Segmenting the brain via sparse inverse covariance estimation and graph-based clustering on high-dimensional fMRI data

Alnur Ali

Wednesday 29th November, 2017

Abstract

Background. An important problem in neuroscience is understanding what parts of the human brain work together. Neuroscientists often approach this problem by generating — largely by hand — clusterings of points in and on the brain. From the point of view of statistics, the relevant tools here are: (1) sparse inverse covariance estimators, which yield partial correlation graphs, and (2) graph-based clustering methods, for clustering the graphs.

Aim. We apply a state-of-the-art estimator of the inverse covariance matrix, called PseudoNet, as well as several graph-based clustering methods (including the well-known Louvain method), to recently released functional magnetic resonance imaging (fMRI) data, in order to generate an entirely data-driven clustering of the brain that reveals insights about how the brain works.

Data. As far as fMRI data go, the data we use is relatively new and good quality; the data is from the Human Connectome Project, a research project started by the government to study what parts of the brain work together. Specifically, the data we use is an $(91,282 \times 91,282)$-dimensional sample correlation matrix that is of size ≈ 30 GB.

Method. We take a two-step approach. First, we generate a partial correlation graph by running the sample correlation matrix through PseudoNet. Then, we use a graph-clustering method to generate clusters from the graph. We compare how close our clustering is to a state-of-the-art, hand-crafted clustering from the neuroscience literature by using a variant of the standard Jaccard score.

Results. Although our method is not perfect, it does capture some important features of the brain known to neuroscientists.

Conclusion. The results we get are encouraging, suggesting that with a little more work, we might be able to generate a clustering that is completely on par with (or better than), in some sense, the hand-crafted one from the neuroscience literature.

Intellectual merit. As far as we can tell, there really does not seem to be a single paper in either the neuroscience or statistics literatures that uses the latest in fMRI data as well as statistical methodology to generate clusterings of the brain. This is probably because (1) fMRI data tends to be very high-dimensional, ruling out many standard tools in a computational scientist's toolbox; (2) neuroscientists may simply, by and large, be unaware of the latest tools in statistics; and (3) good quality fMRI data has only recently become widely available.

Broader impacts. Understanding which parts of the brain work together can help suggest avenues for future research in neuroscience and medicine. Our method can also be applied to other domains, *e.g.*, genetics, finance, and energy, and might be useful there.

Keywords: sparse inverse covariance estimation; community detection; fMRI; brain.

1 Introduction

An important, long-standing question in systems neuroscience is (colloquially speaking): which parts of the brain work together? Having an answer to this question is not only intellectually interesting, but also practically useful because it can help suggest avenues for further research in neuroscience and medicine. Of course, the question is somewhat ill-defined, so we must make precise what we mean by the phrases "parts" and "work together". For "parts", we simply mean a set $V \subset \mathbf{R}^3$ of points/voxels that lie in and on the brain (we make this still more precise later, in Section 5). Turning to "work together", we leave this phrase a little less defined, only taking it to mean: two (or more) parts of the brain "work together" if they are involved in accomplishing the same basic neuroscientific process, while a person is at rest. So, our question amounts to essentially asking for a partition $V_1, \ldots, V_k$ of the set of voxels V, possessing some neuroscientific validity. In the neuroscience literature, these kinds of partitions are usually referred to as *functional connectivity maps*, *functional connectomes*, or *parcellations*, while in statistics the term *clustering* is more common; in this report, we use all these terms interchangeably.

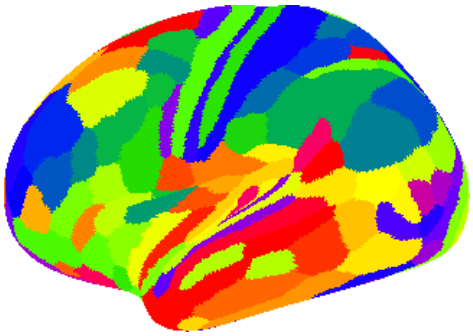


Figure 1: The Brodmann areas, for the left hemisphere of the brain.

Obtaining a reasonable functional connectivity map has been a topic of interest in the neuroscience community since at least the early 1900s, starting with Brodmann's work on the so-called Brodmann areas, one of the first parcellations of the brain, which still has some use even today; see Figure 1 to the right, for a lateral (*i.e.*, side) picture of the Brodmann areas on the left hemisphere of the brain. Recent work has focused on building parcellations, largely by hand and/or by using simple computational methods, starting from high-quality functional magnetic resonance imaging (fMRI)[1] data; see, *e.g.*, Eickhoff et al. (2015) for a good survey, as well as Glasser et al. (2016) for a highly regarded approach here. On the other hand, a smaller stream of recent work has looked at applying more sophisticated statistical methods, but using older low-dimensional fMRI data (Ryali et al., 2012; Hsieh et al., 2013; Fu et al., 2015; Belilovsky et al., 2016; Wang et al., 2016). Notably, what seems to be missing from the literature is a mostly/entirely data-driven, non-invasive (*i.e.*, non-dissective) approach for generating a parcellation that leverages the latest in both fMRI data and statistical methodology.

From a statistical point of view, an initial attempt at generating a parcellation is to simply apply an off-the-shelf clustering algorithm to fMRI data; empirically, this approach fails to generate use-

[1] Recall that fMRI measures oxygen levels throughout the brain as a function of time, providing a dynamic picture of the brain; by contrast, standard MRI does not measure oxygen, and only provides a static snapshot of the brain.

ful results, at least in part because it is not entirely clear what metric and underlying probability model are appropriate here. A more refined two-step approach is, at a high-level, as follows: (1) generate an (unweighted or weighted) undirected graph, based on the (partial) correlations between the voxels detected in the fMRI data, and then (2) apply a graph-based clustering algorithm to generate a parcellation, based on the strength of the correlations. Methods for sparse inverse covariance estimation (see, *e.g.*, Friedman et al. (2008); Peng et al. (2009); Ali et al. (2016a)) are quite useful tools for the first step here, as they have been used to successfully generate conditional independence, marginal correlation, and partial correlation graphs (Wasserman, 2016) from data in many other domains, *e.g.*, genetics (Friedman et al., 2008; Khare et al., 2015), finance (Won et al., 2013; Khare et al., 2015; Ali et al., 2016a), and energy (Wytock and Kolter, 2013; Ali et al., 2016b,a).

In this report, we present an entirely data-driven method for generating a parcellation from recently released high-dimensional fMRI data, by leveraging very recent advances in the sparse inverse covariance estimation and numerical linear algebra literatures; qualitatively, the method is able to generate parcellations that match, in several important ways, a state-of-the-art parcellation generated largely by hand (Glasser et al., 2016). This is useful because it reduces the need for laborious and potentially error-prone engineering by hand when making scientific discoveries; furthermore, our method is fast, and also comes with a number of other computational and statistical guarantees, which essentially assure us that the method is "doing the right thing" (under certain conditions).

An outline for the rest of this report is as follows. In Section 2, we review some related work, and explain in a little more detail some of the concepts we touched on above. In Section 3, we succinctly state our goals for this report. In Section 4, we describe our approach for generating a parcellation, in some detail. In Section 5, we give the details on the fMRI data we use. In Section 6, we present the results of applying our method for generating a parcellation to the fMRI data; in Section 6, we also analyze the results. Finally, in Section 7, we wrap up with a brief look back at our findings.

2 Background and related work

We divide our review of relevant work into two parts: first we cover work coming out of the neuroscience literature, and then we cover work coming out of statistics.

As mentioned above, the study of functional connectivity maps was initiated, more or less, with Brodmann's seminal work in the early 1900s; of course, Brodmann's maps were generated without the benefit of the (non-invasive) fMRI data that we have access to these days. Another example of a functional connectivity map, from the early days of research in this area, can be seen in Table 3 of Felleman and Van Essen (1991), which is essentially just an adjacency matrix, where the vertices (*i.e.*, rows/columns) are different regions of the cerebral cortex (*i.e.*, the surface of the brain). More recently, with the increasing availability of fMRI data, researchers have turned to generating parcellations from data; the state-of-the-art here is really the work of Glasser et al. (2016), which generates a parcellation by applying a multi-class, shallow neural network to high-quality fMRI data recently released by the Human Connectome Project (much more on this data later, in Sec-

tion 5), but then relies on domain knowledge in order to significantly post-processes the results by hand. Although there is no universally agreed upon ground truth for parcellations of the brain, the parcellation of Glasser et al. (2016) is highly regarded in parts of the neuroscience community for at least a few reasons: (1) the authors used the latest available (and hence least noisy) fMRI data, in addition to other data sources (*e.g.*, the distribution of fat across the brain) and domain knowledge (we use neither of these); (2) many of the clusters corroborate the independent findings of neuroscientists studying specific parts of the brain; and (3) the clusters matched fMRI scans of test subjects performing certain tasks.

Turning to the statistics literature, the work of Ryali et al. (2012) applies the SPACE estimator of the inverse covariance matrix (Peng et al., 2009), except augmented with an additional squared ℓ_2-norm penalty, to low-dimensional fMRI data.[2] This approach turns out to be quite related to the approach we pursue in this report, but it nonetheless suffers from some serious limitations that our approach does not: (1) the criterion in defining optimization problem for the estimator in question is nonconvex, meaning that the estimates generated by the method may not be globally optimal, leading to interpretability issues and also meaning that it might be quite computationally expensive to compute any estimate at all; (2) the estimates are not guaranteed to recover the correct sparsity pattern under idealized conditions (see Corollary 4.5 in Ali et al. (2016a)), meaning that the method might actually generate partial correlation graphs with missing edges; and (3) the work uses older, lower-quality fMRI data: the spatial and temporal resolution, number of subjects, and quality of the fMRI machines used are all lower than that of the more recent data made available by the Human Connectome Project.

In terms of other statistical work, Hsieh et al. (2013) apply a method that they call BIG & QUIC to some very high-dimensional fMRI data, in order to generate a conditional independence graph; this is mostly done as a proof-of-concept of the method, as the qualitative analysis the authors present is somewhat cursory. Fu et al. (2015) apply the graphical lasso algorithm of Friedman et al. (2008) to older fMRI data, except with an ℓ_0-"norm" penalty instead of the usual ℓ_1-norm penalty, making the corresponding optimization problem nonconvex. Finally, Belilovsky et al. (2016) appy a pseudolikelihood-based estimator, with a deep neural network as the predictive primitive instead of, say, a lasso regression, to fMRI data obtained from subjects with autism instead of the broader population.

We mention again that, as far as we are aware of, there is really not a single piece of work in either the neuroscience or statistical literatures that takes advantage of the very latest advances in both fMRI data and statistical methodology in order to generate a useful parcellation.

[2] An interesting side note, for statisticians: the authors use stability selection (Meinshausen and Bühlmann, 2010) to choose the tuning parameters for their method.

3 Aim / problem statement

Here, we state more plainly the problem we are trying to solve in this report, deferring most of the details to subsequent sections. As before, let $V \subset \mathbf{R}^3$ be a set of $p = 91,282$ points/voxels lying in and on the brain. Let $X \in \mathbf{R}^{n \times p}$ be a data matrix, constructed by observing the values of the p voxels across $1,200$ people, at different points in time, in an fMRI machine, and then averaging over the people and points in time so that $n = 5,142$ (and $p = 91,282$, still). Now compute the sample correlation matrix S, by standardizing the columns of X and then forming $S = (1/n)X^T X$. We want to use S to generate a clustering of the p voxels, that at least to some extent qualitatively and quantitatively matches the results from Glasser et al. (2016), a highly regarded parcellation from the neuroscience literature. Given the large value of p, we would also like the method we use for generating the parcellation to be computationally efficient, and possess favorable statistical properties.

4 Method

To help orient the reader, we first present an overview of our method for generating a parcellation; each subsection then steps through the details.

4.1 Overview of method

Given a sample correlation matrix $S \in \mathbf{S}_+^p$, the set of $(p \times p)$-dimensional positive semidefinite matrices, generated from the raw fMRI data, ...

1. ... we first generate a partial correlation graph[3] by passing the sample correlation matrix S as the input to PseudoNet, a new method for for sparse inverse covariance estimation (see Section 3.1 of Khare et al. (2015) as well as Ali et al. (2016a) for details). PseudoNet returns an estimate $\hat{\Omega}$ of the underlying inverse covariance (correlation) matrix; the sparsity pattern of $\hat{\Omega}$ yields an adjacency matrix for the partial correlation graph.

2. Next, we use an off-the-shelf graph-based clustering method (actually, we experiment with three different methods) that generates a clustering $\mathcal{C}(\hat{\Omega})$ from the partial correlation graph.

3. Finally, we measure the quality of a parcellation both qualitatively and quantitatively. Qualitatively, by visually inspecting the clustering $\mathcal{C}(\hat{\Omega})$, and comparing it to what is known about the brain from the systems neuroscience literature. Quantitatively, by computing a variant of the Jaccard score, which is a classic measure that evaluates how similar two clusterings are, relative to the parcellation generated by Glasser et al. (2016).

[3] Recall that a partial correlation graph is an undirected graph, where the vertices represent the variables (voxels, for us) and an edge is placed between two vertices if and only if the partial correlation between the corresponding variables is nonzero; when the underlying data-generating process is multivariate normal, then the partial correlation graph exactly coincides with the (perhaps more familiar) conditional independence graph.

4.2 Sparse inverse covariance estimation

Here, we describe PseudoNet in more detail. PseudoNet generates an estimate of the underlying inverse covariance matrix, by solving the following convex optimization problem:

$$\hat{\Omega}(\lambda_1, \lambda_2) = \underset{\Omega \in \mathbf{R}^{p \times p}}{\operatorname{argmin}} \quad \left\{ -\tfrac{1}{2} \log\det(\Omega^2_{\text{diag}}) + \tfrac{n}{2} \operatorname{\mathbf{Tr}} S\Omega^2 + \lambda_1 \|\Omega_{\text{off}}\|_1 + \tfrac{\lambda_2}{2} \|\Omega\|_F^2 \right\}, \tag{1}$$

where $\Omega_{\text{diag}} \in \mathbf{R}^{p \times p}$ is a diagonal matrix whose entries are the diagonal entries of Ω; $\Omega_{\text{off}} \in \mathbf{R}^{p \times p}$ is a matrix whose off-diagonal entries are the off-diagonal entries of Ω, but its diagonal entries are set to zero; $S \in \mathbf{R}^{p \times p}$ is again the sample correlation matrix; $\|\cdot\|_1$ is the elementwise ℓ_1-norm; $\|\cdot\|_F$ is the Frobenius norm; and $\lambda_1, \lambda_2 > 0$ are tuning parameters.

Why PseudoNet? Compared to other estimators of the inverse covariance matrix, PseudoNet possesses a number of useful properties that other estimators do not (see Ali et al. (2016a) for more details): briefly, computing the PseudoNet estimate is (1) computationally fast and scalable, since PseudoNet distributes some of its core numerical linear algebra work across a network of machines (see the next paragraph for more details); (2) capable of handling non-Gaussian data, empirically speaking; and (3) guaranteed to be unique, globally optimal, and recover the sparsity pattern of the underlying inverse covariance matrix.

Optimization algorithm. PseudoNet uses a proximal gradient method to optimize the criterion in (1); the important parts of the method are described in Algorithm 1. Assuming the iterates $\hat{\Omega}^{(1)}, \hat{\Omega}^{(2)}, \ldots$ are sparse (but S is dense), it turns out that the runtime of a single iteration of Algorithm 1 is dominated by the cost of computing the soft-thresholding operator, $O(p^2)$, and the cost of the sparse-dense matrix-matrix multiplication[4] $S\Omega$, $O(\mathbf{nnz}(\Omega) \cdot p)$, where **nnz** counts the number of nonzero entries in its argument. For large values of p, as is the case for us, the method can be slow and require a lot of space; a distributed approach is therefore useful here. For all the experiments in this report, we use the distributed SpDM^3 method of Koanantakool et al. (2016) for computing the product $S\Omega$ (as well as the product ΩS, by symmetry); the code for this version of PseudoNet is available at: `https://bitbucket.org/penpornk/spdm3-hpconcord`.

Choice of tuning parameters. We computed the PseudoNet estimate at all combinations of the tuning parameters

$$\lambda_1 \in \{0.5, 0.6132, 0.6388, 0.6931, 0.722\} \times \lambda_2 \in \{0.1024, 0.128, 0.16, 0.2, 0.25, 0.3125\}.$$

The above grid of tuning parameter values was chosen based on trial and error: making λ_1 or λ_2 much bigger resulted in the identity matrix as the estimate, and making λ_1, λ_2 any smaller resulted in an extremely dense estimate. For a single (λ_1, λ_2) pair on the grid, the proximal gradient method took ≈ 37 minutes to run on an `m4.4xlarge` Amazon EC2 instance[5], with MPI. We also

[4] The phrase "sparse-dense matrix-matrix multiplication" is often written as "SpDM^3", for short.

[5] The `m4.4xlarge` instance has 16 logical cores and 32 GB of RAM.

Algorithm 1 Sketch of the proximal gradient method for computing the PseudoNet estimate

Input: sample correlation matrix $S \in \mathbf{S}_+^p$; tuning parameters $\lambda_1, \lambda_2 > 0$
Output: estimate of the inverse covariance matrix $\hat{\Omega}$
initialize starting point $\Omega \in \mathbf{S}_{++}^p$ (the set of $(p \times p)$-dimensional positive definite matrices); optimization tolerance $\epsilon > 0$; line search parameters $\tau_{\text{init}}, \beta \in (0, 1)$
repeat

- compute the gradient, $g(\Omega) = -\Omega_{\text{diag}}^{-1} + \frac{n}{2}(S\Omega + \Omega S) + \lambda_2 \Omega$, of the smooth part of the criterion in (1) evaluated at Ω; compute $S\Omega$ (and ΩS) in a distributed fashion, using the SpDM^3 method of Koanantakool et al. (2016)
- choose the step size τ via backtracking line search, using the parameters $\tau_{\text{init}}, \beta$
- update $\Omega \leftarrow \mathbf{prox}_{(\lambda_1 \tau)\|\cdot\|_1}(\Omega - \tau g(\Omega))$, where $\mathbf{prox}_{\alpha\|\cdot\|_1}$ is the *proximal operator* of $z \mapsto \|z\|_1$ with parameter $\alpha > 0$, *i.e.*, the *soft-thresholding operator* (Parikh and Boyd, 2013)

until some stopping criterion is satisfied
output the estimate $\hat{\Omega} \leftarrow \Omega$

used warm-starting, from tuning parameter-to-tuning parameter, to further accelerate the method. Later, in Section 6, we present the results for all the estimates, and then compare them qualitatively as well as quantitatively.

We briefly mention two other potential directions here. First, as Section S.1.2 of the supplementary material for Ali et al. (2016a) suggests, we could have used, say, cross-validation to pick a single (λ_1, λ_2) pair to report that optimizes a quantity resembling the Bayesian Information Criterion; however, for this report, we are most interested in seeing visually how the parcellation changes with the tuning parameters. Second, as Section S.1.3 suggests, we could have used sequential strong screening rules (Tibshirani et al., 2012), from tuning parameter-to-tuning parameter, to accelerate the proximal gradient method; however, these rules generate solutions that are not guaranteed to satisfy the Karush-Kuhn-Tucker conditions (although, empirically, the rules seem to work very well).

Where does the criterion in (1) **come from?** The criterion in (1) may look a little mysterious. In the supplement, we motivate it by showing that it follows quite naturally from basic arguments about partial correlations; the criterion can also be seen as a generalization of the criterion for the graphical lasso.

Sample correlation matrices vs. sample covariance matrices. Even though PseudoNet was originally desgined to accept a sample *covariance* matrix as input (see Section 2 in Ali et al. (2016a)), passing in a sample *correlation* matrix does not really change anything, for our purposes; this is important for us because the sample covariance matrix is not readily available to us (but the sample correlation matrix is). The reason for nothing really changing is because using the sample correlation matrix is equivalent to just working with the standardized variables $X_1, \ldots, X_p$; assuming standardized variables, the sample correlation matrix actually is the sample covariance

matrix for the transformed variables. Furthermore, let $\hat{\Omega}$ be the PseudoNet estimate for the transformed variables. Then $\hat{\Omega}^{-1}$ is an estimate of the underlying covariance matrix for the transformed variables, in which case

$$\hat{D}^{-1}\hat{\Omega}^{-1}\hat{D}^{-1}$$

is an estimate of the underlying covariance matrix for the untransformed variables, where $\hat{D}$ is a diagonal matrix whose entries are the sample standard deviations. So,

$$(\hat{D}^{-1}\hat{\Omega}^{-1}\hat{D}^{-1})^{-1} = \hat{D}\hat{\Omega}\hat{D}$$

is an estimate of the underlying inverse covariance matrix for the untransformed (but still zero mean) variables, because $\hat{D}$ is diagonal with strictly positive entries and $\hat{\Omega}$ is positive definite. Finally, we get that

$$\text{the sparsity pattern of } \hat{\Omega} = \text{the sparsity pattern of } \hat{D}\hat{\Omega}\hat{D},$$

since again $\hat{D}$ is diagonal with strictly positive entries; thus we can still generate a partial correlation graph for the untransformed variables, even by working with the sample correlation matrix.

4.3 Graph-based clustering

Having computed a partial correlation graph from the PseudoNet estimate $\hat{\Omega}$, we consider three different methods for computing a clustering from the graph.

1. *Connected components.* We use a standard algorithm (Pearce, 2005) for identifying the connected components in the partial correlation graph. Unfortunately, for all the PseudoNet estimates we compute in previous subsection, the algorithm identifies only a single connected component, making this approach not too useful; consequently, we do not report any results for this approach.

2. *Louvain method.* We use a standard algorithm for community detection, called the Louvain method (Blondel et al., 2008). The Louvain method generates a range of clusterings, going from ones with many clusters (*i.e.*, finer clusterings) to ones with a few (*i.e.*, coarser). Empirically, the Louvain method has been observed to run in $O(p \log p)$ time, which is quite fast, making it a much better choice than other graph-based clustering algorithms, *e.g.*, spectral clustering, which requires an eigenvalue decomposition.

3. *Persistent homology.* We use an off-the-shelf Python package from the persistent homology literature (Morozov, 2016), for computing a clustering from the degree matrix for the graph generated by PseudoNet.

4.4 Measuring the quality of a clustering

To evaluate a clustering $\mathcal{C}(\hat{\Omega})$ generated by our workflow, we compare the clustering to the clustering $\mathcal{C}_{\text{Glas}}$ given in Glasser et al. (2016), using a variation of the standard Jaccard score: we compute

$$\textbf{Jacc}(\mathcal{C}(\hat{\Omega}), \mathcal{C}_{\text{Glas}}) = \frac{1}{2}\left(J(\mathcal{C}(\hat{\Omega}), \mathcal{C}_{\text{Glas}}) + J(\mathcal{C}_{\text{Glas}}, \mathcal{C}(\hat{\Omega}))\right), \tag{2}$$

where

$$J(\mathcal{A}, \mathcal{B}) = \sum_{i=1}^{k} \max_{j=1,\ldots,\ell} \frac{|A_i \cap B_j|}{|A_i \cup B_j|}$$

for two clusterings $\mathcal{A} = \{A_1, \ldots, A_k\}$ and $\mathcal{B} = \{B_1, \ldots, B_\ell\}$.

Trying other metrics, *e.g.*, the *variation of information* introduced by Meilă (2003), for measuring the similarity between two clusterings is certainly worthwhile, as different metrics have different strengths/weaknesses; we leave this kind of investigation to future work. The Jaccard score in particular is useful, at least in part because it is easy to compute and explain.

5 Data

As far as fMRI data goes, the data we use is relatively newer and of higher quality. The data was generated by the Human Connectome Project (Smith et al., 2013), a research project initiated in 2011 by two National Institutes of Health (NIH) grants given to a group of 10 universities; the goal of the project is to understand how the various parts of the cerebral cortex work together.[6]

The data that we use was generated as follows. First, 1,200 subjects were put into a state-of-the-art fMRI machine (see the left panel of Figure 2) and measurements were taken, while the subjects were at rest (*i.e.*, without stimulating the subjects), every 0.7 seconds for an hour, at 2 millimeter $\times$ 2 millimeter $\times$ 2 millimeter cubes/voxels spread evenly throughout the cerebral cortex. Next, as fMRI data is typically very noisy, a significant amount of post-processing was done to denoise the data, ultimately leading to a data matrix with dimensions $\approx 6,171,400 \times 91,282$. To further reduce the level of noise, the columns of the data matrix were then averaged over the 1,200 subjects, leading to a data matrix with dimensions $\approx 5,142 \times 91,282$. Finally, a sample correlation matrix with dimensions $91,282 \times 91,282$ (and of size ≈ 30 GB) was computed, which is used as input for PseudoNet. Figure 2 of Smith et al. (2013) provides a nice overview of the data generation process; for convenience, we present the figure here, in the right panel of Figure 2.

Table 1 presents some basic statistics for this sample correlation matrix; somewhat more interestingly, in Figure 3 we plot the empirical quantile function for the off-diagonal entries of this sample correlation matrix, *i.e.*, the x-axis represents a level $\alpha \in [0, 1]$, while the y-axis represents the smallest threshold $Q(\alpha)$ below which $(100 \cdot \alpha)$ percent of the entries fall. The gently-upward-sloping line segment in the middle of the plot indicates that lots of the entries in the sample correlation matrix are ≤ 0.6, meaning that most voxels are weakly correlated with most other voxels, *i.e.*, there is some strong signal in the data, but also a lot of noise.

Finally, we mention that the entire sample correlation matrix is available at:
`https://db.humanconnectome.org/data/projects/HCP_1200`.

[6] For the curious reader: the Human Connectome Project is related to but distinct from the BRAIN Initiative started by the Obama Administration, in 2012.

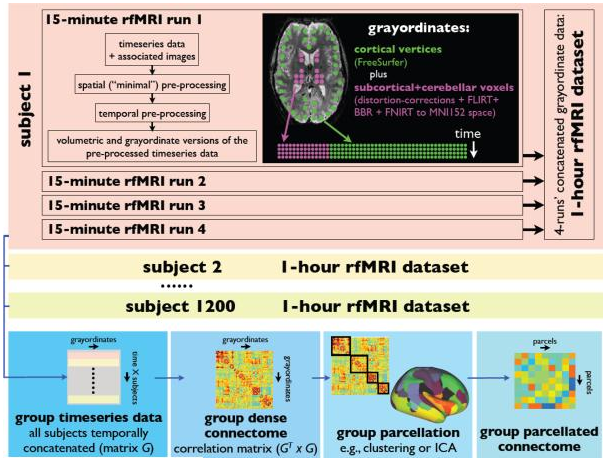


Figure 2: Left: an fMRI machine. Right: an overview of the process used to generate the sample correlation matrix we use; taken from Figure 2 of Smith et al. (2013).

Minimum value	-0.5355
Maximum value	1.0000
Average value	0.1803
Median value	0.1491
Standard deviation	0.1590

Table 1: Some basic statistics about the entries of the sample correlation matrix we use.

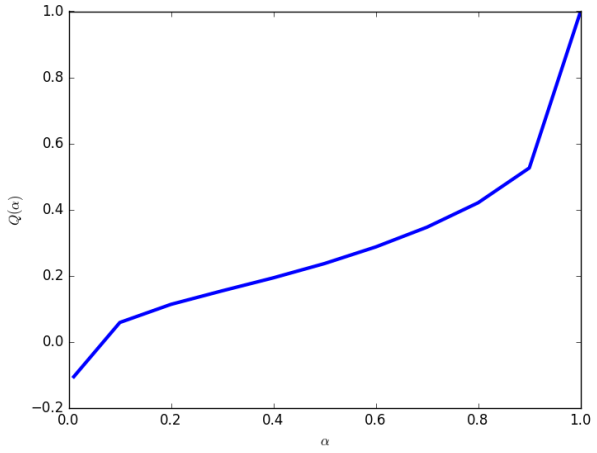


Figure 3: The empirical quantile function for the off-diagonal entries of the sample correlation matrix that we use, showing the level of signal vs. noise in the data.

6 Results and discussion

The top and middle rows of Table 2 present the best clusterings generated by PseudoNet followed by the persistent homology and Louvain methods, respectively, when compared to the clusterings of Glasser et al. (2016) presented in Figure 4, according to the (modified) Jaccard score. We additionally consider a simple baseline, given by discarding {99, 99.1, ..., 99.8, 99.9, 99.91, ..., 99.98, 99.99}% of the sample covariance matrix entries: i.e., keep entries with the largest magnitudes (c.f. Mazumder and Hastie (2012)) in order to generate (marginal) correlation graphs; this baseline lets us probe the comparative advantage of using marginal vs. partial correlations. The bottom row of Table 2 presents the best clusterings generated by this baseline, *i.e.*, thresholding the sample covariance matrix at various levels. The left and middle columns present the results for the left and right hemispheres, respectively. We see that the persistent homology clusterings perform the best, in terms of Jaccard score, across both hemispheres.

Qualitatively, we see that the persistent homology clusterings are able to identify several clusters of interest to the neuroscience community (c.f. Figure 3 in Glasser et al. (2016)); this is certainly encouraging, since we do not expect perfect recovery of all the clusters in Figure 4, as the latter clusters rely on a significant amount of domain knowledge. Some examples:

- Generally speaking, the Glasser et al. (2016), Louvain, and persistent homology method parcellations all exhibit some similarities across the temporal cortex (a part of the brain used for receiving information from the senses); Louvain perhaps does a little better than the persistent homology method here.
- Louvain and the persistent homology method both seem to pick up on area 55b, a part of the brain used for listening.
- Louvain and the persistent homology method both pick up on the lateral intraparietal cortex (LIPv), an area involved in controlling the eyes.
- The persistent homology method seems to identify the middle temporal visual area (MT), also known as area V5, involved in seeing moving objects, while Louvain appears to miss it.
- Finally, the persistent homology method picks up on Brodmann's area 44, used for listening and speaking, while Louvain misses it.

Overall, based on the above observations, it appears that the parcellation generated by the persistent homology method gets a slight edge over Louvain; on the other hand, the clusterings generated by the sample covariance matrix seem to miss the above clusters, as they appear overly smooth.

Lastly, it is also interesting to analyze the sparsity patterns of the PseudoNet and sample covariance matrix estimates yielding the best clusterings; these are presented in the right column of Table 2. Three features here are striking:

1. The PseudoNet estimates possess a block diagonal structure, where the blocks turn out to correspond to the left and right hemispheres.

2. Second, the sparsity patterns of the blocks themselves turn out to correspond to the (spatially) closest voxels. In Figure 5, we present the sparsity pattern, of the left hemisphere only (for simplicity), of the PseudoNet estimate associated with the top left plot of Table 2; in Figure 5, we also present the sparsity pattern of a matrix we constructed, where the (i, j)th entry of the matrix is the great-circle distance between the voxels i and j, after retaining only 0.1% of closest voxels. The sparsity patterns of the distance matrix and PseudoNet estimate indeed look visually similar, suggesting that the PseudoNet estimate has recovered some of the spatial signal in the data, without being "told" to do so. Inspecting the right hemisphere conveys the same message.

3. Third, although the sparsity patterns of the PseudoNet and sample covariance matrix estimates appear vaguely similar, the subtle differences between them drive the (significant) differences in the clusterings. We emphasize that these features arise naturally, without being hard-coded into our method.

The actual Jaccard scores, as well as a significantly expanded set of results, can be found in the supplement.

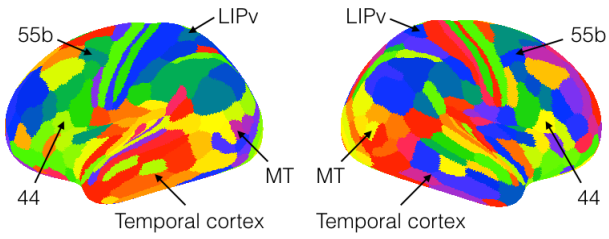
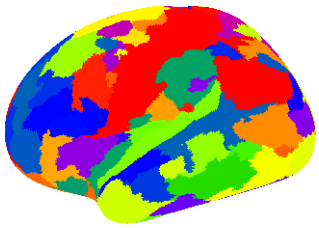
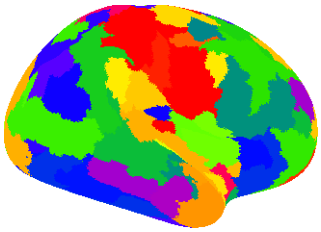


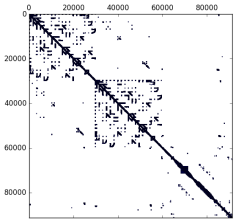
Figure 4: The clusterings from Glasser et al. (2016), for the left and right hemispheres of the brain; the clusterings were generated by applying a multi-class, shallow neural network to the same data we use (Smith et al., 2013), but use a significant amount of domain knowledge in order to post-processes the results by hand. The colors have no significance, except to demarcate the different clusters.



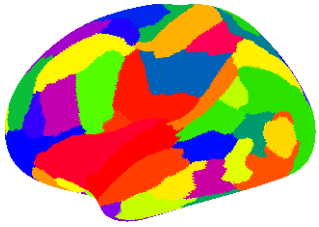
$\lambda_1 = 0.48,\ \lambda_2 = 0.39,\ \epsilon = 3,$
% of best score = 100



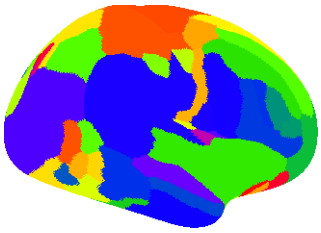
$\lambda_1 = 0.5,\ \lambda_2 = 0.39,\ \epsilon = 3,$
% of best score = 100



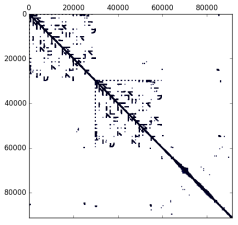
$\lambda_1 = 0.48,\ \lambda_2 = 0.39,\ \epsilon = 3,$
% of best score = 100



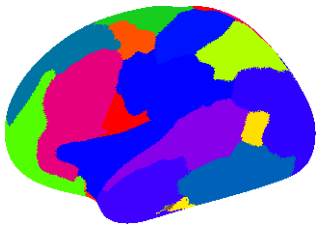
$\lambda_1 = 0.64,\ \lambda_2 = 0.13,\ k = 1,$
% of best score = 75.03



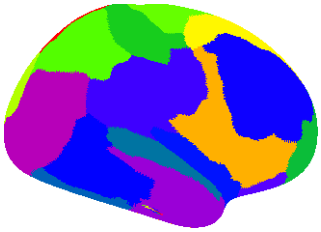
$\lambda_1 = 0.5425,\ \lambda_2 = 0.39,\ k = 0,$
% of best score = 73.45



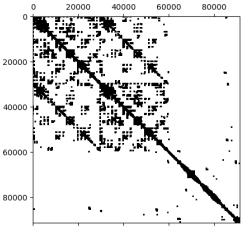
$\lambda_1 = 0.64,\ \lambda_2 = 0.13,\ k = 1,$
% of best score = 75.03



$t = 99.9,\ k = 4,$
% of best score = 32.24



$t = 99.9,\ k = 3,$
% of best score = 32.45



$t = 99.9,\ k = 4$
% of best score = 32.24

Table 2: Top row: the best clusterings generated by PseudoNet followed by the persistent homology method, relative to the clusterings of Glasser et al. (2016) presented in Figure 4, according to the (modified) Jaccard score; the left and middle columns present the results for the left and right hemispheres, respectively, while the right column presents the sparsity pattern (black indicates a nonzero entry) of the PseudoNet estimate yielding the best clustering for the left hemisphere (the sparsity patterns for the right hemisphere are in the supplement). Middle row: the same plots, except for PseudoNet followed by the Louvain method. Bottom row: the same plots, except generated by thresholding the sample covariance matrix at various levels. Indicated below each clustering is the percentage of the best Jaccard score it attains (higher is better); since the persistent homology clusterings perform the best, these percentages are just 100. The actual Jaccard scores, as well as a significantly expanded set of results, can be found in the supplement. Also indicated are the tuning parameter values yielding the clusterings (i.e., λ_1, λ_2 for PseudoNet; $\varepsilon \geq 0$, $k \in \mathbb{Z}_+$ controlling the number of clusters for the persistent homology and Louvain methods, respectively; and t denoting the percentage of discarded sample covariance matrix entries). The colors in the various plots have no special meaning.

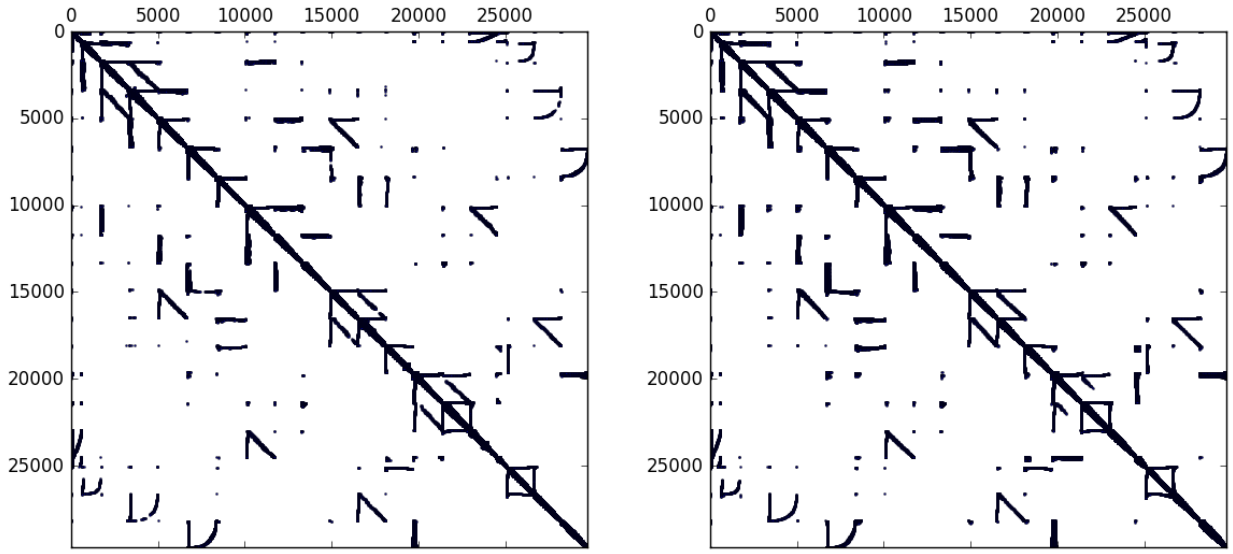


Figure 5: Left: the sparsity pattern of the PseudoNet estimate associated with the top left plot of Table 2, where we have only plotted the 29,696 coordinates belonging to the surface of the left hemisphere (for simplicity). Right: the sparsity pattern of a $(91,282 \times 91,282)$-dimensional (symmetric) matrix we constructed, whose (i,j)th entry is the great-circle distance between the voxels i and j, where we have (again) only plotted the 29,696 coordinates belonging to the left hemisphere, and additionally retained just the 0.1% of closest voxels. In both plots, black indicates a nonzero entry.

7 Conclusion

In this report, we looked at applying methods from statistics and numerical linear algebra to high-quality, high-dimensional fMRI data in order to generate a parcellation of the human brain; data-driven parcellations, like ours, are useful because they can reduce the need for laborious hand-crafted parcellations, and also help drive research agendas in neuroscience and medicine. Although the parcellations we generated were not perfect compared to Glasser et al. (2016), they still captured some features of the brain that are known to the neuroscience community; our approach is also general enough that it could be applied in other domains, *e.g.*, genetics, finance, and energy, without much modification, perhaps yielding insights there, as well. There is also certainly room to fine-tune our approach. For example, we could (and should) consider evaluating the statistical significance of our results; along these lines, it would be interesting to report results, where we use a model selection criterion to pick the "best" clustering (as opposed to searching for the best clustering, as we have done here). Finally, it would be interesting to apply the SpDM^3 method for computing the product $S\Omega$ (in a distributed fashion) to other sparse inverse covariance estimators, *e.g.*, a proximal gradient method applied to the graphical lasso criterion.

Acknowledgements

Alnur Ali was supported by a Department of Energy Computational Science Graduate Fellowship DE-FG02-97ER25308.

References

Ali, A., Khare, K., Oh, S.-Y., and Rajaratnam, B. (2016a). Generalized pseudolikelihood methods for inverse covariance estimation. Technical report. Available at `http://arxiv.org/pdf/1606.00033.pdf`.

Ali, A., Kolter, J. Z., and Tibshirani, R. J. (2016b). The multiple quantile graphical model. In *Advances in Neural Information Processing Systems*. Available at `http://arxiv.org/pdf/1607.00515.pdf`.

Belilovsky, E., Kastner, K., Varoquaux, G., and Blaschko, M. (2016). Learning to discover graphical model structures. Available at `https://arxiv.org/pdf/1605.06359.pdf`.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Eickhoff, S. B., Thirion, B., Varoquaux, G., and Bzdok, D. (2015). Connectivity-based parcellation: Critique and implications. *Human Brain Mapping*, 36(12):4771–4792.

Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Fu, Z., Han, S., Tan, A., Tu, Y., and Zhang, Z. (2015). ℓ_0-regularized time-varying sparse inverse covariance estimation for tracking dynamic fMRI brain networks. In *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1496–1499. IEEE.

Glasser, M. F., Coalson, T. S., Robinson, E. C., Hacker, C. D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C. F., Jenkinson, M., Smith, S. M., and Van Essen, D. C. (2016). A multimodal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., Ravikumar, P. K., and Poldrack, R. (2013). BIG & QUIC: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems*, pages 3165–3173.

Khare, K., Oh, S.-Y., and Rajaratnam, B. (2015). A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees. *Journal of the Royal Statistical Society: Series B*, 77(4):803–825.

Koanantakool, P., Azad, A., Buluç, A., Morozov, D., Oh, S.-Y., Oliker, L., and Yelick, K. (2016). Communication-avoiding parallel sparse-dense matrix-matrix multiplication. In *Proceedings of the 31st IEEE International Parallel and Distributed Processing Symposium*, pages 842–853. IEEE.

Kolar, M., Parikh, A. P., and Xing, E. P. (2010). On sparse nonparametric conditional covariance selection. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 559–566.

Mazumder, R. and Hastie, T. (2012). Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13(Mar):781–794.

Meilă, M. (2003). Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines*, pages 173–187. Springer.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Morozov, D. (2016). Dionysus 2 (a library for computing persistent homology. Available at `http://mrzv.org/software/dionysus2/`.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231.

Pearce, D. J. (2005). An improved algorithm for finding the strongly connected components of a directed graph. Technical report, Technical report, Victoria University, Wellington, NZ.

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746.

Ryali, S., Chen, T., Supekar, K., and Menon, V. (2012). Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty. *NeuroImage*, 59(4):3852–3861.

Smith, S. M., Beckmann, C. F., Andersson, J., Auerbach, E. J., Bijsterbosch, J., Douaud, G., Duff, E., Feinberg, D. A., Griffanti, L., Harms, M. P., et al. (2013). Resting-state fMRI in the human connectome project. *NeuroImage*, 80:144–168.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., and Tibshirani, R. (2012). Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B*, 74(2):245–266.

Wang, Y., Kang, J., Kemmer, P. B., and Guo, Y. (2016). An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation. *Frontiers in neuroscience*, 10.

Wasserman, L. (2016). Lecture notes on "undirected graphical models". Available at `http://www.stat.cmu.edu/~larry/=sml/GraphicalModels.pdf`.

Won, J., Lim, J., Kim, S., and Rajaratnam, B. (2013). Condition number-regularized covariance estimation. *Journal of the Royal Statistical Society: Series B*, 75(3):427–450.

Wytock, M. and Kolter, J. Z. (2013). Sparse Gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1265–1273.

Supplementary material

7.1 Where does the criterion in (1) come from?

Here, we motivate the criterion in (1) by showing that it follows quite naturally from basic arguments about partial correlations; some of these ideas are spread out across Khare et al. (2015) and Ali et al. (2016a), but we consolidate them here.

The connections between partial correlations and linear regression (*i.e.*, that the coefficients in a pseudolikelihood regression are related to the elements of the inverse covariance matrix as well as the partial correlations, in a specific way) are old; Equations 2 and 3 in Kolar et al. (2010) recall the ideas cleanly, and start us off on the right foot. From these equations, a reasonable criterion for sparse inverse covariance estimation might be

$$-\frac{1}{2}\log\det\Omega^2_{\text{diag}} + \frac{1}{2}\sum_{i=1}^{p}\Omega_{ii}^2\left\|X_i - \sum_{j\neq i}^{p} W_{ij}\sqrt{\frac{\Omega_{jj}}{\Omega_{ii}}}X_j\right\|_2^2 + \lambda_1\|\Omega_{\text{off}}\|_1 + \frac{\lambda_2}{2}\|\Omega\|_F^2,$$

with variables $W, \Omega \in \mathbf{R}^{p\times p}$, where W is a (symmetric) matrix containing the partial correlations. Above, the extra Ω_{ii}^2 term outside the squared Euclidean norm can be thought of as a weighting factor on each term in the sum, while the $-\log\det\Omega^2_{\text{diag}}$ term can be thought of as preventing the diagonal entries of Ω from getting too small.

Now, plugging in Equation 1 from Kolar et al. (2010), *i.e.*,

$$W_{ij} = \frac{-\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}},$$

we get that

$$
\begin{aligned}
&-\frac{1}{2}\log\det\Omega_{\text{diag}}^2+\frac{1}{2}\sum_{i=1}^{p}\Omega_{ii}^2\left\|X_i+\sum_{j\neq i}^{p}\frac{\Omega_{ij}}{\Omega_{ii}}X_j\right\|_2^2+\lambda_1\|\Omega_{\text{off}}\|_1+\frac{\lambda_2}{2}\|\Omega\|_F^2\\
&=-\frac{1}{2}\log\det\Omega_{\text{diag}}^2+\frac{1}{2}\sum_{i=1}^{p}\Omega_{ii}^2\left\|\frac{\Omega_{ii}}{\Omega_{ii}}\cdot\left(X_i+\sum_{j\neq i}^{p}\frac{\Omega_{ij}}{\Omega_{ii}}X_j\right)\right\|_2^2+\lambda_1\|\Omega_{\text{off}}\|_1+\frac{\lambda_2}{2}\|\Omega\|_F^2\\
&=-\frac{1}{2}\log\det\Omega_{\text{diag}}^2+\frac{1}{2}\sum_{i=1}^{p}\Omega_{ii}^2\left\|\frac{\Omega_{ii}}{\Omega_{ii}}\cdot X_i+\frac{1}{\Omega_{ii}}\cdot\sum_{j\neq i}^{p}\Omega_{ij}X_j\right\|_2^2+\lambda_1\|\Omega_{\text{off}}\|_1+\frac{\lambda_2}{2}\|\Omega\|_F^2\\
&=-\frac{1}{2}\log\det\Omega_{\text{diag}}^2+\frac{1}{2}\sum_{i=1}^{p}\Omega_{ii}^2\cdot\frac{1}{\Omega_{ii}^2}\left\|\Omega_{ii}X_i+\sum_{j\neq i}^{p}\Omega_{ij}X_j\right\|_2^2+\lambda_1\|\Omega_{\text{off}}\|_1+\frac{\lambda_2}{2}\|\Omega\|_F^2\\
&=-\frac{1}{2}\log\det\Omega_{\text{diag}}^2+\frac{1}{2}\sum_{i=1}^{p}\left\|\sum_{j=1}^{p}\Omega_{ij}X_j\right\|_2^2+\lambda_1\|\Omega_{\text{off}}\|_1+\frac{\lambda_2}{2}\|\Omega\|_F^2\\
&=-\frac{1}{2}\log\det\Omega_{\text{diag}}^2+\frac{n}{2}\,\mathbf{Tr}\,S\Omega^2+\lambda_1\|\Omega_{\text{off}}\|_1+\frac{\lambda_2}{2}\|\Omega\|_F^2,
\end{aligned}
$$

which is precisely the criterion in (1); the criterion can also be seen as a generalization of the criterion for the graphical lasso.

7.2 Expanded set of results

Here, we provide an expanded set of figures and tables, that present (some more of) the clusterings generated by PseudoNet, as well as thresholding the sample covariance matrix at various levels, followed by the persistent homology and Louvain methods, at some values of each method's tuning parameters; our intent is simply to give the reader a sense of how the clusterings can vary with each method's tuning parameters. To help orient the reader, we include the following guide to the expanded results.

- Table 3 presents the clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 13 presents the Jaccard scores (2) for these clusterings.

- Table 4 presents the clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 14 presents the Jaccard scores (2) for these clusterings.

- Table 5 presents the clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 15 presents the Jaccard scores (2) for these clusterings.

- Table 6 presents the clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 0$ (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 16 presents the Jaccard scores (2) for these clusterings.

- Table 7 presents the clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: $k = 0$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 17 presents the Jaccard scores (2) for these clusterings.

- Table 8 presents the clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: $k = 0$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 18 presents the Jaccard scores (2) for these clusterings.

- Table 9 presents the clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: the largest value of k considered by Louvain (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 19 presents the Jaccard scores (2) for these clusterings.

- Table 10 presents the clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: the largest value of k considered by Louvain (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 20 presents the Jaccard scores (2) for these clusterings.

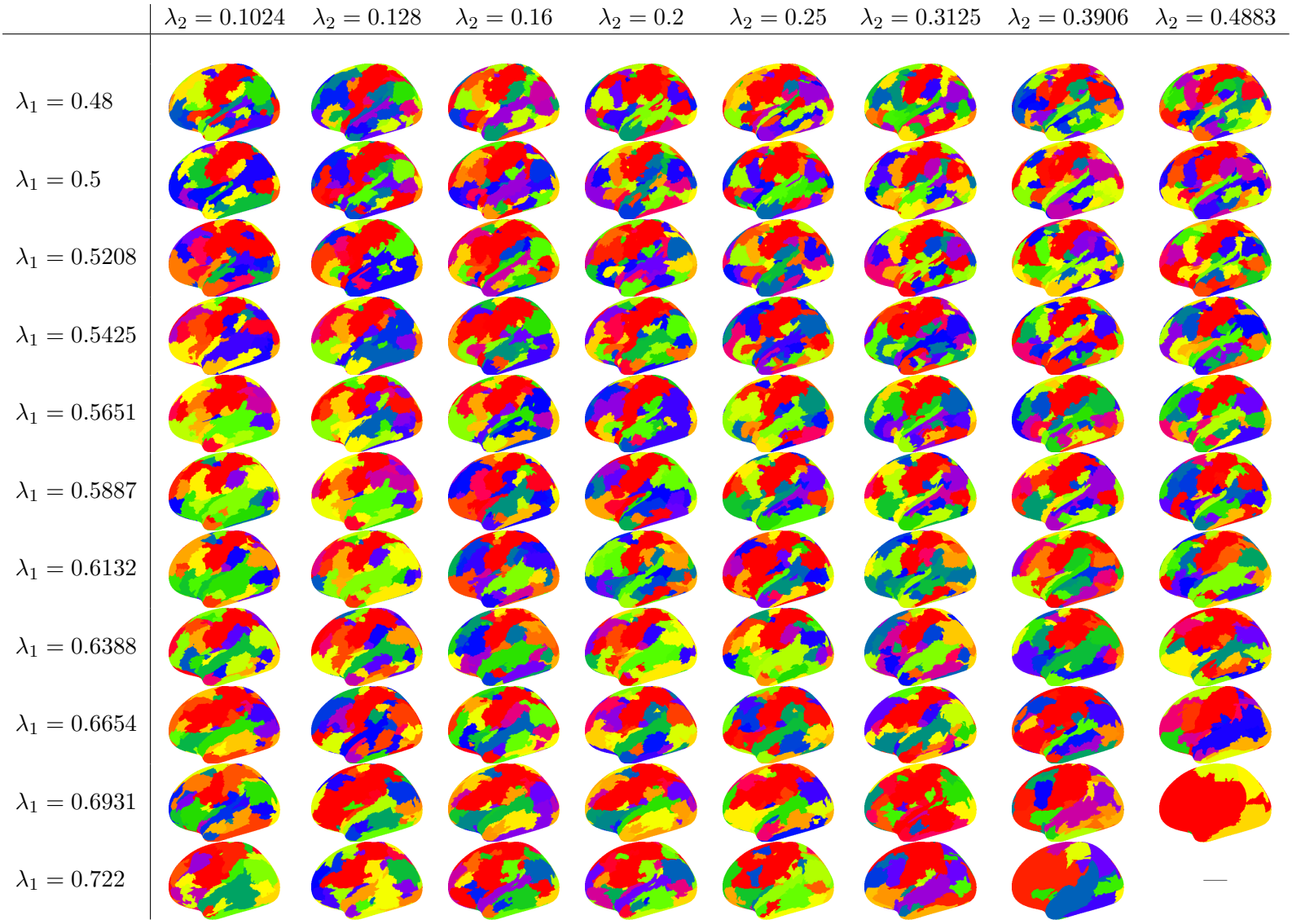


Table 3: The clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 13 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

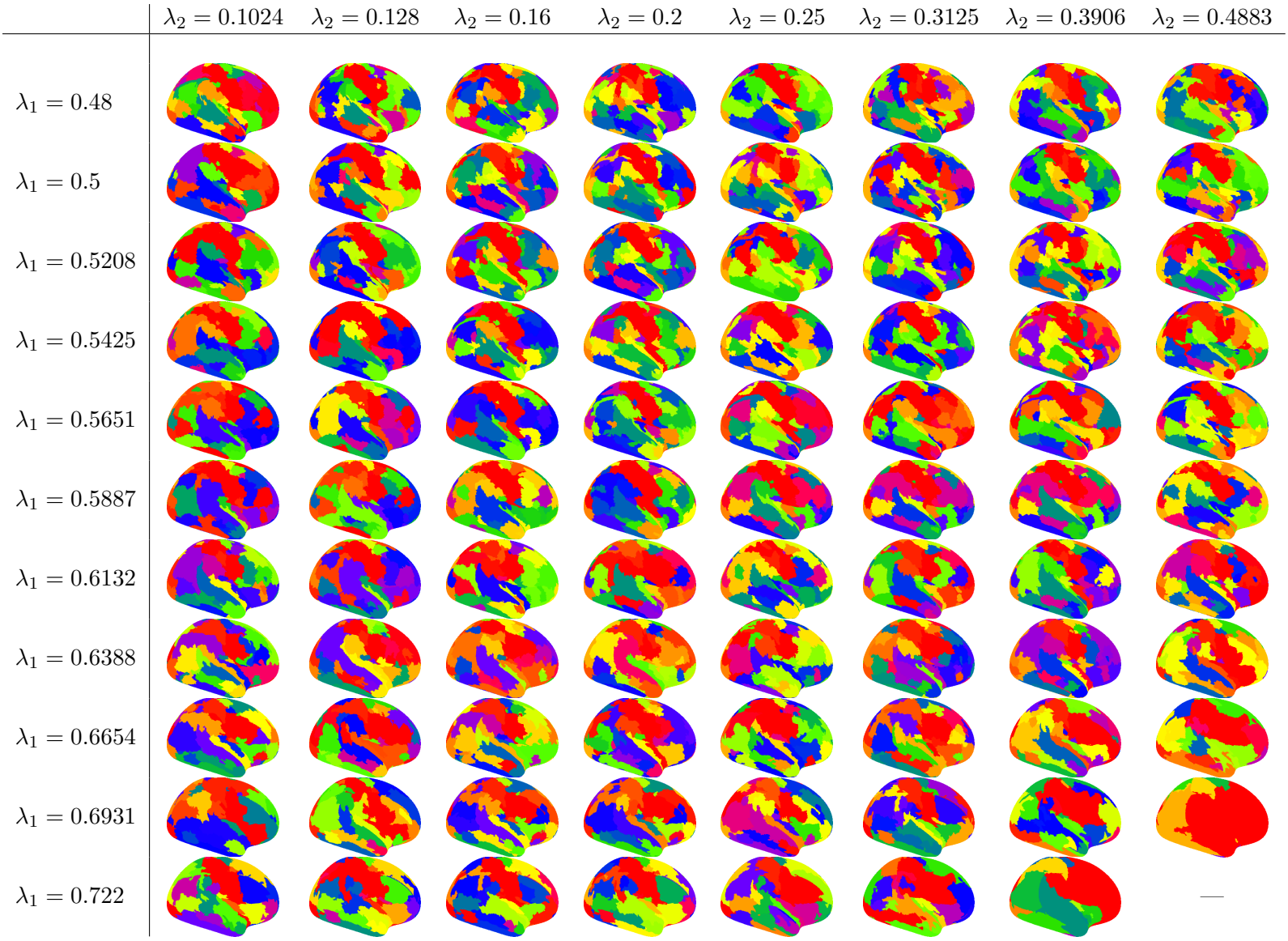


Table 4: The clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 14 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

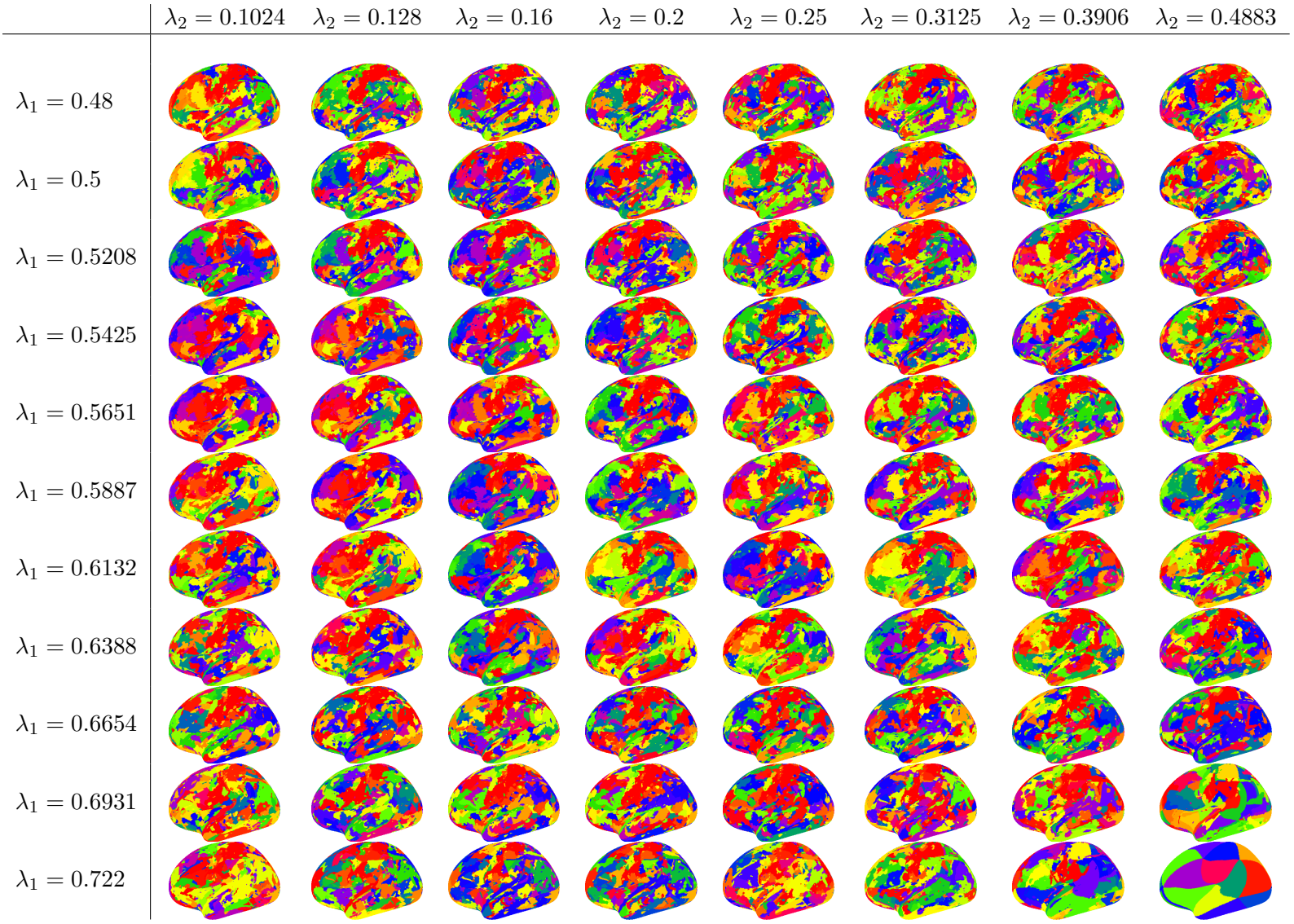


Table 5: The clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 15 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

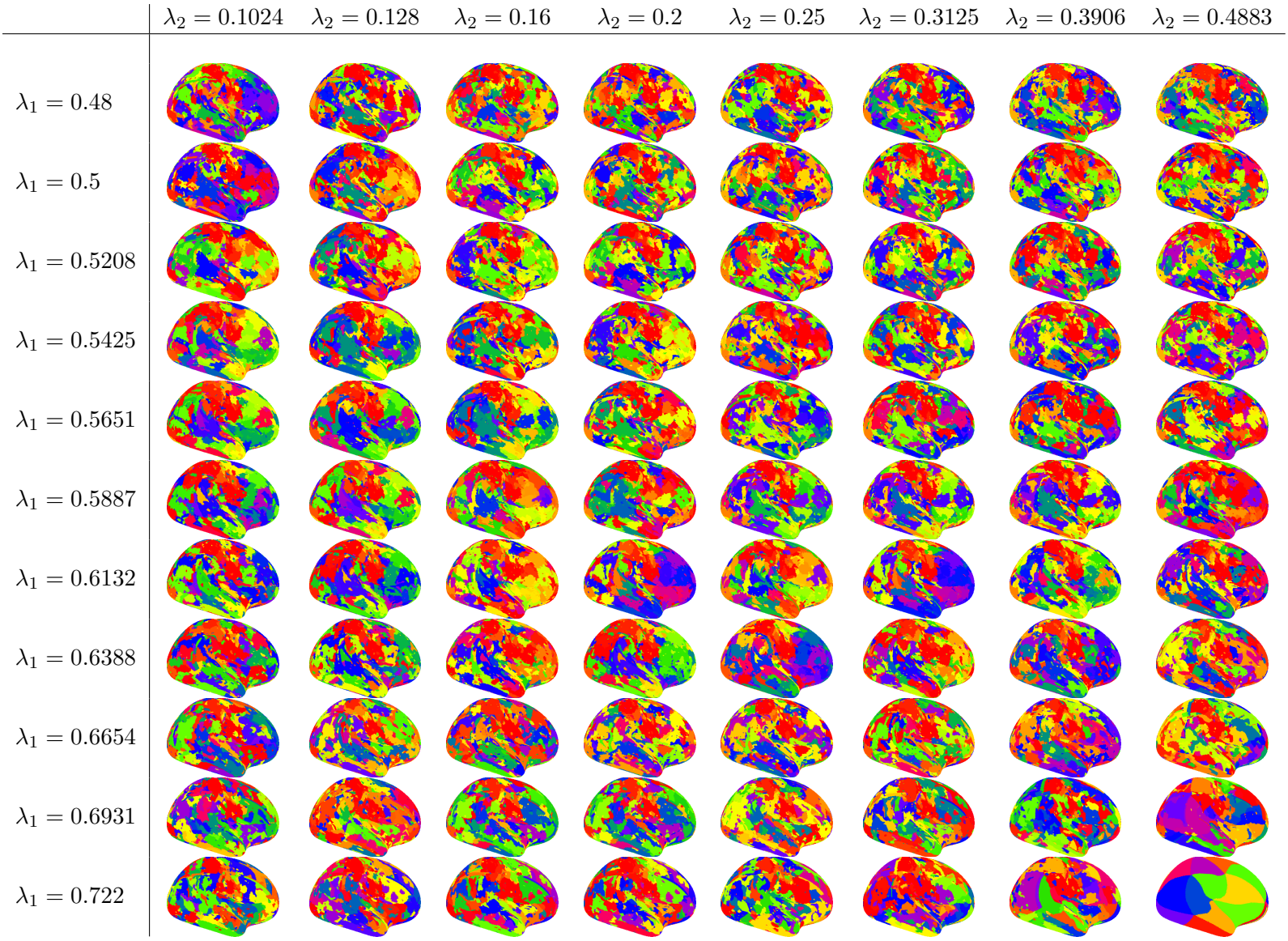


Table 6: The clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 0$ (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 16 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

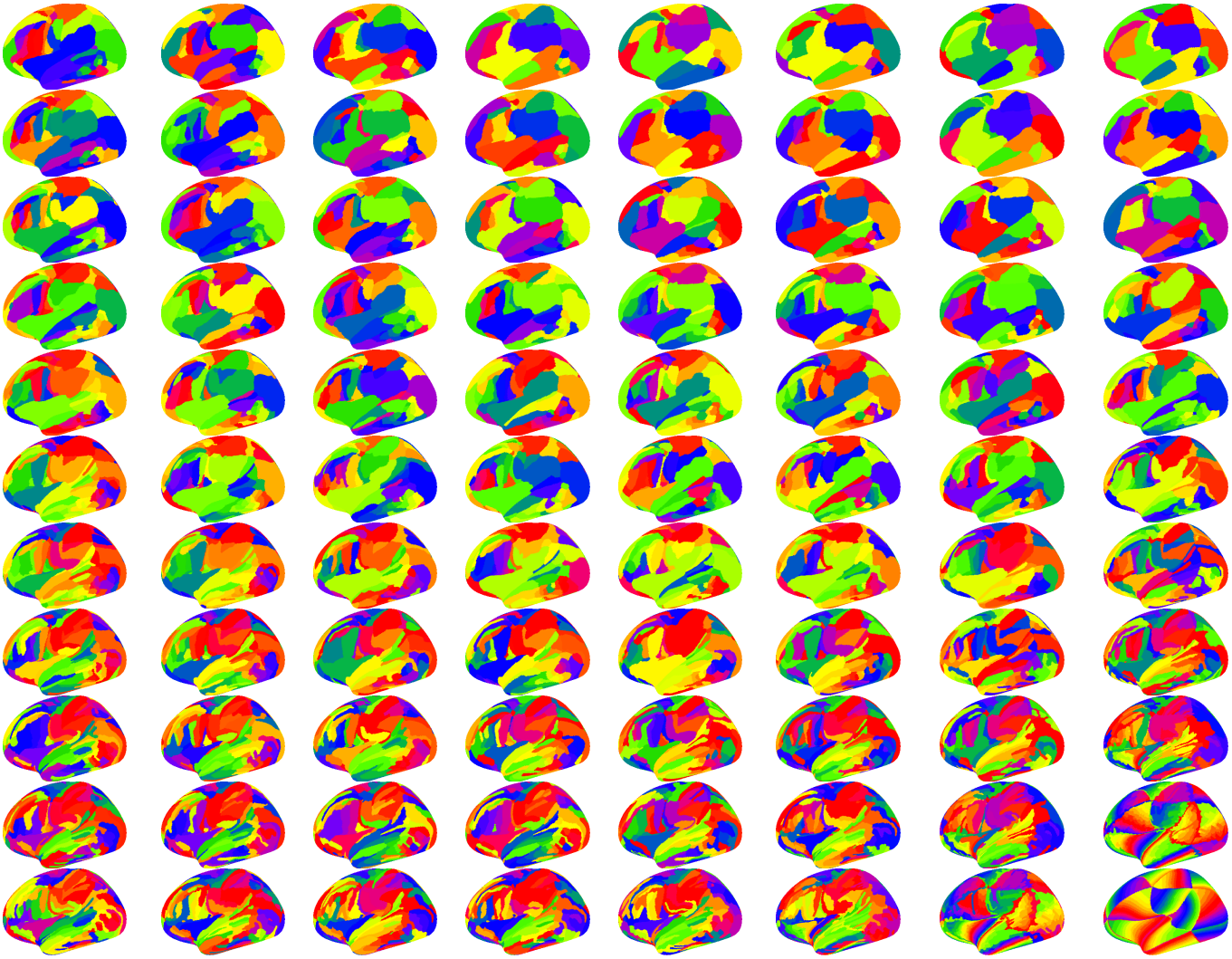
	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$								
$\lambda_1 = 0.5$								
$\lambda_1 = 0.5208$								
$\lambda_1 = 0.5425$								
$\lambda_1 = 0.5651$								
$\lambda_1 = 0.5887$								
$\lambda_1 = 0.6132$								
$\lambda_1 = 0.6388$								
$\lambda_1 = 0.6654$								
$\lambda_1 = 0.6931$								
$\lambda_1 = 0.722$								

Table 7: The clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: $k = 0$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 17 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

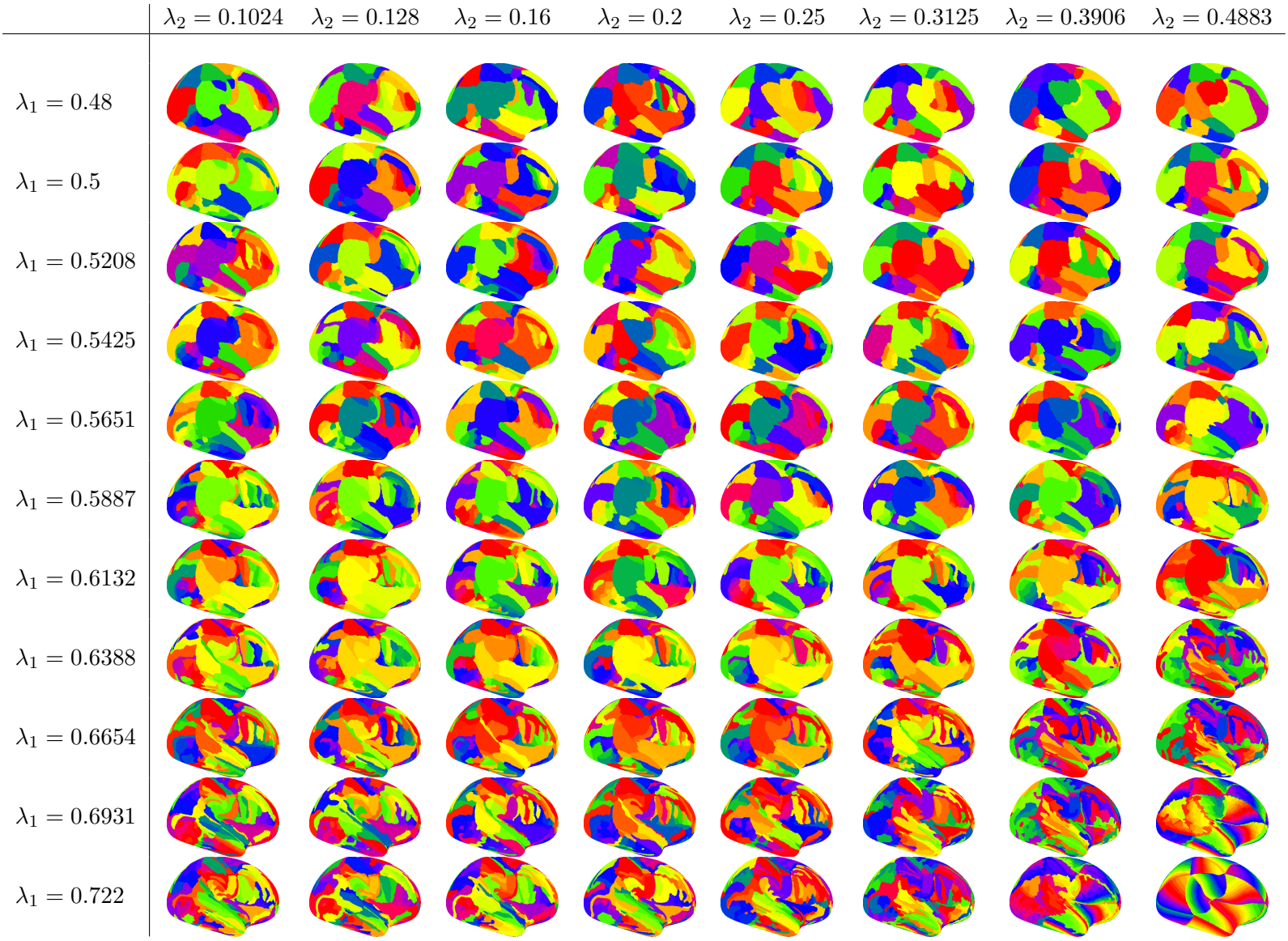


Table 8: The clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: $k = 0$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 18 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

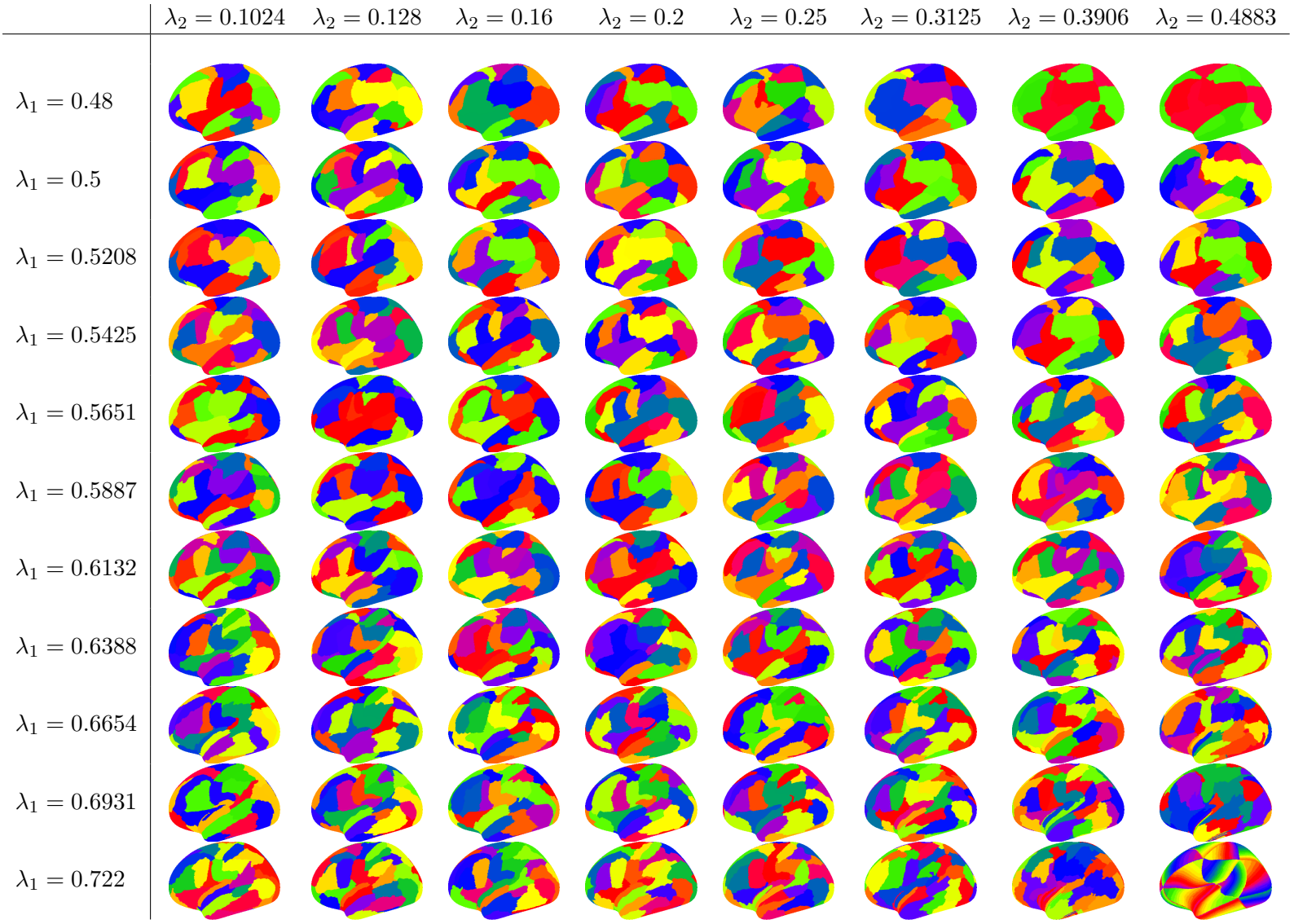


Table 9: The clusterings, for the *left* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: the largest value of k considered by Louvain (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 19 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

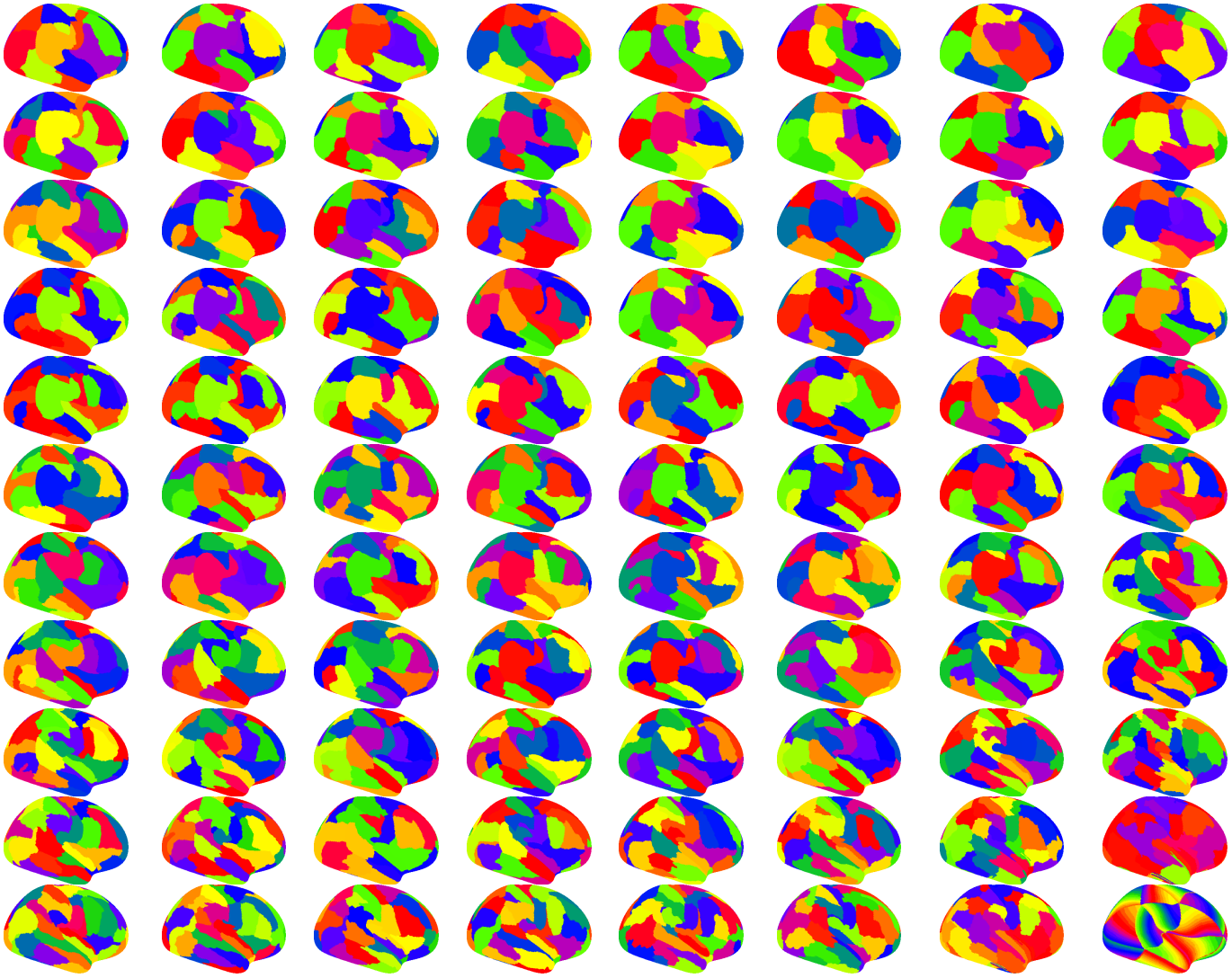
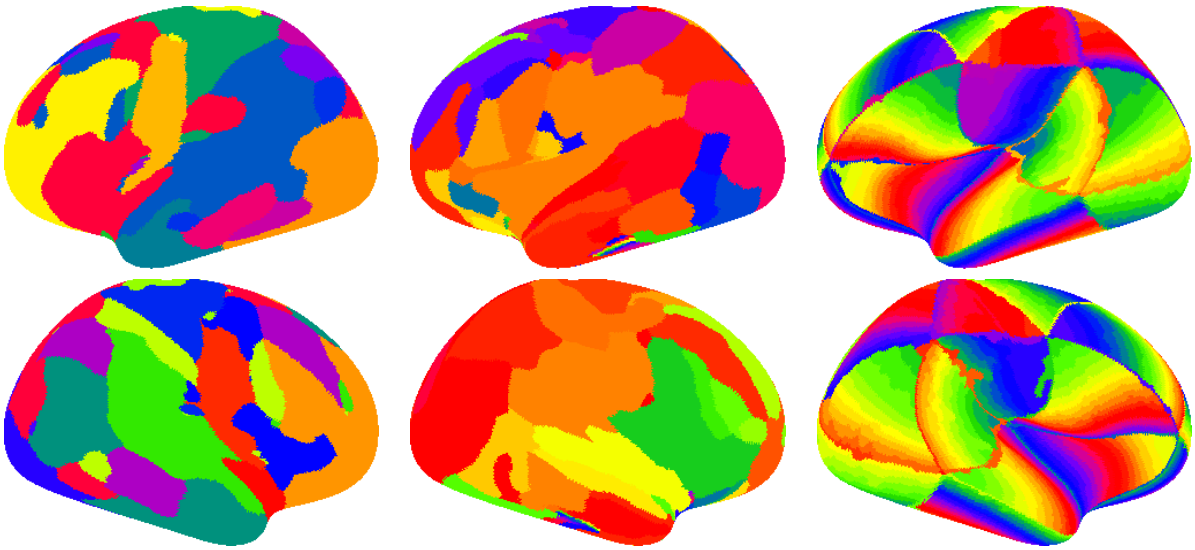
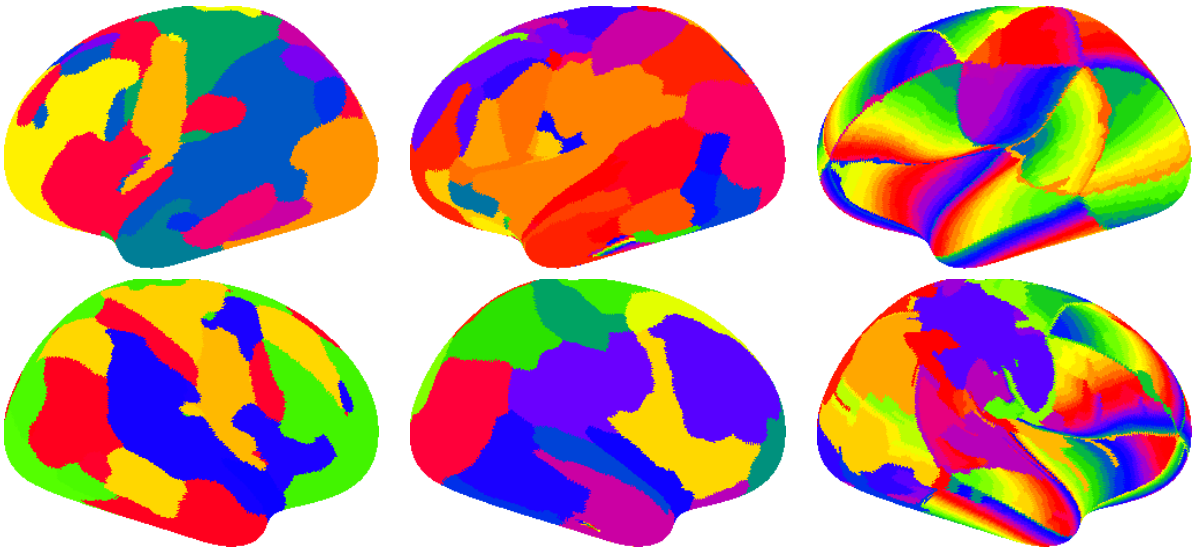
	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$								
$\lambda_1 = 0.5$								
$\lambda_1 = 0.5208$								
$\lambda_1 = 0.5425$								
$\lambda_1 = 0.5651$								
$\lambda_1 = 0.5887$								
$\lambda_1 = 0.6132$								
$\lambda_1 = 0.6388$								
$\lambda_1 = 0.6654$								
$\lambda_1 = 0.6931$								
$\lambda_1 = 0.722$								

Table 10: The clusterings, for the *right* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: the largest value of k considered by Louvain (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. Table 20 presents the Jaccard scores (2) for these clusterings. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.



$t = 99$ $t = 99.9$ $t = 99.99$

Table 11: The clusterings, for the *left* and *right* hemispheres (see the top and bottom rows, respectively), generated by thresholding the sample covariance matrix at various levels t followed by the *Louvain* method, at the tuning parameter values: $k = 0$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4.



$t = 99$ $t = 99.9$ $t = 99.99$

Table 12: The clusterings, for the *left* and *right* hemispheres (see the top and bottom rows, respectively), generated by thresholding the sample covariance matrix at various levels t followed by the *Louvain* method, at the tuning parameter values: the largest value of k considered by Louvain (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.2043	0.2199	0.2242	0.2326	0.2277	0.2422	0.2447	0.23
$\lambda_1 = 0.5$	0.2112	0.224	0.2315	0.2329	0.2343	0.2283	0.2197	0.2185
$\lambda_1 = 0.5208$	0.1964	0.1895	0.2264	0.2385	0.2317	0.2282	0.2348	0.2358
$\lambda_1 = 0.5425$	0.1905	0.1972	0.1951	0.2181	0.2268	0.2295	0.2289	0.2255
$\lambda_1 = 0.5651$	0.1833	0.197	0.1973	0.1981	0.2125	0.2268	0.2242	0.2213
$\lambda_1 = 0.5887$	0.1838	0.1845	0.1992	0.2067	0.1953	0.2057	0.2078	0.2155
$\lambda_1 = 0.6132$	0.1702	0.1752	0.198	0.1995	0.2121	0.2014	0.2036	0.1891
$\lambda_1 = 0.6388$	0.1698	0.1693	0.1864	0.1837	0.1859	0.191	0.1831	0.1785
$\lambda_1 = 0.6654$	0.1538	0.1854	0.1759	0.1701	0.1748	0.1844	0.1805	0.1467
$\lambda_1 = 0.6931$	0.1652	0.1689	0.1664	0.1686	0.1722	0.162	0.1472	0.0516
$\lambda_1 = 0.722$	0.1382	0.1536	0.1556	0.1536	0.1442	0.1394	0.0758	—

Table 13: The Jaccard scores (2) for the clusterings of the *left* hemisphere in Table 3, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.2258	0.2315	0.2461	0.2279	0.2451	0.2436	0.2311	0.2431
$\lambda_1 = 0.5$	0.2036	0.2245	0.2328	0.2326	0.2427	0.2314	0.2654	0.2528
$\lambda_1 = 0.5208$	0.2255	0.2166	0.2317	0.2311	0.2427	0.2399	0.2381	0.2417
$\lambda_1 = 0.5425$	0.21	0.2172	0.232	0.2355	0.2279	0.2299	0.245	0.2349
$\lambda_1 = 0.5651$	0.2233	0.2182	0.2236	0.2341	0.2367	0.231	0.2286	0.2413
$\lambda_1 = 0.5887$	0.2055	0.2187	0.2179	0.2369	0.2261	0.2321	0.2279	0.2067
$\lambda_1 = 0.6132$	0.1843	0.2002	0.2245	0.2224	0.2113	0.219	0.2256	0.21
$\lambda_1 = 0.6388$	0.1817	0.1843	0.2024	0.204	0.2154	0.2161	0.1981	0.1826
$\lambda_1 = 0.6654$	0.1786	0.1678	0.1824	0.1891	0.1952	0.1749	0.1851	0.1273
$\lambda_1 = 0.6931$	0.1652	0.1714	0.1686	0.1736	0.1714	0.1702	0.1284	0.061
$\lambda_1 = 0.722$	0.1372	0.1562	0.162	0.1563	0.1364	0.1264	0.0875	—

Table 14: The Jaccard scores (2) for the clusterings of the *right* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.0507	0.051	0.0527	0.0532	0.0511	0.0503	0.051	0.0518
$\lambda_1 = 0.5$	0.053	0.0518	0.052	0.0519	0.0526	0.0531	0.0517	0.0516
$\lambda_1 = 0.5208$	0.0517	0.0519	0.0527	0.0517	0.0524	0.0526	0.0522	0.0536
$\lambda_1 = 0.5425$	0.0522	0.0519	0.0509	0.0516	0.0516	0.0514	0.0532	0.0533
$\lambda_1 = 0.5651$	0.0514	0.0524	0.0512	0.0528	0.0529	0.0518	0.0518	0.0533
$\lambda_1 = 0.5887$	0.0498	0.0524	0.0534	0.0521	0.0522	0.0521	0.0526	0.0532
$\lambda_1 = 0.6132$	0.0504	0.0501	0.0531	0.052	0.0523	0.0529	0.0523	0.0505
$\lambda_1 = 0.6388$	0.053	0.052	0.0494	0.0502	0.0517	0.0502	0.0516	0.0543
$\lambda_1 = 0.6654$	0.0526	0.0529	0.0536	0.0533	0.0537	0.0506	0.0535	0.0558
$\lambda_1 = 0.6931$	0.0529	0.054	0.0518	0.052	0.0532	0.0543	0.0566	0.0815
$\lambda_1 = 0.722$	0.0549	0.0528	0.0525	0.0534	0.056	0.0561	0.0718	0.0884

Table 15: The Jaccard scores (2) for the clusterings of the *left* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 3$ (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.0532	0.0506	0.0519	0.0516	0.0522	0.0521	0.0516	0.0508
$\lambda_1 = 0.5$	0.0538	0.0536	0.0513	0.0512	0.0515	0.0525	0.0524	0.0506
$\lambda_1 = 0.5208$	0.052	0.0519	0.0521	0.0509	0.0527	0.0517	0.0522	0.0509
$\lambda_1 = 0.5425$	0.0505	0.0523	0.0528	0.0532	0.0511	0.0529	0.0526	0.0522
$\lambda_1 = 0.5651$	0.0523	0.0501	0.0513	0.0513	0.053	0.0521	0.0512	0.0528
$\lambda_1 = 0.5887$	0.0516	0.0528	0.0504	0.0515	0.0518	0.0515	0.0523	0.0511
$\lambda_1 = 0.6132$	0.0505	0.0517	0.0525	0.0534	0.0511	0.0516	0.0543	0.0534
$\lambda_1 = 0.6388$	0.0514	0.0543	0.0516	0.0522	0.0519	0.0533	0.0532	0.0544
$\lambda_1 = 0.6654$	0.0544	0.0528	0.0514	0.0518	0.0525	0.0529	0.0565	0.061
$\lambda_1 = 0.6931$	0.0527	0.0555	0.0525	0.0528	0.055	0.0529	0.0597	0.0845
$\lambda_1 = 0.722$	0.0535	0.0524	0.0535	0.0527	0.0553	0.0566	0.0698	0.0918

Table 16: The Jaccard scores (2) for the clusterings of the *right* hemisphere, generated by PseudoNet followed by the *persistent homology* method, at the tuning parameter values: $\varepsilon = 0$ (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.1069	0.1101	0.0956	0.1042	0.1015	0.0958	0.0901	0.0905
$\lambda_1 = 0.5$	0.1123	0.1076	0.1065	0.102	0.1089	0.1053	0.1007	0.107
$\lambda_1 = 0.5208$	0.1097	0.111	0.1092	0.1096	0.1065	0.0982	0.1001	0.105
$\lambda_1 = 0.5425$	0.1313	0.1123	0.1148	0.1085	0.1166	0.1143	0.1065	0.117
$\lambda_1 = 0.5651$	0.1258	0.1216	0.1134	0.1167	0.1164	0.1097	0.1151	0.12
$\lambda_1 = 0.5887$	0.129	0.1228	0.1233	0.1091	0.1203	0.1205	0.1238	0.1188
$\lambda_1 = 0.6132$	0.1337	0.1294	0.1298	0.1289	0.1185	0.1285	0.1231	0.1455
$\lambda_1 = 0.6388$	0.1477	0.1368	0.1363	0.1296	0.131	0.1344	0.1473	0.1517
$\lambda_1 = 0.6654$	0.1486	0.1486	0.1458	0.1405	0.1488	0.1534	0.1486	0.1583
$\lambda_1 = 0.6931$	0.1469	0.1453	0.1512	0.1483	0.146	0.1627	0.1706	0.0273
$\lambda_1 = 0.722$	0.1581	0.1608	0.1557	0.1608	0.1661	0.1779	0.0461	0.0061

Table 17: The Jaccard scores (2) for the clusterings of the *left* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: $k = 0$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.1105	0.1014	0.1034	0.1042	0.0988	0.0976	0.0976	0.0935
$\lambda_1 = 0.5$	0.1084	0.1064	0.1011	0.1105	0.1003	0.1012	0.0992	0.1022
$\lambda_1 = 0.5208$	0.1263	0.1059	0.1195	0.1056	0.0995	0.1059	0.1	0.1051
$\lambda_1 = 0.5425$	0.122	0.1167	0.1113	0.1111	0.0997	0.1085	0.1125	0.0997
$\lambda_1 = 0.5651$	0.1219	0.1212	0.1144	0.1022	0.1044	0.1109	0.1029	0.1189
$\lambda_1 = 0.5887$	0.1218	0.1205	0.1184	0.1219	0.1159	0.1202	0.1183	0.135
$\lambda_1 = 0.6132$	0.132	0.1259	0.1339	0.1265	0.1269	0.124	0.1294	0.1361
$\lambda_1 = 0.6388$	0.1362	0.1364	0.1289	0.1286	0.1318	0.1279	0.1357	0.158
$\lambda_1 = 0.6654$	0.1483	0.1451	0.1428	0.142	0.1438	0.1498	0.1626	0.1675
$\lambda_1 = 0.6931$	0.1518	0.1552	0.1451	0.1473	0.1552	0.1671	0.1736	0.027
$\lambda_1 = 0.722$	0.1648	0.1725	0.1556	0.1607	0.1643	0.1758	0.0482	0.0061

Table 18: The Jaccard scores (2) for the clusterings of the *right* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: $k = 0$ (generally corresponding to *fewer* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.1678	0.1666	0.154	0.14	0.1297	0.135	0.1311	0.1284
$\lambda_1 = 0.5$	0.1632	0.1778	0.1595	0.1515	0.1537	0.1454	0.128	0.1422
$\lambda_1 = 0.5208$	0.1578	0.1719	0.1589	0.1663	0.1604	0.1572	0.145	0.1609
$\lambda_1 = 0.5425$	0.1538	0.1572	0.166	0.1542	0.1702	0.1689	0.1684	0.1569
$\lambda_1 = 0.5651$	0.1503	0.1602	0.1541	0.1473	0.1561	0.1587	0.1502	0.155
$\lambda_1 = 0.5887$	0.1526	0.158	0.1537	0.1622	0.1564	0.1547	0.149	0.1338
$\lambda_1 = 0.6132$	0.1438	0.1425	0.154	0.1487	0.151	0.1489	0.1327	0.1191
$\lambda_1 = 0.6388$	0.1414	0.1453	0.134	0.1431	0.1393	0.1357	0.1238	0.0967
$\lambda_1 = 0.6654$	0.1252	0.1263	0.1403	0.1301	0.1279	0.1196	0.0987	0.0653
$\lambda_1 = 0.6931$	0.1137	0.1159	0.1161	0.1163	0.109	0.0937	0.0701	0.0216
$\lambda_1 = 0.722$	0.1008	0.1005	0.1015	0.0961	0.0891	0.0679	0.0298	0.0061

Table 19: The Jaccard scores (2) for the clusterings of the *left* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: the largest value of k considered by Louvain (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.

	$\lambda_2 = 0.1024$	$\lambda_2 = 0.128$	$\lambda_2 = 0.16$	$\lambda_2 = 0.2$	$\lambda_2 = 0.25$	$\lambda_2 = 0.3125$	$\lambda_2 = 0.3906$	$\lambda_2 = 0.4883$
$\lambda_1 = 0.48$	0.1719	0.1697	0.1633	0.1763	0.1499	0.1475	0.1442	0.1365
$\lambda_1 = 0.5$	0.1689	0.1675	0.167	0.17	0.1661	0.1587	0.1411	0.1729
$\lambda_1 = 0.5208$	0.1651	0.1581	0.1808	0.1694	0.1655	0.1528	0.153	0.1512
$\lambda_1 = 0.5425$	0.1697	0.1556	0.1634	0.1637	0.1591	0.1581	0.1857	0.1598
$\lambda_1 = 0.5651$	0.1651	0.1663	0.1509	0.1554	0.1567	0.1542	0.151	0.1416
$\lambda_1 = 0.5887$	0.1492	0.1602	0.1635	0.1541	0.1512	0.1586	0.1506	0.1536
$\lambda_1 = 0.6132$	0.1474	0.1596	0.1586	0.1593	0.1649	0.1548	0.1436	0.1168
$\lambda_1 = 0.6388$	0.1321	0.1337	0.1495	0.1502	0.1458	0.1272	0.1188	0.0938
$\lambda_1 = 0.6654$	0.119	0.1203	0.1233	0.1221	0.1185	0.1125	0.0973	0.0635
$\lambda_1 = 0.6931$	0.112	0.1136	0.1128	0.111	0.107	0.0932	0.0694	0.0213
$\lambda_1 = 0.722$	0.0943	0.098	0.0994	0.0937	0.0832	0.0672	0.0299	0.0061

Table 20: The Jaccard scores (2) for the clusterings of the *right* hemisphere, generated by PseudoNet followed by the *Louvain* method, at the tuning parameter values: the largest value of k considered by Louvain (generally corresponding to *more* clusters) as well as all the λ_1, λ_2 values we describe in Section 4. "—", if present, indicates a degenerate clustering that puts either all the voxels into a single cluster or each voxel into its own cluster.