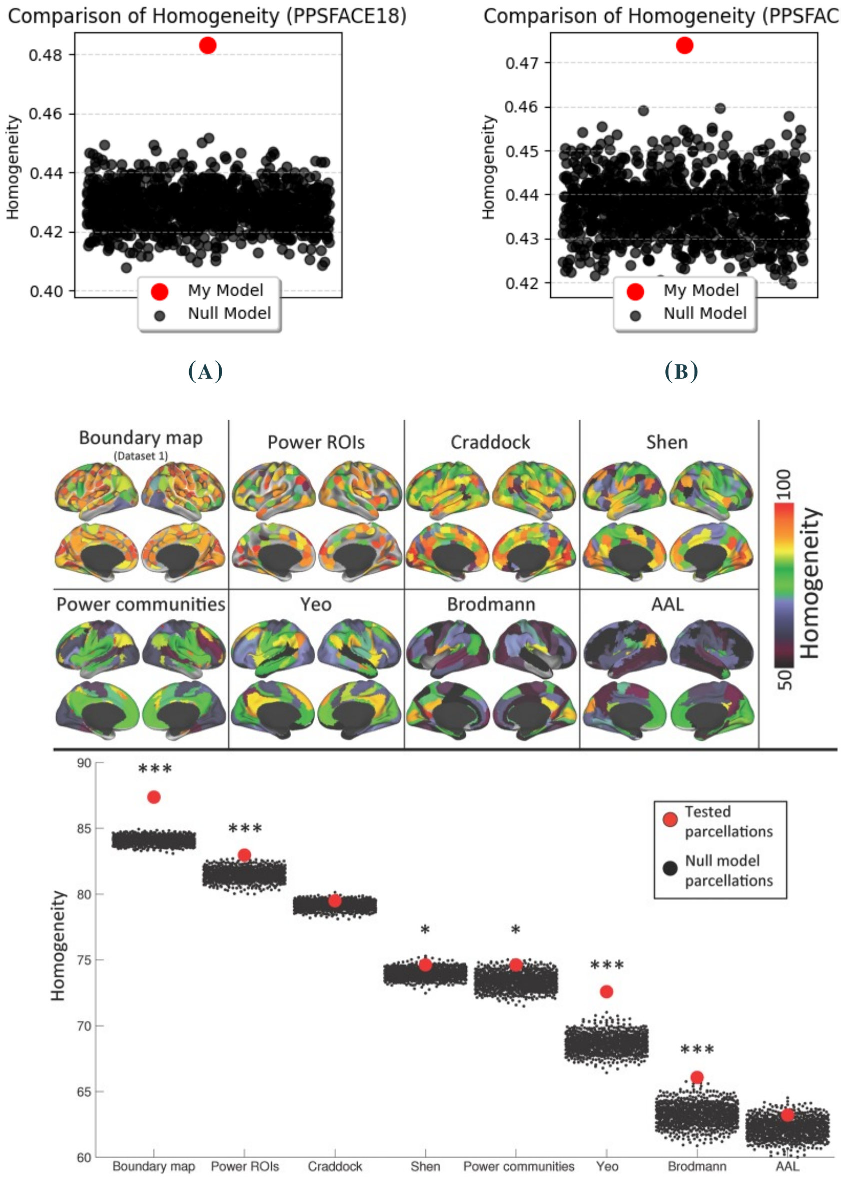
FIGURE 4.3
Number of parcels identified for each dataset. For PPSFACE18: 222 parcels in the left hemisphere and 221 parcels in the right hemisphere, totaling 443 parcels. For PPSFACE20: 238 parcels in the left hemisphere and 227 parcels in the right hemisphere, totaling 465 parcels. The null model represents the output of a random parcellation, resulting in n = 220 parcels.

4.2 PARCEL EVALUATION

In this section, specific metrics are employed to compare the proposed model with alternative approaches.

4.2.1 HOMOGENEITY

As described in Section 3.3.1, homogeneity is a measure commonly used in brain parcellation studies. It quantifies how internally consistent a given parcel is. Higher homogeneity values indicate that the connectivity patterns within each parcel are more uniform. Homogeneity values range from 0 to 1.



(A)

(B)

(C)

FIGURE 4.4

Comparison of parcellation homogeneity using my model versus a null model. (a) and (b) show the homogeneity results on PPSFACE18 and PPSFACE20, respectively, with a statistically significant p-value ($p < 0.001$). (c) presents a reference from Gordon et al. [5], where a different null model was used to evaluate homogeneity across major literature-based parcellations. Although their null model differs from mine, the key principle of maintaining the same number of parcels is consistent (*** $= p < 0.001$).

4.2.2 HOMOGENEITY, SUBJECT-LEVEL VS GROUP-LEVEL PARCELLATION

A key question in brain imaging is whether an individual (subject-level) parcellation gives better results than a group-level approach. In neuroscience, researchers often combine data to create group atlases representing an average brain. However, each brain is unique, and differences such as handedness (right/left handed), disabilities, or stroke history can lead to significant variability. Here, I check whether single-subject parcellations outperform group-level parcellations for that subject.

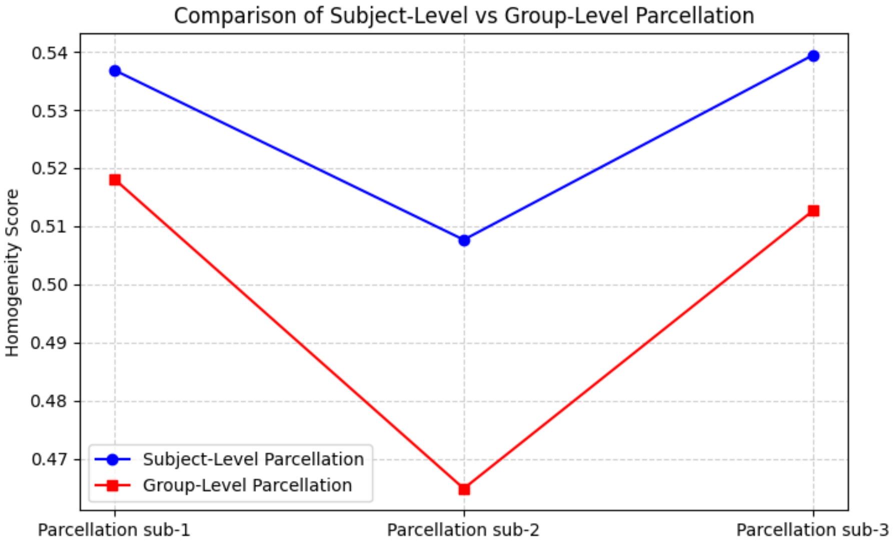
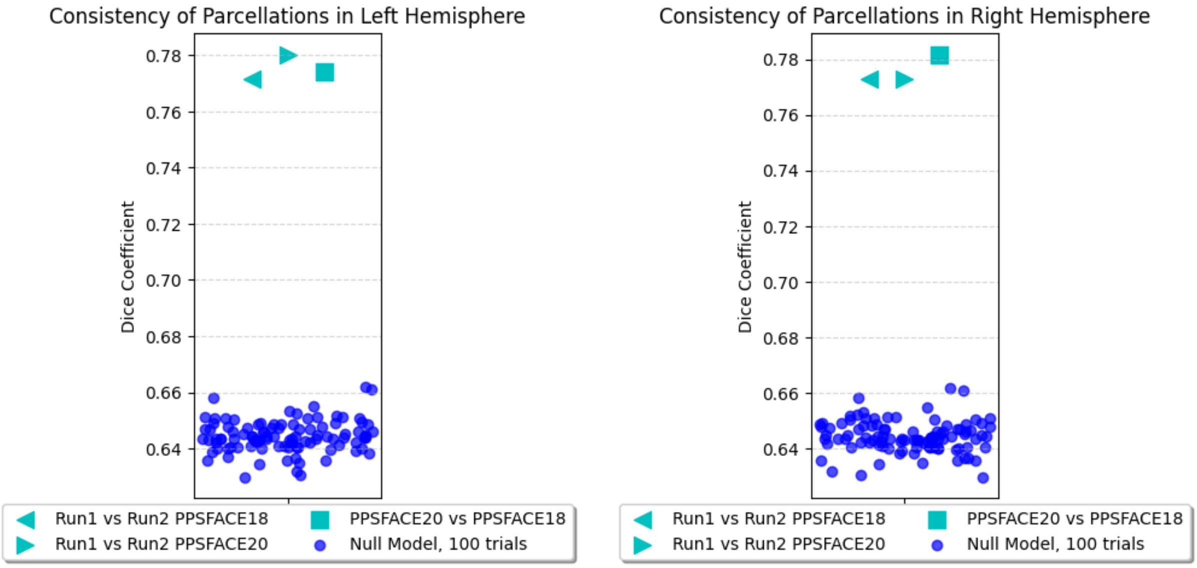


FIGURE 4.5
Subject-level parcellations from three participants were compared in terms of homogeneity. As expected, each participant's own parcellation exhibited higher homogeneity for that individual, though a larger sample size would be needed to confirm this finding.

.

4.2.3 DICE COEFFICIENT METRIC

he Dice coefficient quantifies the overlap between two parcellations (see Methods 3.3.2 for details). Here, I compare the Dice coefficient across different runs and datasets for both hemispheres.



(A) Dice coefficient on left hemisphere

(B) Dice coefficient on the right hemisphere

FIGURE 4.6
Each subject underwent two separate rs-fMRI scans (run 1 and run 2). Here, I compare the dice coefficients between the parcellations derived from these two runs and also the dice coefficient between the datasets. The null model compares two random parcellations. As expected, the model's parcellations show significantly better overlap than the null model (p-value <0.01). Interestingly, the overlap between runs is not significantly higher than the overlap between datasets, which is surprising.

4.2.4 GRAPH THEORY METRICS

In this part, the average time series of the parcels are correlated to obtain a $n_{\text{parcel}} \times n_{\text{parcel}}$ correlation matrix. The matrix is then thresholded to retain only the top 20% of correlations, thereby constructing an unweighted, undirected graph [17]. For comparison, the same metric was applied on a random graph (Erdős–Rényi [18]).

TABLE 4.1
Graph metrics for PPSFACE18, PPSFACE20, and Null Model (20% connections)

	Clustering Coefficient		Modularity		Shortest Path		Small-worldness	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
PPSFACE18	0.5535	0.02974	0.3154	0.03696	1.9831	0.06566	2.5347	0.06626
PPSFACE20	0.5631	0.03133	0.3293	0.02867	1.9592	0.09043	2.6186	0.22383
Erdős–Rényi	0.20057	0.00191	0.06754	0.00202	1.80024	0.00157	0.99893	0.00327

The results reveal that both PPSFACE18 and PPSFACE20 networks exhibit higher clustering coefficients ($\sim$ 0.55) than the random graph ($\sim$ 0.20), indicating that the brain networks form strong local clusters.

These findings are consistent with previous results reported in the literature [19]. In addition, the modularity values of the model networks ($\sim$ 0.315–0.329) confirm a clear community structure, suggesting that functionally similar regions tend to group together.

Although the average shortest path lengths in the real networks ($\sim$ 1.96–1.98) are slightly longer than in the random graph ($\sim$ 1.80) (which is expected because of increased clustering) they still support efficient global integration.

Consequently, the small-world indices of the models (2.53 for PPSFACE18 and 2.62 for PPSFACE20) are significantly higher than that of the random graph, underscoring a robust small-world organization.

4.3 TUNING OF THE MODEL

4.3.1 SIMILARITY MATRIX

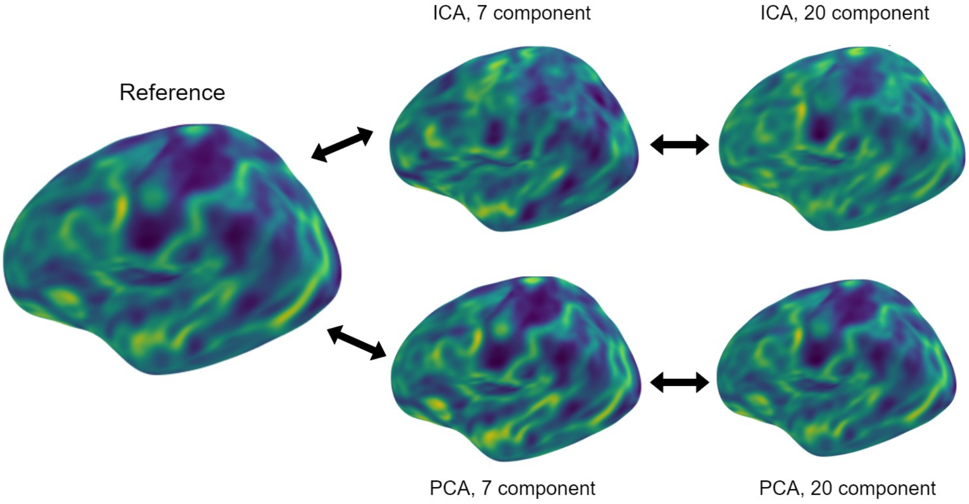


FIGURE 4.7

Due to the computational constraints, it was infeasible to compute the large correlation matrices used in [4, 5]. To manage these limitations, I conducted a visual inspection of the gradient using two different dimensionality reduction methods for the similarity matrix: Principal Component Analysis (PCA) and Independent Component Analysis (ICA). As a reference, a similarity matrix was constructed by randomly selecting 20,000 voxels, correlating them with each vertex, and then performing an autocorrelation to obtain the final similarity matrix. This process was computationally intensive and consumed nearly all of the available system memory. I subsequently computed the average gradient from this matrix, which provided an overview of the underlying connectivity structure.

4.3.2 ROLE OF SMOOTHING

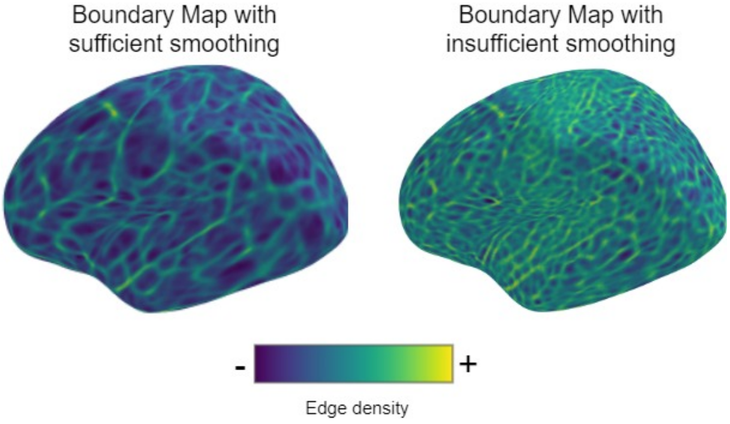

Boundary Map with sufficient smoothing
Boundary Map with insufficient smoothing
-
+
Edge density

FIGURE 4.8

Another key parameter in the parcellation process is the smoothing, which is performed on the similarity matrix and on the gradient maps. Higher smoothing values lead to larger parcels, and lower smoothing will give fine grained parcellation. In this study, I aimed to generate approximately ~ 450 parcels. Therefore an empirical trials was conducted to determine the appropriate smoothing setting that would meet this goal.

CHAPTER 5

DISCUSSION

In this work, I investigated the performance of a Python-based brain parcellation method using resting-state fMRI data, evaluated at both subject and group levels. The approach aim to delineate functional boundaries on cortical surfaces without removing smaller parcels, in contrast to the method proposed by Gordon and al. [5]. Overall, the findings highlight both the strengths and potential limitations of the parcellation pipeline.

A preliminary visual inspection showed symmetrical boundary mapping in the left and right hemispheres 4.1, suggesting that the parcellation approach successfully captures expected bilateral functional organization. Consistent boundary definitions is also observed across different datasets and across multiple runs 4.2a 4.2b, indicating that the method is relatively stable. Interestingly, despite these symmetries, the parcellation produced different numbers of parcels between hemispheres and between datasets 4.3. This discrepancy might be related to inter-individual variability in functional connectivity.

Quantitatively, the model demonstrated statistically superior homogeneity (see Figures 4.4a and 4.4b) and dice coefficient values (Figure 4.6) compared to a null model ($p < 0.001$). This results indicates that the detected parcels are not arbitrary, but rather exhibit distinct and consistent boundaries (an observation that is in line with findings from previous studies 4.4c, [5]). Such homogeneity underscores the quality of the parcellation, demonstrating its potential for characterization of brain functional organization.

Notably, parcellations generated at the single-subject level appeared to achieve higher homogeneity than those at the group level 4.5. This finding implies that individual connectivity patterns can be captured more effectively when the parcellation process does not average across multiple subjects. However, further subject-level parcellation analysis is necessary to confirm it.

One unexpected observation came from the dice coefficient analyses when comparing run 1 to run 2 and dataset 1 to dataset 2 4.6. I anticipated that the dice coefficient for repeated runs within the same dataset (run 1 vs run 2) would be higher than comparisons across datasets. However, the differences were minimal. This outcome may indicate a certain stability of the parcellation across datasets, further investigation with larger samples would be needed to confirm this.

Concerning the graph theory metrics (Table 4.1), both PPSFACE18 and PPSFACE20 networks exhibit higher clustering coefficients compared to the random graph, indicating that the brain networks form strong local clusters.The average shortest path in the model networks are slightly longer than those of the random graph, which is expected due to increased local clustering. However, these short paths still support efficient global integration. Consequently, the calculated small-worldness are higher than 1, suggesting a small-world organization.

A key advantage of the pipeline is its simplicity and accessibility. Aside from surface extraction via

FreeSurfer, the method relies on readily available Python libraries and user-friendly scripts, which I provide in a public GitHub repository 5.3. This accessibility should encourage broader adoption of individualized and group-level parcellation approaches and facilitate reproducibility in the future.

Overall, the Python-based brain parcellation method yielded promising results in terms of homogeneity and boundary consistency. Single-subject analyses highlight the potential benefits of personalized functional mapping. Future work may further optimize the pipeline, investigate additional graph-theoretical properties, and examine how the removal or preservation of small parcels influences the reliability of the resulting parcellations. Building on these findings, a more nuanced understanding of resting-state functional organization in both individuals and larger populations can be achieved.

5.1 LIMITATIONS

Despite the promising results and the simplicity of the Python-based pipeline, several limitations should be noted.

One of the primary constraints was limited computational power and time. With only local processing available, I had to make compromises regarding the number of run and the complexity of the algorithm to keep it close to established methods in the literature. This limitation restricted the size of data I could handle at once, as computing a correlation matrix for every vertex or voxel often exceeded the memory capacity.

The approach was greatly inspired by Gordon et al. [5], but I did not replicate their method exactly. The largest divergence lies in how the similarity matrix is computed. Instead of correlating all vertices and voxels (impractical given the memory constraints) I used a dimensionality reduction step. Through a visual analysis comparing PCA and ICA, I found that PCA with the first 17 components provided a performance profile most similar to Gordon's approach 4.7. While this choice simplified the computations, it may not perfectly capture all functional details.

The CONN toolbox was employed for data cleaning but omitted certain preprocessing steps recommended by other pipelines, like spatial and temporal scrubbing [5]. A more thorough preprocessing routine, potentially including advanced motion correction, physiological noise removal, may further improve both the quality and reliability of the parcellations.

Overall, these limitations highlight areas where future work could refine the pipeline, including more robust computational infrastructure, alternative dimensionality reduction approaches (like UMAP) and more elaborate preprocessing pipelines.

5.2 FUTURE DIRECTIONS AND APPLICATIONS

- **Comparing Healthy and Patient Groups:** A direct next step is to apply this parcellation method to healthy individuals versus patients with a specific condition. Analyzing their respective dice coefficients and homogeneity would help reveal how functional boundaries differ between the two groups.
- **Stroke Patient Parcellation:** As it can perform subject-level parcellation, it would also be valuable to perform it on stroke patients to determine whether areas of damage can be interpreted by their resting-state functional connectivity profiles.

5.3 SUPPLEMENTARY MATERIAL

- **Code Availability:** The complete code for the parcellation pipeline is available on GitHub at `https://github.com/RobinJunod/brain_parcellation.git`.

5.4 ACKNOWLEDGMENT

I would like to thank my supervisor, Michel Akselrod, for his guidance and support throughout this thesis. A special thanks to the MySpace Lab team for the interesting discussions and valuable collaboration.

BIBLIOGRAPHY

[1] Korbinian Brodmann. *Vergleichende Lokalisationslehre der Grosshirnrinde*. Leipzig: Johann Ambrosius Barth, 1909.

[2] Jonathan D. Power et al. 'Functional Network Organization of the Human Brain'. en. In: *Neuron* 72.4 (Nov. 2011), pp. 665–678. ISSN: 08966273. DOI: 10.1016/j.neuron.2011.09.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627311007926 (visited on 8th Jan. 2025).

[3] B. T. Thomas Yeo et al. 'The organization of the human cerebral cortex estimated by intrinsic functional connectivity'. en. In: *Journal of Neurophysiology* 106.3 (Sept. 2011), pp. 1125–1165. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.00338.2011. URL: https://www.physiology.org/doi/10.1152/jn.00338.2011 (visited on 8th Jan. 2025).

[4] Gagan S. Wig, Timothy O. Laumann and Steven E. Petersen. 'An approach for parcellating human cortical areas using resting-state correlations'. en. In: *NeuroImage* 93 (June 2014), pp. 276–291. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2013.07.035. URL: https://linkinghub.elsevier.com/retrieve/pii/S105381191300791X (visited on 8th Jan. 2025).

[5] Evan M. Gordon et al. 'Generation and Evaluation of a Cortical Area Parcellation from Resting-State Correlations'. en. In: *Cerebral Cortex* 26.1 (Jan. 2016), pp. 288–303. ISSN: 1047-3211, 1460-2199. DOI: 10.1093/cercor/bhu239. URL: https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhu239 (visited on 8th Jan. 2025).

[6] Alexander Schaefer et al. 'Local-Global Parcellation of the Human Cerebral Cortex from Intrinsic Functional Connectivity MRI'. en. In: *Cerebral Cortex* 28.9 (Sept. 2018), pp. 3095–3114. ISSN: 1047-3211, 1460-2199. DOI: 10.1093/cercor/bhx179. URL: https://academic.oup.com/cercor/article/28/9/3095/3978804 (visited on 8th Jan. 2025).

[7] Ru Kong et al. 'Spatial Topography of Individual-Specific Cortical Networks Predicts Human Cognition, Personality, and Emotion'. en. In: *Cerebral Cortex* 29.6 (June 2019), pp. 2533–2551. ISSN: 1047-3211, 1460-2199. DOI: 10.1093/cercor/bhy123. URL: https://academic.oup.com/cercor/article/29/6/2533/5033556 (visited on 8th Jan. 2025).

[8] Elena A. Allen et al. 'A Baseline for the Multivariate Comparison of Resting-State Networks'. en. In: *Frontiers in Systems Neuroscience* 5 (2011). ISSN: 1662-5137. DOI: 10.3389/fnsys.2011.00002. URL: http://journal.frontiersin.org/article/10.3389/fnsys.2011.00002/abstract (visited on 20th Feb. 2025).

[9] Koen V. Haak, Andre F. Marquand and Christian F. Beckmann. 'Connectopic mapping with resting-state fMRI'. en. In: *NeuroImage* 170 (Apr. 2018), pp. 83–94. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2017.06.075. URL: https://linkinghub.elsevier.com/retrieve/pii/S1053811917305463 (visited on 8th Jan. 2025).

[10] S. Trabanelli et al. 'Neural sensing of virtual infections activates innate immunity'. Under review in *Nature Neuroscience*. 2025.

[11] Bruce Fischl. 'FreeSurfer'. In: *NeuroImage* 62.2 (2012), pp. 774–781.

[12] K.J. Friston et al. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. SPM (RRID:SCR$_0$07037). London: Elsevier, 2007.

[13] S. Whitfield-Gabrieli and A. Nieto-Castanon. *CONN: A functional connectivity toolbox for correlated and anticorrelated brain networks*. 2012.

[14] Evan M. Gordon et al. 'Precision Functional Mapping of Individual Human Brains'. en. In: *Neuron* 95.4 (Aug. 2017), 791–807.e7. ISSN: 08966273. DOI: 10.1016/j.neuron.2017.07.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S089662731730613X (visited on 8th Jan. 2025).

[15] Ziteng Han. 'Investigating the heterogeneity within the somatosensory-motor network and its relationship with the attention and default systems'. en. In: ().

[16] R. Cameron Craddock et al. 'A whole brain fMRI atlas generated via spatially constrained spectral clustering'. en. In: *Human Brain Mapping* 33.8 (Aug. 2012), pp. 1914–1928. ISSN: 1065-9471, 1097-0193. DOI: 10.1002/hbm.21333. URL: https://onlinelibrary.wiley.com/doi/10.1002/hbm.21333 (visited on 24th Feb. 2025).

[17] Raymond Salvador et al. 'Neurophysiological Architecture of Functional Magnetic Resonance Images of Human Brain'. In: *Cerebral Cortex* 15.9 (Sept. 2005), pp. 1332–1342. DOI: 10.1093/cercor/bhi016.

[18] Paul Erdös and Alfréd Rényi. 'On the evolution of random graphs'. In: *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5 (1960), pp. 17–60.

[19] Sophie Achard et al. 'A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs'. In: *The Journal of Neuroscience* 26.1 (2006).

CHAPTER 6

ANNEX

* put the supplementatry data and images here

6.1 PSEUDO CODES

Algorithm 1 NullModel(G, n_{clusters})

```
$n \leftarrow$ number of vertices in $G$
Initialize array labels of size $n$ with all entries set to $-1$      ▷ All vertices unassigned
Randomly select $n_{\text{clusters}}$ distinct vertices as seeds
for $i \leftarrow 0$ to $n_{\text{clusters}} - 1$ do
    labels[seeds[i]] $\leftarrow i$      ▷ Assign a unique cluster ID to each seed
end for
Initialize a queue $Q_i$ for each seed, with $Q_i \leftarrow \{$seeds[$i$]$\}$
while at least one queue $Q_i$ is not empty do
    for each cluster $i$ such that $Q_i$ is not empty do
        $current \leftarrow$ Dequeue($Q_i$)
        for each $neighbor \in G[current]$ do
            if labels[neighbor] $= -1$ then
                labels[neighbor] $\leftarrow i$
                Enqueue($neighbor$, $Q_i$)
            else if labels[neighbor] $\geq 0$ and labels[neighbor] $\neq i$ then
                labels[current] $\leftarrow -2$      ▷ Mark as boundary
                labels[neighbor] $\leftarrow -2$      ▷ Mark as boundary
            end if
        end for
    end for
end while
return labels
```

Algorithm 2 WatershedByFlooding(G, $values$)

```
$n \leftarrow$ **number of nodes in** $G$    ▷ Nodes indexed $0, 1, \ldots, n-1$
$minima\_indices \leftarrow$ FINDLOCALMINIMA$(values, G)$
Initialize array $labels[0 \ldots n-1] \leftarrow -1$    ▷ All vertices unassigned
**for** each $i \in minima\_indices$ with basin ID $b$ **do**
    $labels[i] \leftarrow b$
**end for**
Initialize empty priority queue $pq$
**for** each $i \in minima\_indices$ **do**
    **Push** $(values[i], i)$ into $pq$
**end for**
**while** $pq$ is not empty **do**
    $(current\_value, current\_vertex) \leftarrow$ **Pop** minimum element from $pq$
    $current\_label \leftarrow labels[current\_vertex]$
    **if** $current\_label < 0$ **or** $current\_label = -2$ **then**
        **continue**    ▷ Skip unassigned or boundary vertices
    **end if**
    **for** each neighbor $nbr \in$ NEIGHBORS$(current\_vertex, G)$ **do**
        $nbr\_label \leftarrow labels[nbr]$
        **if** $nbr\_label = -1$ **then**
            $labels[nbr] \leftarrow current\_label$
            **Push** $(values[nbr], nbr)$ into $pq$
        **else if** $nbr\_label \geq 0$ **and** $nbr\_label \neq current\_label$ **then**
            $labels[current\_vertex] \leftarrow -2$    ▷ Mark current vertex as boundary
            $labels[nbr] \leftarrow -2$    ▷ Mark neighbor as boundary
            **break**    ▷ Stop processing current vertex
        **end if**
    **end for**
**end while**
**return** $labels$
```

Algorithm 3 SmoothSurfaceGraph(G, $values$, $iterations$)

```
$n \leftarrow$ number of vertices in $G$
if $G$.number_of_nodes() $\neq n$ or $values$ is not 1D or 2D then
    error: "Invalid input dimensions."
    return
end if                                                    ▷ Build the adjacency list for each vertex
for $u \leftarrow 0$ to $n - 1$ do
    $adjacency\_list[u] \leftarrow \{\, v \mid (u, v) \in E(G)\}$
end for
$smoothed\_map \leftarrow values$                                                      ▷ Current map
$new\_map \leftarrow$ array of zeros with the same shape as $values$
for $iter \leftarrow 1$ to $iterations$ do
    for $u \leftarrow 0$ to $n - 1$ do
        $neighbors \leftarrow adjacency\_list[u]$
        if $values$ is 1D then
            $total \leftarrow smoothed\_map[u]$
            $count \leftarrow 1$
            for all $v \in neighbors$ do
                $total \leftarrow total + smoothed\_map[v]$
                $count \leftarrow count + 1$
            end for
            $new\_map[u] \leftarrow total/count$
        else
            $total \leftarrow smoothed\_map[u, :]$                                ▷ Vector for vertex $u$
            $count \leftarrow 1$
            if $neighbors \neq \varnothing$ then
                $total \leftarrow total + \sum_{v \in neighbors} smoothed\_map[v, :]$
                $count \leftarrow count + |\text{neighbors}|$
            end if
            $new\_map[u, :] \leftarrow total/count$
        end if
    end for
    Swap($smoothed\_map$, $new\_map$)
end for
return $smoothed\_map$
```

Algorithm 4 ComputeGradients(G, S)

```
Convert $S$ to a NumPy array with shape $(N, M)$ of type float16
$(N, M) \leftarrow$ shape of $S$
Initialize $squared\_sums \leftarrow$ zero array of shape $(N, M)$, type float16
for each edge $(v, u)$ in $G$.edges() do
    $diff \leftarrow S[v, :] - S[u, :]$
    $diff\_sq \leftarrow diff \cdot diff$                                     ▷ Elementwise square
    $squared\_sums[v, :] \leftarrow squared\_sums[v, :] + diff\_sq$
    $squared\_sums[u, :] \leftarrow squared\_sums[u, :] + diff\_sq$
end for
$gradients \leftarrow \sqrt{squared\_sums}$
return $gradients$
```