Machine learning application to human brain network studies: a kernel approach

Anvar Kurmukov, Yulia Dodonova, and Leonid Zhukov

National Research University Higher School of Economics, Russia
kurmukovai@gmail.com, ya.dodonova@mail.ru, lzhukov@hse.ru

Abstract. We consider a task of predicting normal and pathological phenotypes from macroscale human brain networks. These networks (connectomes) represent aggregated neural pathways between brain regions. We point to properties of connectomes that make them different from graphs arising in other application areas of network science. We discuss how machine learning can be organized on brain networks and focus on kernel classification methods. We describe different kernels on brain networks, including those that use information about similarity in spectral distributions of brain graphs and distances between optimal partitions of connectomes. We compare performance of the reviewed kernels in tasks of classifying autism spectrum disorder versus typical development and carriers versus non-carriers of an allele associated with an increased risk of Alzheimer's disease.

Keywords: machine learning, brain networks, classification, kernel SVM, graph spectra, clustering

1 Introduction

Recently, network representation of human brains (called connectomes) has gained increasing attention in neuroscience research. One of the challenges posed by connectomics is classification of normal and pathological phenotypes based on brain networks [1]. Mathematically, this is a problem of classifying small undirected connected graphs with uniquely labeled nodes.

We start this paper with a brief overview of pitfalls common to all machine learning studies on neuroimaging data. We describe how brain networks can be constructed based on magnetic resonance images (MRI) and discuss why these networks differ from graphs arising in other application areas of network science, such as chemistry or molecular biology. We next focus on a kernel approach to classification of brain networks. We adopt kernels previously described in other contexts and also review kernels proposed in our previous studies specifically for brain networks. We compare performance of these kernels based on two real-life datasets of structural connectomes.

2 Machine learning application to neuroimaging data

Machine learning based on neuroimaging data is becoming increasingly popular; until recently, group-level statistical comparisons dominated the field. A paper [2] discusses this fundamental shift in paradigm and also highlights some pitfalls of neuroimaging-based machine learning studies. For example, these include normal anatomical inter-individual variability which can mask disease-related changes, or normal inter-individual variation in cognitive reserve which adds a lot of uncertainty to the reference standards that are based on clinical diagnoses. Also, important part of variability in neuroimaging data stems from patient selection, inter-scanner variability and data preprocessing.

A caveat of neuroimaging-based machine learning studies is also a dysbalance between a dimensionality of the feature space and the number of subjects, and hence the problem of data reduction and a high risk of overfitting. A review [3] gives a good idea of the sample sizes typical for machine learning studies in the field of neuroscience. The authors [3] summarize 118 studies that used machine learning algorithms to predict psychiatric diagnoses based on neuroimaging data. Sample sizes in a majority of those studies did not exceed 100 participants, and most of the studies were based on less than 50 participants.

Finally, a most recent comprehensive review of neuroimaging-based single subject prediction of brain disorders can be found in [4]. Based on the analysis of more than 200 papers, the authors discuss several biases common for neuroimaging-based machine learning studies, such as a feature selection bias and an issue of hyperparameter optimization. Again, the authors [4] emphasize that the main bottleneck of this field is the limited sample size.

Importantly, the majority of studies in the area deal with voxel-level and region-level features. The former include features that are extracted at the level of individual voxels, such as voxel brightness or fractional anisotropy computed based on diffusion tensor imaging (DTI). Region-based features (e.g., region volumes or region average thicknesses) are derived by parceling brain images into zones, for example on the basis of a standardized brain atlas.

However, there exists an alternative way of representing human brains that makes full usage of network science concepts and ideas. We discuss this approach (called connectomics) in the next section.

3 Network representation of a human brain

A term connectome was proposed by [5] and [6]. It stands for a network that represents brain regions and their interconnections. For very simple organisms, such as Caenorhabditis elegans, these connections can be modeled at the level of individual neurons. For human brains, connectomes represent aggregated neural pathways at the macroscopic scale. For a review of this rapidly evolving research area, we refer to [1].

To produce human structural connectomes, brain gray matter is identified on MRI scans using a segmentation algorithm and is next parceled into regions

according to a brain atlas. These regions are the nodes of the constructed network. White matter streamlines are detected using a tractography algorithm. The number of streamlines that connect each pair of brain regions produces a weight for an edge between the respective nodes.

The above pipeline produces DTI-based structural connectomes. It is also possible to define so-called functional connectomes based on the fMRI scans. In this case, strength of co-activation of each pair of the regions provides weights for the edges. For a review on network modeling methods on fMRI data, we refer to [7]; we do not discuss this approach here.

Since the structural connectome is a discrete mathematical model of a human brain, the algorithms of discretization chosen in a given study largely affect the size and the structure of the resulting brain networks (e.g., see [8] for a discussion on methodological pitfalls of connectome construction). First, there is no unique way to define a set of nodes for brain graphs; we refer to [9] for a discussion on how the choice of nodal scale and gray-matter parcellation scheme affects the structure and topological properties of whole-brain structural networks.

Second, network edges can be defined differently depending on a tractography algorithm used to reconstruct white matter streamlines; for example, a paper [10] examines how outcomes of machine learning on connectomes change depending on tractography algorithms underlying edge reconstruction.

Regardless of a particular algorithm used to produce network edges, raw edge weights in the resulting structural connectomes are proportional to the number of detected streamlines. A researcher next makes a choice on whether to work with unweighted or weighted networks. The former approach implies that all raw weights are binarized. Given an undirected weighted graph with n nodes, let A be the $n \times n$ adjacency matrix with entries a_{ij}, where a_{ij} is the weight between the respective nodes. Unweighted graph is produced by:

$$a_{ij}^{binarized} = 1 if a_{ij} > 0,\ 0\ else. \tag{1}$$

Sometimes a threshold is set to a non-zero value to eliminate low weights. Alternatively, a threshold can be set different across participants, while the sparsity of the resulting networks is fixed across all brains (e.g., the authors of [12] compute graph metrics for the unweighted networks within a range of sparsity levels and next average the obtained values).

When a study analyses weighted brain networks, normalization of connectivity matrices is recommended [13], [14]. This is because raw number of streamlines is known to vary from individual to individual and can be affected by fiber tract length, volume of cortical regions and other factors. Normalization itself can involve geometric properties such as volumes of the cortical regions or physical path lengths between the regions (e.g., [13], [14]), or be purely based on topological properties of the networks (e.g., [15], [16]). A paper [17] examines how topological and geometric normalizations of brain networks affect the predictive quality of machine learning algorithms run on these networks. The results of [17] suggest that a combination of both topological and geometric normalizations is the most informative.

To sum up, there is certainly some ambiguity in how structural connectomes should be constructed from DTI scans. However, regardless of the particular aspects of the network reconstruction pipeline, the resulting brain graphs share some important properties. These are usually small undirected connected networks. The vertices are labelled according to brain regions, and a set of uniquely labeled vertices is the same across different connectomes constructed with the same atlas. The networks are spatially embedded: vertices are localized in 3D space, and edges have physical lengths. In what follows, we discuss how machine learning algorithms can be applied to these objects.

4 Machine learning on brain networks

Hence, a problem of classifying scans of normal and pathological brains transforms into a problem of classifying the respective brain networks. Mathematically, this is a task of pattern recognition on graphs; however, it differs from a more usual understanding of machine learning on graphs. More commonly, a graph itself becomes an object defining a metric between the vertices, and machine learning algorithms are run on vertices or neighborhoods (e.g., algorithms aiming at link prediction in social networks). Connectomics poses a different challenge: small brain graphs are now examples of classes to be distinguished by an algorithm. In this section, we provide a formal problem statement and discuss how it can be tackled.

4.1 Problem statement

Let G_i be a brain network, y_i be a class label, $y_i \in \{0, 1\}$ throughout this study. Given a training set of pairs (G_i, y_i) and the test set of input objects G_j, the task is to make a best possible prediction of the unknown class label y_j. In what follows, we use G to denote a brain graph, either unweighted or weighted, and A to denote the respective adjacency matrix which includes values from $\{0, 1\}$ if the graph is unweighted or holds edge weights if the graph is weighted. We consider the classification problem for both unweighted and weighted networks, and make special remarks on the work of the algorithms in these two cases when needed.

In some sense, this problem is similar to the problem of classifying molecules that arises in chemistry and molecular biology. A paper [18] describes some benchmark datasets from that subject area and the respective tasks, for example a task of assigning protein molecules to a class of enzymes or non-enzymes, predicting whether or not a given molecule exerts a mutagenic effect, or whether or not a given chemical compound is cancerogenic. Each molecule or compound is modeled as a graph, with the nodes representing atoms and the edges representing bonds between the atoms; each node is labeled with its atom type.

Similarly to brain networks, molecules are small connected graphs which should be assigned a class label. The major difference between the two problems is that each node in brain networks has a unique label, and hence the problem

of graph isomorphism does not arise in connectomics. This means that machine learning algorithms shown to be useful in other subject areas are to be modified to be valid for classifying brain networks; in Section 5, we show how this can be accommodated. Besides, an important prerequisite of classifying brain networks is that all brain graphs have the same number of nodes and the same set of node labels; in Section 5.5, we show how this very specific property of brain networks can be used to develop machine learning algorithms on brain graphs.

4.2 A kernel approach

The most obvious approach to machine learning within these settings would be to adopt some strategy of graph embedding and transform adjacency matrices into vectors from $\mathbb{R}^p$ because most classifiers work with this type of input objects. One could vectorize a matrix by taking the values of its upper triangle (the so-called "bag of edges") or compute some local or global graph metrics and use them as feature vectors. For an excellent example of a study within this framework, we refer to [16]. The authors work with the "bag of edges" and also compute edge betweenness centralities, network efficiency, clustering coefficients and some other topological metrics and classify different sex and kinship groups based on these features.

In this paper, we focus on a different approach that defines kernels on structured data directly and hence allows for classifying brain networks without embedding brain graphs into a real vector space. This is possible due to an important property of the SVM classifier to accept any input objects, not necessarily vectors from $\mathbb{R}^p$ [19], [20]. This means that any positive semi-definite function $K(\mathbf{x}_i, \mathbf{x}_j) : \mathbb{X}^2 \rightarrow \mathbb{R}$ on the input data $\mathbb{X}$ can be used as a kernel for the SVM classifier provided that:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j)c_i c_j \geq 0$$

for any $(x_1, x_2, \ldots, x_n) \in \mathbb{X}$ and any coefficients $(c_1, c_2, \ldots, c_n) \in \mathbb{R}$. There are no constraints on the structure of the input data $\mathbb{X}$.

In the next sections, we review several kernels that can be useful for classifying brain networks and compare their performance based on two real-life datasets.

5 Kernels on brain networks

Below we review some kernels that can be useful for a task of classifying brain networks. Recall that we only deal with structural connectomes which represent anatomical connections between brain regions. It is also possible to define functional connectomes, for which the elements of weighted adjacency matrices are the correlations between time series the respective brain regions activation; as such, certain specific kernel methods can be developed for this particular type

of input data (e.g., those that account for the geometry of the manifold of the positive definite correlation matrices [21]). In this sense, structural connectomes are more problematic as they do not lie in a specific space with known properties. In what follows, we only discuss kernel methods applicable for the analysis of structural connectomes.

We discuss two approaches to producing graph kernels. The first approach defines a kernel function that generates a positive semi-definite Gram matrix. A second approach introduces a function quantifying a distance between graphs and next obtains a kernel by exponentiating this distance.

5.1 Random walk kernel

We first consider a walk kernel described in [19], which computes the number of walks common for each pair of graphs. Since all brain graphs have the same set of uniquely labeled nodes Γ, we modify the walk kernel as described below.

The walk kernel is now computed on a graph G_* which is a minimum of G G'. The graph G_* has the same set of nodes $\Gamma_* = \Gamma$ and an adjacency matrix A_*:

$$A_* = a_{*ij} = \{\min(a_{ij}, a'_{ij}) : a_{ij} \in A, a'_{ij} \in A'\}. \tag{2}$$

Note that the equation (2) produces a correct minimum graph regardless of whether the adjacency matrix A is unweighted or holds the edge weights.

We compute the walk kernel on G and G' by:

$$K_{walk}(G, G') = \sum_{i,j=1}^{|\Gamma_*|} [\sum_{k=0}^{\infty} \mu_k A_*^k]_{ij}. \tag{3}$$

We set $\mu_k = \mu^k$. Hence, the (3) becomes:

$$K_{walk}(G, G') = \sum_{i,j=1}^{|\Gamma_*|} [\sum_{k=0}^{\infty} \mu^k A_*^k]_{ij} = \sum_{i,j=1}^{|V_*|} [(I - \mu A_*)^{-1}]_{ij} \tag{4}$$

To ensure convergence, μ must be lower than the inverse maximal eigenvalue. In this paper, we report results for μ set to 0.95 times the inverse maximal eigenvalue of A_*; lower values of μ tried in preliminary studies result in slightly worse classification quality. Conceptually, the factor μ downweights longer walks and makes short walks dominate the graph similarity score. A paper [18] discusses this effect.

In addition to sensitivity to the length of walks taken into account, walk kernel suffers from the so-called tottering effect [22]. Since walks allow for repetitions of nodes and edges, the same fragment is counted repeatedly in a graph similarity measure. In undirected graphs, a random walk may start tottering on a cycle or even between the same two nodes in the product graph, leading to an artificially high graph similarity score even when the structural similarity between two graphs is minor.

5.2 Kernel on shortest path lengths

Second, we consider a kernel on shortest path lengths. Kernels on shortest path lengths are proposed in [18] as an alternative to random walk kernel that overcomes its shortcomings discussed above. The authors [18] define a kernel on graphs that compares paths instead of walks. In this study, we only use one version of a kernel based on paths, with some preliminary modification aiming to account for unique node labels of brain networks.

For a graph G with a set of uniquely labeled nodes Γ a matrix of shortest path lengths is given by:

$$\Upsilon_{ij} = \upsilon(\gamma_i, \gamma_j), \tag{5}$$

where γ_i and γ_j are the nodes of the graph G and $\upsilon(\gamma_i, \gamma_j)$ is the length of the shortest path between these nodes (weighted or unweighted, depending on the nature of graph G).

We next define a path kernel by:

$$K_{path}(\Upsilon, \Upsilon') = \sum_{\substack{\upsilon_{ij} \in \Upsilon \\ \upsilon'_{ij} \in \Upsilon'}} K_1(\upsilon_{ij}, \upsilon'_{ij}), \tag{6}$$

where $K_1(\upsilon_{ij}, \upsilon'_{ij})$ is a kernel on pairs of paths from G and G'. For the later, we use a polynomial kernel $K_1(\mathbf{x}, \mathbf{x}')$ given by:

$$K_{poly}(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + c)^p, \quad p = 2. \tag{7}$$

For a general definition of path kernels and proof of their positive definiteness, we refer to [18].

5.3 Distance-based kernels: L_1 and L_2 norms

The above methods produce Gram matrices on graphs straightforwardly. An alternative approach is to introduce a distance between graphs and produce a kernel based on this distance measure.

Let G and G' be the networks and $\omega(G, G')$ be a distance between these networks. We build a graph kernel K using the distance ω as follows:

$$K(G, G') = e^{-\alpha \omega(G, G')} \tag{8}$$

Positive semi-definiteness of this kernel is guaranteed when ω is a metric. A paper [11] discusses kernels which are not necessarily positive semi-definite, namely those for which triangle inequality does not hold for a distance measure ω in (8). The authors claim that these kernels can always be made positive definite by an appropriate choice of the parameter α; however, forcing a kernel to be positive definite reduces its expressiveness and diminishes classification accuracy. In this study, we vary the parameter α for all distance-based kernels, including those using true metric ω.

We first define distances between networks via the L_1 and L_2 norms between the respective adjacency matrices. For two networks G and G' with $n \times n$ adjacency matrices $A = \{a_{ij}\}$ and $A' = \{a'_{ij}\}$ (either unweighted or weighted) an L_1 distance is given by:

$$\omega_{L_1}(G, G') = \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij} - a'_{ij}| \tag{9}$$

An L_2 (Frobenius) norm is defined by:

$$\omega_{L_2}(G, G') = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (a_{ij} - a'_{ij})^2} \tag{10}$$

We next produce kernels (8) based on these distance measures. This is the simplest possible way to define a distance between the adjacency matrices. In the next sections, we discuss more sophisticated procedures aiming to quantify pairwise distances between networks.

5.4 Kernels on distances between spectral distributions

Studies [23] and [24] proposed to measure similarity between brain networks based on distances between spectral distributions of the respective graphs. An idea behind spectral-based kernels is that graph eigenvalue distributions capture important information about network structure and hence might be useful for a task of classifying networks.

To construct spectral-based kernels, we use spectra of the normalized graph Laplacians. Let D be a diagonal matrix of weighted node degrees:

$$d_i = \sum_j a_{ij}. \tag{11}$$

The graph Laplacian matrix is given by:

$$L = D - A, \tag{12}$$

The normalized graph Laplacian is given by:

$$\mathcal{L} = D^{-1/2} L D^{-1/2} \tag{13}$$

Normalized Laplacians are correctly defined by (13) regardless of whether the graphs are unweighted or weighted, provided that for weighted graph the matrix A holds edge weights. The eigenvalues of the normalized Laplacians are always in range from 0 to 2. We refer to [25] for theory on the normalized Laplacian spectra and to [26] for examples of the eigenvalue distributions of the normalized Laplacians in structural brain networks of the cat, macaque and Caenorhabditis elegans.

A paper [23] defines distances between brain networks via the information-theory based measures of difference between spectral distributions. A motivation behind this approach is that we are most interested in comparing shapes of the distributions of eigenvalues rather than the vectors of eigenvalues per se. This is because multiplicity of particular eigenvalues and specific peaks in their distributions capture important information about graph structure [25].

To quantify distance between distributions, we use the Kullback-Leibler (KL) divergence and the Jensen-Shannon (JS) divergence. For two probability distributions with densities $p(x)$ and $q(x)$ the KL divergence is:

$$KL(p||q) = \int_{-\infty}^{\infty} p(x) log \frac{p(x)}{q(x)} dx \tag{14}$$

The Kullback-Leibler kernel [27] is obtained by exponentiating the symmetric KL divergence:

$$K_{KL}(p,q) = e^{-\alpha(KL(p||q)+KL(q||p))} \tag{15}$$

The JS divergence [28] is:

$$JS(p||q) = \frac{1}{2}(KL(p||r) + KL(q||r)), \tag{16}$$

where $r(x) = \frac{1}{2}(p(x) + q(x))$.

We compute Jensen-Shannon kernel by:

$$K_{JS}(p,q) = e^{-\alpha\sqrt{JS(p||q)}} \tag{17}$$

The KL and JS kernels work with the probability density functions restored from the samples. In [23], we split the entire range of eigenvalues into equal intervals (bins) and computed frequencies within each bin as a proxy for the underlying probabilities. However, the results of the entire classification pipeline were highly sensitive to the choice of the number of bins used to reconstruct density. In this study, we overcome this shortcoming by applying kernel density reconstruction prior to computation the KL and JS divergences. We use the Gaussian kernel and produce the values:

$$f(x) = \sum_{s_j} \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{|x-s_j|^2}{2\sigma^2}), \tag{18}$$

where s_j is the j-th eigenvalue of $\mathcal{L}$. To compute this, we use the statmodels [43] function for univariate kernel density estimation, which is a fast Fourier transform-based implementation that has an advantage of automatic selection of optimal bandwidth according to the Silverman's rule. We next compute the kernels (15) and (17) based on these values.

There also exists a different approach that compares spectral distributions directly based on the vectors of eigenvalues [24]. For this purpose, it uses an

earth mover's distance (EMD) [29], which measures the minimum cost of transforming one sample distribution into another. Provided that each distribution is represented by some amount of dirt, EMD is the minimum cost of moving the dirt of one distribution to produce the other. The cost is the amount of dirt moved times the distance by which it is moved.

Let $\{s_1^i, ..., s_n^i\}$ be the eigenvalues of the normalized Laplacian spectrum $\mathcal{S}_i$. We put an equal measure $1/n$ to each point s_k^i on a real line. Let f_{kl} be the flow of mass between the points s_k^i and s_l^j. The EMD is the normalized flow of mass between sets $\mathcal{S}_i = \{s_1^i, ..., s_n^i\}$ and $\mathcal{S}_j = \{s_1^j, ..., s_n^j\}$ that minimizes the overall cost:

$$emd(\mathcal{S}_i, \mathcal{S}_j) = \underset{F=\{f_{kl}\}}{\operatorname{argmin}} \frac{\sum_{k,l} f_{kl} |s_k^i - s_l^j|}{\sum_{k,l} f_{kl}}, \tag{19}$$

with the constraints: $f_{kl} \geq 0, \qquad \sum_{k=1}^{n} f_{kl} = 1/n, \qquad \sum_{l=1}^{n} f_{kl} = 1/n.$

A EMD-based kernel is next computed by (8) using (19) as a measure of distance between the respective graphs.

5.5 Kernels on distances between network partitions

The last group of graph kernels analyzed in this study quantifies similarity between brain networks based on whether or not their nodes cluster into similar communities [30]. Partition-based kernels make the full use of the uniqueness of node labels in brain networks and the identity of label sets across different brains. These kernels are based on the idea that brain networks belonging to a same class produce partitions that are more similar than those obtained for networks from different classes.

Similarly to [30], this study uses three algorithms to obtain partition of each brain network: Newman leading eigenvector method [31], Louvian method [32], and Greedy modularity optimization [33]. All these methods use modularity as a function to be optimized. *Modularity* [33] is a property of a network and a particular division of that network into communities. It measures how good is the division in the sense that whether there are many edges within communities and only a few between them. Modularity Q is given by:

$$Q = \frac{1}{2m} \sum_{ij} \left[a_{ij} - \frac{d_i d_j}{2m}\right] \delta(i, j), \tag{20}$$

where a_{ij} is an element of a graph adjacency matrix, m is a total number of edges in a given graph, d_i, d_j - degrees of nodes i and j as defined by (11).

Louvain algorithm is a two step iterative procedure. It starts with all nodes put in separate clusters. Next, for each node i and its neighbors j the algorithm computes gain in modularity that would take place after removing i from its cluster and placing it to a cluster of j; after repeating for all neighbors j, i is placed in the cluster where gain in modularity is maximal. This process repeats until there is no such node i for which its movement to another cluster produces gain in modularity. The second step of the algorithm builds a new weighted graph

wherein nodes are final clusters from the previous step and an edge between two nodes represents the sum of edges between two corresponding clusters at the previous step. Once the second step is over, the algorithm reapplies the first step and iterates.

The Newman leading eigenvector method uses normalized graph Laplacian given by (12). It starts with all nodes placed in a single cluster; different nodes next get their labels according to a sign of the respective values of a Laplacian eigenvector corresponding to the second smallest eigenvalue. The procedure repeats for each cluster till convergence. Greedy modularity optimization method is another division clustering approach which allows for fast detecting communities in large graphs or sets of many small graphs.

All these partition algorithms are defined for both unweighted and weighted graphs. We next estimate pairwise similarity of partitions of different brain networks using the adjusted Rand score (ARI). Let $U = \{U_1, U_2, \cdots U_l\}$ and $V = \{V_1, V_2, \cdots V_k\}$ be partitions of two networks G_U and G_V with the same sets of node labels, l and k be the number of clusters in the partitions U and V, respectively. To define ARI between these partitions, we construct a contingency table:

U, V	V_1	V_2	$\dots$	V_k	sum
U_1	ν_{11}	ν_{12}	$\dots$	ν_{1k}	a_1
U_2	ν_{21}	ν_{22}	$\dots$	ν_{2k}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_l	ν_{l1}	ν_{l2}	$\dots$	ν_{lk}	a_l
sum	b_1	b_2	$\dots$	b_k	

Here ν_{ij} denotes a number of objects common between U_i and V_j. ARI is then given by:

$$ARI(U,V) = \frac{\sum_{i,j}\binom{\nu_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_i}{2}\right]/\binom{\nu}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_i}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_i}{2}\right]/\binom{\nu}{2}}. \tag{21}$$

ARI takes the the value of 1 when the partitions are identical and values close to 0 in case of random labeling. We thus define a distance $\omega(G_U, G_V)$ between networks G_U and G_V by:

$$\omega(G_U, G_V) = 1 - ARI(U,V), \tag{22}$$

Hence, networks with the same partitions have zero distance, and the maximum distance is close to 1. We next produce three kernels (8) based on these pairwise distances, one for each algorithm of clustering brain networks.

6 Summary: methods

We compare performance of the kernels described in the previous section based on two tasks of classifying brain networks. In this section, we overview the classification pipeline and describe a metric used to compare performance of different kernels; in the next section, we describe the tasks and the datasets.

6.1 Classification pipeline

Figure 1 summarizes our classification pipeline. We deal with brain networks that represent different phenotypes (i.e., normal and pathological brains). We compute Gram matrices between these brain networks using the following kernels:

- Random walk (RW) kernel [19]
- Shortest path length (SPL) kernel [18]
- L_1-distance kernel
- L_2-distance kernel
- Kullback-Leibler (KL) kernel [27]
- Jensen-Shannon (JS) kernel [23]
- Earth mover's distance (EMD) kernel [24]
- Newman partition (NP) kernel [30]
- Louvain partition (LP) kernel [30]
- Greedy partition (GP) kernel [30]

We next feed these Gram matrices to an SVM classifier, train it on part of a sample and make prediction for an unseen part of a sample. In computation of distance-based kernels, we vary the values of α in the range from 0.01 to 10. The penalty parameter of the SVM classifier varies from 0.1 to 50. We report the results for models with the optimal values of α and the penalty parameter.

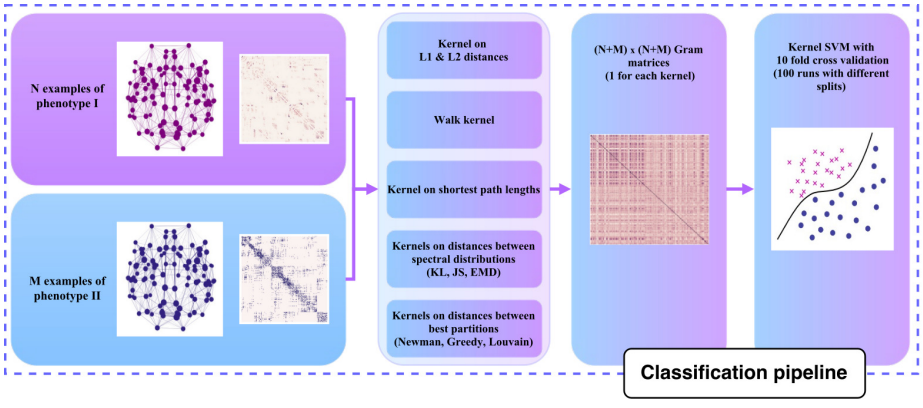

Fig. 1. Classification pipeline

6.2 Classification quality evaluation

We use the area under the receiver operating characteristic curve (ROC AUC) to assess the predictive quality of models with different kernels. We run all models with 10-fold cross-validation and combine predictions on test folds to evaluate the quality of prediction on the entire sample. For each model, we repeat this procedure 100 times with different 10-fold splits, thus producing 100 ROC AUC values.

Note that datasets under consideration are too small to divide them into three parts (train, validation and test) for parameter estimation. To deal with this, we find optimal values of the parameters based on 10 different 10-fold splits (with random states fixed for splitting) and evaluate models with optimal parameter values based on 100 other 10-fold splits. Importantly, parameter estimation for the described models is robust to the particular splitting, and it is highly unlikely that the reported validation procedure biased the results in any noticeable way.

6.3 Data analysis tools

We use Python and IPython notebooks platform[34], specifically NumPy [35], SciPy [36], pandas [37], matplotlib [38], seaborn [39], networkX [40], community [41], igraph [42], statsmodels [43], pyemd [44] and scikit-learn [45] libraries. All scripts are available at https://github.com/kurmukovai/NET2016/.

7 Data

We compare the performance of the kernels described in Section 5 based on two datasets of precomputed matrices of structural connectomes. We describe the datasets in this section and also provide some relevant information on the resulting networks.

7.1 Datasets

UCLA Autism dataset (UCLA Multimodal Connectivity Database [46], [12]) includes DTI-based connectivity matrices of 51 high-functioning autism spectrum disorder (ASD) subjects (6 females) and 43 typically developing (TD) subjects (7 females). Average age (age standard deviation) is 13.0 (2.8) for ASD group and 13.1 (2.4) for TD group. Nodes of brain networks are defined using a parcellation scheme by Power et al. [47] which is based on a meta-analysis of fMRI studies combined with whole-brain functional connectivity mapping. This approach produces 264 equal-size brain regions and thus 264×264 connectivity matrices. Network edges are produced based on deterministic tractography performed using the fiber assignment by continuous tracking (FACT) algorithm [48]; edge weights are proportional to the number of streamlines detected by FACT.

UCLA APOE-4 dataset (UCLA Multimodal Connectivity Database [46], [49]) includes DTI-based connectivity matrices of carriers and noncarriers of the APOE-4 allele associated with the higher risk of Alzheimer's disease. The sample includes 30 APOE-4 noncarriers, mean age (age standard deviation) is 63.8 (8.3), and 25 APOE-4 carriers, mean age (age standard deviation) is 60.8 (9.7). Each brain is partitioned into 110 regions using the Harvard-Oxford subcortical and cortical probabilistic atlases as implemented in FSL [50]. Therefore, this dataset includes 110×110 connectivity matrices. Network edges are obtained using the FACT algorithm [48]. Raw fiber counts in these matrices are adjusted for the unequal region volumes by scaling each edge by the mean volume of its two adjacent regions.

The authors of both datasets only report the results of statistical group comparison based on graph metrics. Hence, there is no publicly available machine learning baselines for these datasets.

7.2 Edge weights

For each classification task, we evaluate performance of all kernels on both unweighted and weighted brain networks. We produce unweighted brain networks by (1). In this case, each network contains information only on presence or absence of edges between nodes, and all edges carry equal weights.

To produce weighted brain networks, we take the original edge weights that represent streamline count between each pair of brain regions and scale them by the physical distances between the respective regions:

$$a_{ij}^{scaled} = \frac{a_{ij}}{\lambda_{ij}}, \tag{23}$$

where a_{ij} is the original weight of the edge between the nodes i and j, and λ_{ij} is the Euclidean distance between centers of the regions i and j. The distances are computed based on the standard Montreal Neurological Institute (MNI) coordinates of region centers.

To enhance between-subject comparison, we next normalize the obtained weights by:

$$a_{ij}^{normed} = \frac{a_{ij}^{scaled}}{\sum_{i,j} a_{ij}^{scaled}}. \tag{24}$$

Note that this latter scaling does not affect the kernels that are based on normalized Laplacian spectra and the partition-based kernels.

We report classification results for both weighted and unweighted connectivity matrices.

8 Results: kernel comparison

Figure 2 compares performance of the SVM classifier with different kernels in a task of classification typical development versus autism spectrum disorder.

Figure 3 provides results for a task of classifying of carriers versus non-carriers of an allele associated with an increased risk of Alzheimer's disease.

First, the results show that the classifiers run on weighted brain networks clearly outperform those run on unweighted brain graphs (the only exception is an SVM with the Newman-based partition kernel run on the UCLA APOE-4 dataset). This means that edge weights in human macroscale brain networks capture information important for classifying normal and pathological phenotypes. This is true regardless of whether we construct a kernel based on similarity of single edges or shortest paths, or random walks common between brain graphs, or distances between graph spectral distributions or partitions of brain networks. Importantly, edge weights in this study incorporate information on both strengths of connections (the number of streamlines detected by a tractography algorithms) and their lengths (approximated by Euclidean distances between the centers of brain regions).

Second, there is no kernel (or no family of kernels) that provides the best classification quality on both datasets. Random walk kernel and L_1 and L_2 distance-based kernels do not perform satisfactorily in both classification tasks.

In a task of classifying autism spectrum disorder versus typical development, kernels based on distances between spectral distributions perform the best. There is virtually no difference in behaviors of the two kernels of this type, computed with Jensen-Shannon divergence and earth mover's distance. Spectral distributions of brain networks seem to capture some information important for distinguishing this type of pathology from typical development.

For classification of carriers versus non-carriers of an APOE-4 allele, the most expressive kernels are using comparison of shortest path lengths and the distances between Louvain-based partitions of brain networks. Interestingly, the three partition-based kernels differed in their performance, which means that the analyzed partition algorithms capture different aspects of brain network structures and thus produce distances between brain networks in a different manner.

For the two best models on each dataset, we plot the ROC-curves in Figure 4. The curves are averaged over 100 repetitions of the algorithms. Interestingly, the ROC-curves do not coincide. This means that although the best-working models are close in terms of classification quality, they capture different aspects of the data and differ in terms of prediction outcome.

9 Conclusions

In this paper we considered machine learning on macroscale human brain networks. These networks (called connectomes) represent connections between brain regions reconstructed from neuroimaging data. A question is whether connectomes can be useful in discriminating between normal and pathological brain structures, which can be considered a task of classification on graphs. We point to properties of brain networks that make a task of classifying connectomes differ from a task of classifying graph objects from other subject areas.

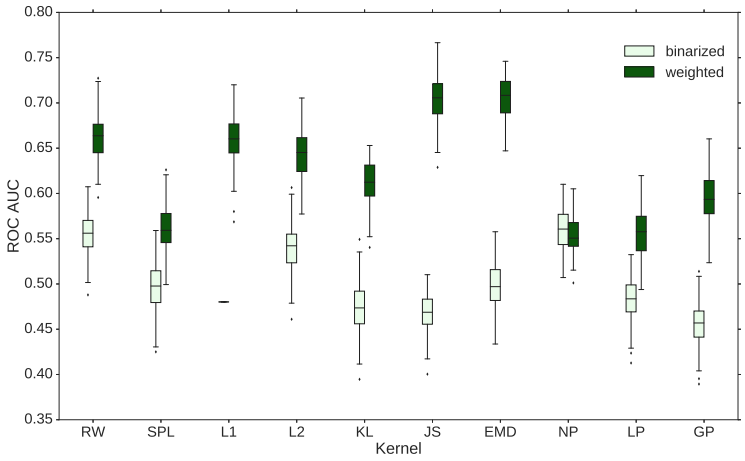

Fig. 2. Classification of typical development versus autism spectrum disorder; boxplots show ROC AUC values over 100 runs of the algorithm with different splits into train and test samples; abbreviations of the kernels are the same as in Section 6.1: RW - random walk, SPL - shortest path length, L1 - L_1-distance, L2 - L_2-distance, KL - Kullback-Leibler, JS - Jensen-Shannon, EMD - earth mover's distance, NP - Newman-based partition, LP - Louvain-based partition, GP - greedy partition.

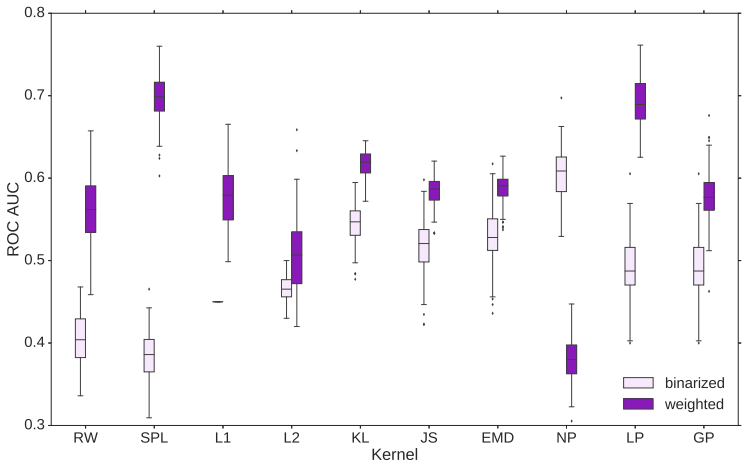

Fig. 3. Classification of carriers versus non-carriers of an allele associated with an increased risk of Alzheimer's disease; boxplots show ROC AUC values over 100 runs of the algorithm with different splits into train and test samples; abbreviations of the kernels are the same as in Section 6.1: RW - random walk, SPL - shortest path length, L1 - L_1-distance, L2 - L_2-distance, KL - Kullback-Leibler, JS - Jensen-Shannon, EMD - earth mover's distance, NP - Newman-based partition, LP - Louvain-based partition, GP - greedy partition.

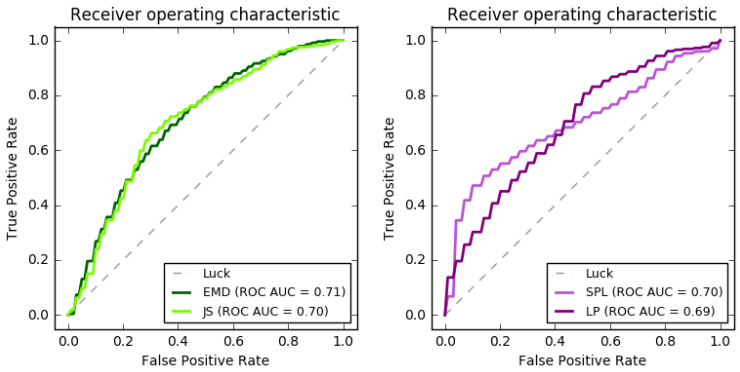

Fig. 4. ROC-curves for the best-performing models in the autism spectrum disorder (left) and APOE-4 allele carriers (right) classification tasks. Abbreviations of the kernels are the same as in Section 6.1: SPL - shortest path length, JS - Jensen-Shannon, LP - Louvain-based partition.

We next focus on a kernel classification approach and discuss several kernels that can be useful for machine learning on connectomes. We show how a random walk kernel [19] and a kernel based o shortest path lengths [18] can be modified to account for the uniqueness of node labels in brain graphs. We consider an approach that produces kernels based on distances between the adjacency matrices of the respective graphs and use L_1 and L_2 as the simplest examples of such distances. We next describe a family of kernels that are based on graph spectral distributions; of these, two kernels use measures that quantify information divergence between spectral distributions [23], and the remaining kernel is based on a distance that arises as a solution to a transportation problem. Finally, we consider a family of partition kernels [30] that quantify similarity between brain networks based on whether or not their nodes cluster into similar communities; hence, this latter approach makes the full use of the fact that brain networks share the same set of unique node labels.

We compared performance of the above kernels in two classification tasks: a task of classifying typical development versus autism spectrum disorder and a task of distinguishing carriers and non-carriers of an allele associated with an increased risk of Alzheimer's disease. We additionally questioned whether brain networks with edge weights carrying information on strengths and lengths of the respective connections are more informative for these classification tasks than unweighted brain networks which only model the presence of connections.

The answer to this latter question was quite clear: the classifiers run on weighted brain networks outperformed those run on unweighted brain graphs in both tasks, regardless of the particular kernel function. Edge weights should not be ignored in classification of human macroscale brain networks.

The best-performing kernels were task-specific. In a task of classifying autism spectrum disorder versus typical development, spectral distributions of brain

networks seem to carry information useful for distinguishing between these two classes; the two best-performing models quantified distances between networks based on similarity in their spectral distributions. However, these kernels did not perform well in classification of carriers and non-carriers of an allele associated with an increased risk of Alzheimer's disease. In this latter task, the kernels based on shortest path lengths and the similarity in partitions of brain networks were the most expressive.

The kernels analyzed in this study seem to capture different aspects of network structures specific for normal and pathological brains. Future studies may aim at aggregating information stemming from different kernel models in order to improve the quality of machine learning on brain networks.

Acknowledgements

The study was supported within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2016 (grant #16-05-0050) and by the Russian Academic Excellence Project "5-100".

References

1. Craddock, R.C., Jbabdi, S., Yan, C.G., Vogelstein, J.T. (2013) Imaging human connectomes at the macroscale. Nature Methods, 10, 6, 524–539.
2. Haller, S., Lovblad, K.-O., Giannakopoulos, P., Van De Ville, D. (2014) Multivariate pattern recognition for diagnosis and prognosis in clinical neuroimaging: state of the art, current challenges and future trends. Brain topography, 27, 3, 329–337.
3. Wolfers, T., Buitelaar, J.K., Beckmann, C.F., Franke, B., Marquand, A.F. (2015) From estimating activation locality to predicting disorder: A review of pattern recognition for neuroimaging-based psychiatric diagnostics. Neurosci Biobehav Rev., 57, 328–349.
4. Arbabshirani, M.R., Plis, S., Sui, J., Calhoun, V.D. (2016) Single subject prediction of brain disorders in neuroimaging: Promises and pitfalls. NeuroImage, in press.
5. Hagmann, P. (2005). From diffusion MRI to brain connectomics (Thesis). Lausanne: EPFL.
6. Sporns, O., Tononi, G., Ktter, R. (2005). The Human Connectome: A Structural Description of the Human Brain. PLoS Computational Biology, 1, 4, e42.
7. Smith, S.M., Miller, K.L., Salimi-Khorshidi, G., Webster, M. et al. (2011) Network modelling methods for FMRI. NeuroImage, 54(2), 875–891.
8. Fornito, A., Zalesky, A., Breakspear, M. (2013) Graph analysis of the human connectome: Promise, progress, and pitfalls. Neuroimage, 15, 80, 426–444.
9. Zalesky, A., Fornito, A., Harding, I.H., Cocchi, L., Ycel, M., Pantelis, C., Bullmore, E.T. (2010) Neuroimage, 15, 50, 3, 970–983.
10. Zhan, L., Zhou, J., Wang, Y., et al. (2015) Comparison of nine tractography algorithms for detecting abnormal structural brain networks in Alzheimer's disease. Front Aging Neurosci, 14, 7, 48.
11. Chan, A.B., Vasconcelos, N., Moreno, P.J. (2004) A family of probabilistic kernels based on information divergence. Univ. California, San Diego, CA, Tech. Rep. SVCL-TR-2004-1.

12. Rudie, J.D., Brown, J.A., Beck-Pancer, D., Hernandez, L.M., Dennis, E.L., Thompson, P.M., et al. Altered functional and structural brain network organization in autism. Neuroimage Clin 2, 79–94 (2013)
13. Bassett, D.S., Brown, J.A., Deshpande, V., Carlson, J.M., Grafton, S., Conserved and variable architecture of human white matter connectivity. Neuroimage 54, 2, 12621279 (2011)
14. Hagmann, P., Kurant, M., Gigandet, X., Thiran, P., Wedeen, V.J., Meuli, R., Thiran, J.-T. Mapping human whole-brain structural networks with diffusion MRI. PLoS One 2, 7, e597 (2007)
15. Gong, G., Rosa-Neto, P., Carbonell, F., Chen, Z.J., He, Y., Evans, A.C. Age- and gender-related differences in the cortical anatomical network. J. Neurosci. 29, 50, 1568415693 (2009)
16. Duarte-Carvajalino, J.M., Jahanshad, N., Lenglet, C., McMahon, K.L., de Zubicaray, G.I., Martin, N.G., Wright, M.J., Thompson, P.M., Sapiro, G.Hierarchical topological network analysis of anatomical human brain connectivity and differences related to sex and kinship. Neuroimage 59, 4, 37843804 (2012)
17. Petrov D., Dodonova Y., Zhukov L. E., Belyaev M. Boosting connectome classification via combination of geometric and topological normalizations, in: IEEE 6th International Workshop on Pattern Recognition in Neuroimaging (PRNI), 2016. http://dx.doi.org/10.1109/PRNI.2016.7552353
18. Borgwardt K.M. Graph kernels. Dissertation (2007)
19. Gartner T. A survey of kernels for structured data. SIGKDD Explorations, 5, 1, 49–58 (2003)
20. Kashima, H., Tsuda, K., and Inokuchi, A. Marginalized kernels between labeled graphs. In Proc. Intl. Conf. Machine Learning, 321–328.
21. Dodero L., Minh H.Q., Biagio M.S., Murino V., Sona D. Kernel-based classification for brain connectivity graphs on the Riemannian manifold of positive definite matrices. Proc. of the International Symposium on Biomedical Imaging, 42-45.
22. Mah e, P., Ueda, N., Akutsu, T., Perret, J.-L., and Vert, J.-P. (2004). Extensions of marginalized graph kernels. In Proceedings of the Twenty-First International Conference on Machine Learning, 552–559.
23. Dodonova Y., Korolev S., Tkachev A., Petrov D., Zhukov L. E., Belyaev M. Classification of structural brain networks based on information divergence of graph spectra, in: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), 2016. http://dx.doi.org/10.1109/MLSP.2016.7738852
24. Dodonova Y., Belyaev M., Tkachev A., Petrov D., Zhukov L. E. Kernel classification of connectomes based on earth mover's distance between graph spectra, in: 2016 1st Workshop on Brain Analysis using COnnectivity Networks (BACON MICCAI), 2016. https://arxiv.org/abs/1611.08812
25. Chung F. (1997) Spectral Graph Theory.
26. de Lange S.C., de Reus M.A., van den Heuvel M.P. (2014) The Laplacian spectrum of neural networks. Frontiers in Computational Neuroscience, 1–12.
27. Moreno, P. J., Ho, P.,Vasconcelos, N. (2003) A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. Advances in Neural Information Processing Systems.
28. Lin, J. (1991) Divergence measures based on Shannon entropy. IEEE Transactions on Information Theory, 37, 14, 145 – 151.
29. Rubner, Y. , Tomasi, C., Guibas, L. J.: The earth movers distance as a metric for image retrieval, International Journal of Computer Vision, 40 (2000)

30. Kurmukov, A., Dodonova, Y., Zhukov, L. Classification of normal and pathological brain networks based on similarity in graph partitions, in: The Sixth IEEE ICDM Workshop on Data Mining in Networks. IEEE Computer Society (to appear).
31. Newman, M. E. J. (2006) Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E, 74, 036104.
32. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, R. (2008) Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment, 10, P10008.
33. Clauset, A., Newman, M. E. J., Moore, C. (2004) Finding community structure in very large networks. Phys Rev E, 70, 066111 .
34. Pérez, F., Granger, B. E. (2007) IPython: A System for Interactive Scientific Computing. Computing in Science & Engineering, 9, 21–29.
35. van der Walt, S., Colbert, S. C., Varoquaux, G. (2011) The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13, 22–30
36. Jones, E., Oliphant, E., Peterson, P., et al. (2001) SciPy: Open Source Scientific Tools for Python, http://www.scipy.org/ [Online; accessed 2016-06-03].
37. McKinney, W. (2010) Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.
38. Hunter, J. D. Matplotlib: A 2D graphics environment. Computing In Science Engineering 9, 3. 90–95 (2007)
39. Seaborn: v0.5.0. DOI 10.5281/zenodo.12710
40. Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. Proc. of the 7th Python in Science Conference, 11–15 (2008).
41. Available at: http://perso.crans.org/aynaud/communities/api.html
42. Csardi G., Nepusz T. The igraph software package for complex network research. InterJournal, Complex Systems 1695. (2006) Available at: http://igraph.org/python/
43. Seabold, S., and Perktold, J. Statsmodels: Econometric and statistical modeling with python. Proc. of the 9th Python in Science Conference. (2010)
44. Available at: https://github.com/garydoranjr/pyemd
45. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É. (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
46. Brown, J.A., Rudie, J.D., Bandrowski, A., Van Horn, J.D., Bookheimer, S.Y. (2012) The UCLA multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. Frontiers in Neuroinformatics 6, 28.
47. Power, J.D., Cohen, A.L., Nelson, S.M., Wig, G.S., Barnes, K.A., Church, J.A., Vogel, A.C., Laumann, T.O., Miezin, F.M., Schlaggar, B.L., Petersen, S.E. (2011) Functional net- work organization of the human brain. Neuron 72, 665–678.
48. Mori, S., Crain, B.J., Chacko, V.P., Van Zijl, P.C. (1999) Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. Annals of Neurology, 45, 265–269.
49. Brown, J.A., Terashima, K.H., Burggren, A.C., et al. (2011) Brain network local interconnectivity loss in aging APOE-4 allele carriers, PNAS, 108, 51, 20760–20765.
50. Jenkinson, M., Beckmann, C.F., Behrens, T.E., Woolrich, M.W., Smith, S.M. (2012) FSL. NeuroImage, 62, 782–790.