Differences in Structural Connectomes between Typically Developing and Autism Groups

Dmitry Petrov[1,2], Yulia Dodonova[2], and Leonid Zhukov[2]

[1] Institute for Information Transmission Problems
Bolshoy Karetny 19-1, Moscow, Russia
to.dmitry.petrov@gmail.com,
[2] National Research University Higher School of Economics,
Faculty of Computer Science, Kochnovsky 3, Moscow. Russia

Abstract. We study differences in structural connectomes between typically developing and autism spectrum disorders individuals with machine learning techniques using connection weights and network metrics as features. We build linear SVM classifier with accuracy score 0.64 and report 16 features (seven connection weights and nine network node centralities) best distinguishing these two groups.

Keywords: brain networks, graph theory, machine learning, autism spectrum disorders

1 Introduction

Connectome graphs are discrete mathematical models which represent structural or functional connections between anatomically distinct brain areas [1]. One of the key goals of modern neuroscience is to link inter-individual connectome differences and phenotypes. For example, psychiatric disorders have been found to be associated with variation in connectomes [2], [3]. This is an important area of studies because it not only provides insights into the nature of neurological disorders but also has valuable practical applications in medical diagnostics.

To date diagnostic of psychiatric disorders based on neuroimaging data is far from being accurate (for a recent review on this topic, see [4]). This is true for many psychiatric conditions, including autism spectrum disorders considered in our study. Most of the studies reported to date are based on relatively small samples and mostly incorporate the logics of group-based comparison without any cross-validation procedures (for review of findings specific to autism spectrum disorders, see [5], [6]).

In our study we use machine learning algorithms with cross-validation procedures to investigate structural differences between typically developing (TD) and autism spectrum disorder (ASD) subjects. In each step we build our models on train datasets and then validate them on unknown test data. This provides an insight of how well our models will behave on newly coming observations, while predictive power of the features based on whole-group analysis remains unclear [7].

Details on the dataset we used are provided in the next section. It was analysed by Rudie et al. [8], who mostly focused on functional connectivity matrices and also tried to jointly analyse structural and functional networks. Their major conclusions concerned associations between metrics obtained based on structural and functional networks, which tended to differ in the ASD and TD groups.

Our analysis differs from the original study in several key aspects. First, we only consider DTI-based structural networks and use an extensive list of local and global graph metrics as potential predictors of group membership, while the authors of the original study only analysed 6 global graph metrics. Second, we normalize original connectivity matrices prior to any analysis, a step that seems to be omitted in the original study. This gives us a possibility to compare topological properties of brain networks and eliminate differences due to physical length of fibers and volumes of cortical regions. Finally, as mentioned above, we use cross-validation techniques to test predictive power of observed structural differences. Note that the original study did not involve machine learning techniques and only reported group differences in graph metrics; hence, our results are not directly comparable to those obtained by Rudie et al.[8].

2 Methods

2.1 Dataset

We use UCLA autism dataset publicly available for download at the UCLA Multimodal Connectivity Database [9], [10]. The dataset includes DTI-based connectivity matrices of 51 ASD subjects (6 females) and 43 TD subjects (7 females). Average age (age standard deviation) were 13.0 (2.8) for ASD group and 13.1 (2.4) for TD group. To control for possible confounding effects, we included both age and sex as features in all analyses. Details on participants recruitment, DTI scans acquisition and construction of connectivity matrices can be found in the paper by Rudie et al. [8]. In this section, we only focus on some key aspects of the pipeline.

DTI scans were acquired on a Siemens 3T Trio. The DTI sequence consisted of 32 scans with different diffusion-weighted directions (b=1000 s/mm^2), three scans with no diffusion sensitization at b=0, and six scans at b=50 s/mm^2. An in-plane voxel dimension was 2×2mm with 2-mm thick axial slices, and total scan time was 8 min 1 s. Subjects with excessive motion artifacts were not included in the final sample. Mean and maximum relative motion did not differ in ASD and TD groups. Motion and eddy current correction was performed on the diffusion-weighted images using "eddy correct" in FMRIB's Diffusion Toolbox.

Whole brain deterministic tractography was performed on voxelwise fractional anisotropy (FA) values using the fiber assignment by continuous tracking (FACT) algorithm [11] in Diffusion Toolkit [12]. Tractography was carried out with relaxed constraints: maximum turn angle was set at 50^o, and no FA cutoff was applied. This means that the algorithm implied somewhat boosted likelihood of detecting longer fibers between spatially distant areas. Fibers were smoothed

using a spline filter; fibers shorter than 5 mm were excluded from connectivity count.

As mentioned in the previous section, the choice of brain volume parcelling scheme is an important step in connectivity matrix construction. It determines the number and location of the vertices of brain networks and thus the structure of the graph to be analysed. For this dataset, definition of nodes was somewhat unusual for DTI-based networks that commonly use atlas-based or voxel-wise parcellation approaches. Instead, connectivity matrices in this dataset were created using parcellation scheme recently proposed by Power et al. [13] based on a large meta-analysis of fMRI studies combined with whole brain functional connectivity mapping. This approach produced 264 brain regions and thus 264×264 connectivity matrices. For the purposes of this study, we take this parcellation scheme as is and do not discuss its potential benefits and caveats.

The number of streamlines connecting each pair of regions was used to set the respective edge weights. Thus, the resulting adjacency matrices were symmetric and weighted, with larger weights indicating more streamlines detected between the respective brain regions. Following recommendations by Jones et al. [14] we prefer not to use the term 'fiber count' because the number of streamlines detected by tractography algorithm does not necessarily correspond to the number of actual white matter fibers.

2.2 Normalization

The original dataset includes non-normalized DTI connectivity matrices. Neither [8] nor [9] report any normalization procedures applied to them. But number of detected streamlines is known to vary from individual to individual and can also be affected by fiber tract length, volume of cortical regions and other factors. Normalization of connectivity matrices is highly recommended prior to any analyses (e.g., see [15] and [16]).

There is no consensus on how to normalize the streamline count. There seems to be two major approaches to it. The first approach directly involves geometric measures such as volume of the cortical regions or physical path lengths between the regions [15], [16]. The second requires purely topological normalizations (e.g., see [17] and [18]). We used the latter approach in this study.

Topological normalizations themselves can differ in what effects they aim to eliminate. In the simplest case, streamline count for each pair of regions is normalized by the total number of streamlines in the entire brain, thus reducing variability among the connectivity matrices due to differences in the total number of detected streamlines. More sophisticated procedures involve weighting each edge by the arithmetic mean or geometric mean of the total number of streamlines leaving its adjacent regions. Yet another approach aims to interpret weights as probabilities of coming from one region to another and thus produces non-symmetric matrices as a result of normalization.

For the data at hand, different topological normalizations did not result in any meaningful differences in the outcomes. We only report results obtained

based on the simplest topological normalization:

$$w_{ij} = \frac{a_{ij}}{\sum_{ij} a_{ij}}. \tag{1}$$

In addition, this normalization is modified by dividing each normalized matrix by its maximum value, as recommended by [18]. This further reduces differences between different connectivity matrices and allows comparison based on purely topological characteristics.

2.3 Network features

Here we describe network features generated to implement supervised machine learning techniques. Here and forth V is the set of all nodes in our network G and $n = |V|$ the number of nodes. E is the set of all edges in the network, and $m = |E|$ is number of them. Edges (i, j) have weights w_{ij}, which are normalized $(0 \leq w_{ij} \leq 1)$. a_{ij} is the connection status between nodes i and j. $A^W = w_{ij}$ and $A = a_{ij}$ are weighted and unweighted adjacency matrices.

Bag of edges. The simplest method to produce features is to treat matrix as a vector. Each weighted edge acts as a feature, and no relationships between them are taken into account. For 264×264 connectivity matrices this method produces 34,716 features (because DTI connectivity matrices are symmetric with zero diagonal).

Node-level local metrics. To capture properties of overall network structure, we also compute local node-based and global graph metrics. We use weighted metrics whenever possible and unweighted when there is no ready-made solution (see Programming tools section). For each metric, we also compute mean across all nodes, standard deviation, interquartile range and percentiles from zero to 100-th with step 10, thus producing 14 summary statistics of metric distribution, which we also use as features for machine learning. For a discussion of possible metrics in brain connectivity analysis we recommend [19].

Weighted degree

$$k_i^W = \sum_{j \in V} w_{ij}, \tag{2}$$

Average weighted neighbour degree

$$k_{nn,i}^W = \frac{\sum_{j \in V} w_{ij} k_j^W}{k_i^W}. \tag{3}$$

Closeness centrality. Inverse of average weighted distance to other nodes [20]

$$L_i^{-1} = \frac{n-1}{\sum_{j \in V, j \neq i} d_{ij}^W}, \tag{4}$$

where d_{ij}^W is weighted shortest path length between nodes i and j.

Betweenness centrality. Quantifies the number of times a node acts as a bridge along the shortest path between two other nodes [20]. We use the weighted version with shortest paths being computed for the weighted graph

$$b_i = \frac{2}{(n-1)(n-2)} \sum_{\substack{h,j \in V \\ h \neq j, h \neq i, j \neq i}} \frac{\rho_{hj}(i)}{\rho_{hj}}, \tag{5}$$

where ρ_{hj} is the number of weighted shortest paths between h and j, and $\rho_{hj}(i)$ is the number of weighted shortest paths between h and j that pass through i.

Eigenvector centrality. Gives high values to vertices that are connected to many other well-connected vertices [21]

$$ec_i = v_i, \tag{6}$$

where v is eigenvector, corresponding to the largest eigenvalue of A^W.

Weighted clustering coefficient. Represents how nodes in a graph tend to cluster together [22]

$$c_i = \frac{1}{k(i)(k(i)-1)} \sum_{j,k} (w_{ij} w_{ik} w_{jk})^{1/3}, \tag{7}$$

where k_i is unweighted degree of node i.

Number of triangles around node

$$t_i = \frac{1}{2} \sum_{j,k \in V} a_{ij} a_{ik} a_{jk}. \tag{8}$$

Global graph metrics. We also use the following global metrics to produce features describing our networks. They are all unweighted because there were no consistent ready-made solutions for their weighted version.

Network density. Quantifies how the number of edges differs from that in a complete graph [22]

$$D = \frac{2|E|}{|V|(|V|-1)}. \tag{9}$$

Degree assortativity coefficient. Pearson correlation coefficient of unweighted degree between pairs of connected nodes [23]

$$r = \frac{|E|^{-1} \sum_{(i,j) \in E} k_i k_j - \left[|E|^{-1} \sum_{(i,j) \in E} \frac{1}{2}(k_i + k_j) \right]^2}{|E|^{-1} \sum_{(i,j) \in E} \frac{1}{2}(k_i^2 + k_j^2) - \left[|E|^{-1} \sum_{(i,j) \in E} \frac{1}{2}(k_i + k_j) \right]^2}. \tag{10}$$

Maximal clique size. Number of vertices in maximal complete subgraph of network.

Transitivity of the graph [22]

$$T = \frac{\sum_{i \in E} 2t_i}{\sum_{i \in E} k_i(k_i - 1)}. \tag{11}$$

Diameter and radius of the graph

$$d = \max \max_{i,j \in V} d_{ij}^{min}, \; r = \min \max_{i,j \in V} d_{ij}^{min}, \tag{12}$$

where d_{ij}^{min} is the shortest unweighted path length between nodes i and j.

2.4 Machine learning and cross-validation

We consider the two classes "typically developing" (TD) and "autism spectrum disorder" (ASD) based on diagnosis, and classify individual brain networks based on constructed features. We consider a combined set of edgewise and graph-based metrics which give us (together with age and sex) a 94×36,670 matrix of features. We also treat edgewise and graph-based sets of features separately, dealing with 94×34,718 and 94×1,954 matrices of features, respectively.

Constructed feature vectors have very high dimensionality which can lead to situation when some of the features are correlated with diagnosis simply by chance [7]. We deal with this problem using k-fold cross-validation and train and test our classifier on different data. We use 10-fold cross-validation because it provides a good trade-off between robustness to unobserved data and using as much data as possible to train the classifier [24].

To reduce dimensionality of the data we use univariate feature selection based on χ^2 statistic between each feature and the diagnosis. To avoid model overfitting, we perform feature selection on each training set within cross-validation separately, and obtain quality-of-fit statistics for unobserved test data. We repeat this algorithm 100 times for each number of selected features in range from 2 to 40 and report mean accuracy and mean area under the ROC-curve for 100 runs. For the number of selected features that we consider optimal, we plot ROC-curves based on 1,000 runs of 10-fold cross-validation procedure.

With this approach, different sets of features can be selected within each fold. Although sometimes it boosts accuracy score, it also limits ones ability to interpret the results. To deal with it, we use a hierarchical algorithm of cross-validation, with leave-one-out procedure as an external cross-validation. In each step, one observation is left aside, and 10-fold cross-validation with feature selection is run on the remaining 93 training observations. To fit classifier and predict 94th test observation, we use features which are selected above 0.9 threshold (in 9 or 10 folds). Again, we repeat this procedure 100 times. We report accuracy of classification, precision and recall scores for each run and also extract features that survived internal cross-validation threshold.

For all algorithms we use Support Vector Machine (SVM) [25] classifier as it exploits high discrimination possibility in high dimensional spaces. We report results for SVM with linear kernel.

2.5 Programming tools

We perform coding in Python 3.4.3 (2.7.9 for second author) and IPython command shell 3.0.0 [26]. We make calculations with matrices, computation of graph metrics and numerical analysis in NumPy 1.9.2 [27] and NetworkX 1.9.1 [28]. We implement machine learning algorithms in scikit-learn 0.15.2 [29]. Figures are plotted in matplotlib 1.4.3 [30].

3 Results

3.1 Cross-validation with feature selection

Fig. 1 shows the results of 100 runs of 10-fold cross-validation procedure with different number of features set to be selected on train subsets and validated on test subsets. Accuracy scores and values of the area under the ROC-curve are plotted on the same graph (recall, however, that due to different class sizes our baseline chance accuracy was about 0.54).

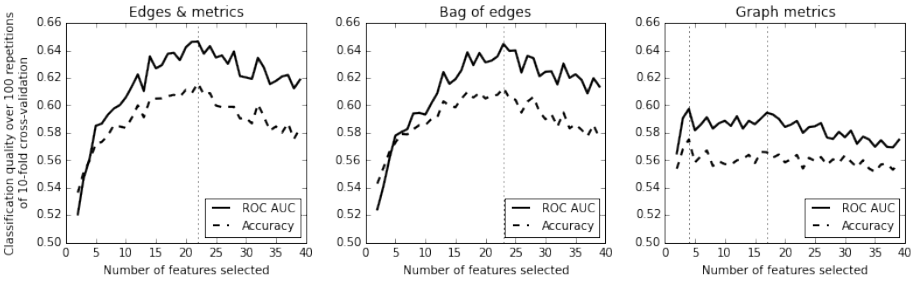

Fig. 1. Results of feature selection within 10-fold cross-validation. From left to right: combined matrix of features (edges and graph metrics), only edges used as features, only graph metrics used as features

Plots for the combined matrix (edges and network metrics) are very similar to those obtained for edge-based features only. Overall quality of prediction is higher for the bag of edges model (and also the combined model) than for the network metrics.

We obtain highest accuracy scores for different numbers of selected features in case of edgewise and network metrics feature matrices. For the bag of edges model and the combined matrix of features it is 23 and 22 features respectively,

and for local and global graph metrics two maximums of the observed curves correspond to 4 and 17 selected features.

To output the results, we run the algorithm 1,000 times with these numbers of features selected within cross validation. The resulting mean ROC-curves (based on interpolation of the original ROC-curves) are plotted in Fig. 2. In Fig. 3, we also show boxplots of ROC AUC values obtained across 1,000 runs of the algorithm (averaged across 10 folds within each run).

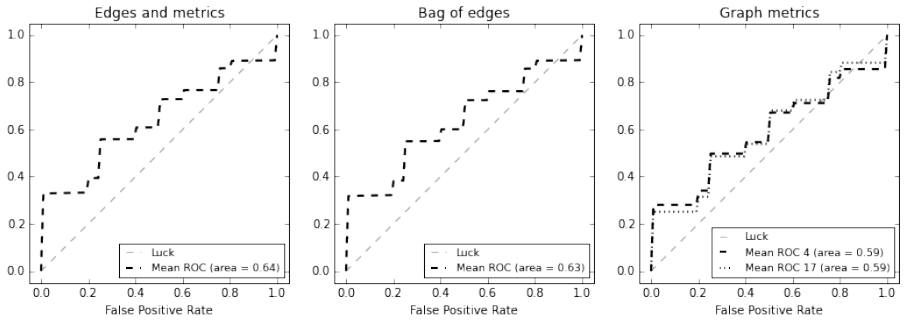

Fig. 2. ROC-curves interpolated based on 1,000 runs of the cross-validation procedure. From left to right: combined matrices (22 features selected), bag of edges (23 features selected), and graph-based metrics (4 and 17 features selected, see the legend)

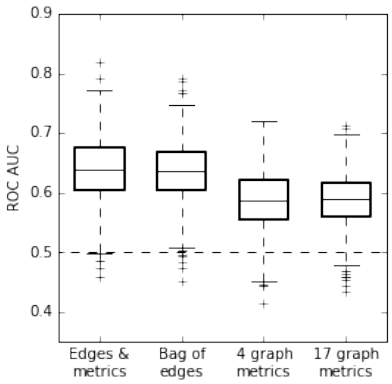

Fig. 3. Boxplots of the mean ROC AUC values across 1,000 runs of the algorithm

Quality of prediction is higher for models that include edge-based features than for those based on network metrics only. There is no difference between the two graph-metrics-based models. Regardless of the number of network metric

features selected, quality of prediction is only slightly better than that of a trivial classifier.

3.2 Robust feature selection and external leave-one-out cross-validation

A disadvantage of the procedure described above is that different sets of features can be selected in each step thus rendering the interpretation of results challenging. To overcome it, we select features within 10-fold cross-validation algorithm, choose features that appear in 9 or 10 folds and validate this newly formed set of features within an external leave-one-out procedure. For the internal cross-validation, the number of features to be initially selected was set at 50. Different numbers of features could survive our frequency-of-being-selected filter. For each of 94 leave-one-out validations, we output the sets of features selected for validation in addition to the quality of classification scores. We repeated the entire procedure 100 times.

Fig.4 shows overall accuracy, precision and recall of the models obtained in 100 runs based on the combined matrix of features and on edge-based and graph-based features.

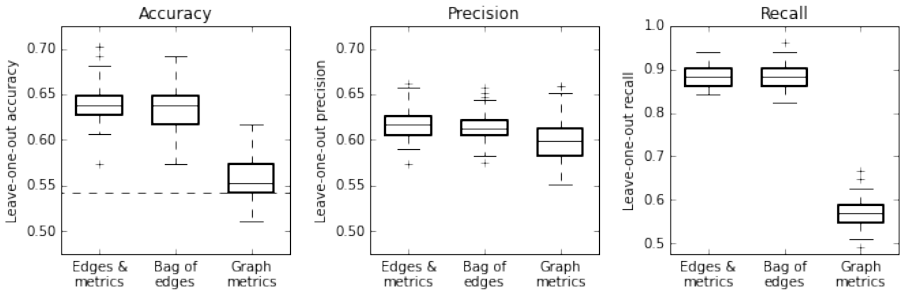

Fig. 4. Leave-one-out accuracy, precision and recall values (boxplots represent results obtained in 100 runs of the algorithm. Dashed line shows accuracy that would be obtained by a trivial classifier, which assigns all observations to ASD class

Note that for edge-based features and combined feature matrices moderate precision scores go together with relatively high recall scores. It means that our model is correct in detecting true positives (ASD individuals), but tends to produce false positives lowering the precision score. As we repeat the leave-one-out validation procedure 100 times, we get 9400 of feature sets selected by the algorithm (recall that randomness in our leave-one-out procedure comes from the 10-fold cross-validation run on the bottom-level to select features). In Fig. 5, we output frequencies of features to be selected and the numbers of features selected with the respective frequencies.

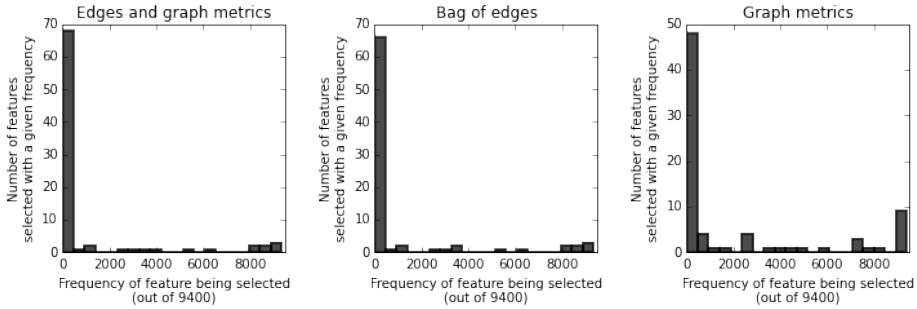

Fig. 5. Frequencies of features selected in 100 runs of 10-fold cross-validation inside leave-one-out procedure

Bag-of-edges features produce the model with the highest accuracy, although different sets of features are selected for each train sample in cross-validation. Only seven edges occur in more than 80 percent of train sets. Their respective adjacent brain regions are listed in Table 1. Boxplots for these weighted edges for ASD and TD groups are compared in Fig. 6 (left). For graph-based metrics, nine features are selected in almost all train sets. They are listed in Table 1, and the respective boxplots are shown in Fig. 6 (right). In case of combined edges and metrics matrix χ^2 feature selection always prefer edges to metrics, and exactly the same edges are always selected as in the bag of edges model.

Table 1. Selected features and edges

Number	Features and brain regions
1	Edge: Left Cingulate Gyrus anterior division, Right Frontal Pole
2	Edge: Right Pallidum, Right Parahippocampal Gyrus posterior division
3	Edge: Right Precentral Gyrus, Left Thalamus
4	Edge: Right Lateral Occipital Cortex superior division, Left Hippocampus
5	Edge: Right Precuneous Cortex, Right Parahippocampal Gyrus posterior division
6	Edge: Left Precuneous Cortex, Left Central Opercular Cortex
7	Edge: Left Precuneous Cortex, Right Occipital Fusiform Gyrus
8	Degree: Left Postcentral Gyrus
9	Betweenness centrality: Right Frontal Pole
10	Betweenness centrality: Right Frontal Orbital Cortex
11	Betweenness centrality: Right Insular Cortex
12	Betweenness centrality: Right Precentral Gyrus
13	Eigenvector centrality: Right Precuneous Cortex
14	Eigenvector centrality: Left Frontal Operculum Cortex
15	Eigenvector centrality: Left Frontal Pole
16	Number of triangles: Right Frontal Pole

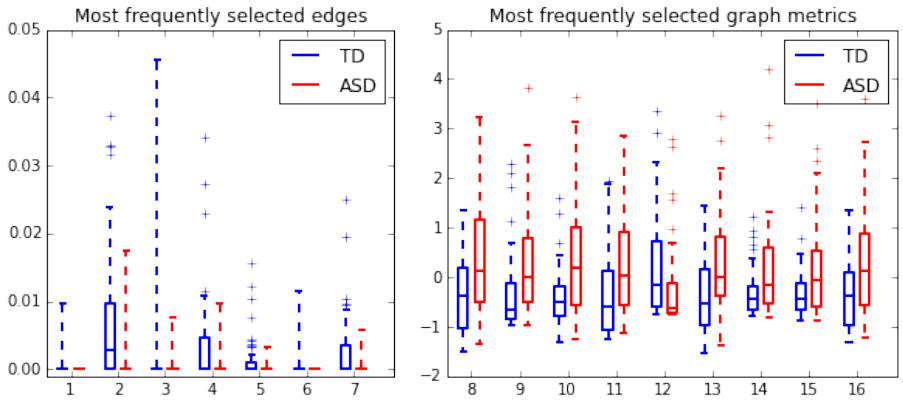

Fig. 6. Boxplots of the edges (left) and network metrics (right) which are most frequently chosen by cross-validation techniques. The labels of features correspond to those explained in Table 1. Network metrics are standardized

Finally, we plot 2D-projections of some best-discriminating features, with the respective separating hyperplanes obtained by SVM fitted on the entire dataset and only two features considered. These are shown in Fig. 7 (for edges) and Fig. 8 (for network metrics).

4 Discussion

The best obtained classifier is based solely on network connection weights and has 0.64 accuracy score. It tends to overestimate the rate of positive diagnoses thus producing high recall scores and somewhat lower precision. Models based on network metrics and node centralities turned out to be only slightly better than a trivial classifier.

Network features that contribute most to discrimination between the two groups are seven edges and nine local node metrics (in the metrics-based model only). Edge-based classification reveals presence of considered edges in TD group and their absence in ASD group. Particular edge weights did not seem to be important for the bag of edges model. Of nine network node metrics selected, four were betweenness centralities and three were eigenvector centralities for different nodes. The remaining two metrics characterized weighted degree of a node and the number of triangles adjacent to another node.

There are several limitations of this study. First of all, sample size is quite small for the results to be conclusive. Although the groups of ASD and TD subjects are relatively large compared to similar studies published to date, it is highly desirable to replicate the analyses on larger samples. This is primarily due to high dimensionality of the task at hand. For example, analysis based on the bag of edges involved tens of thousands of features with only 94 observations.

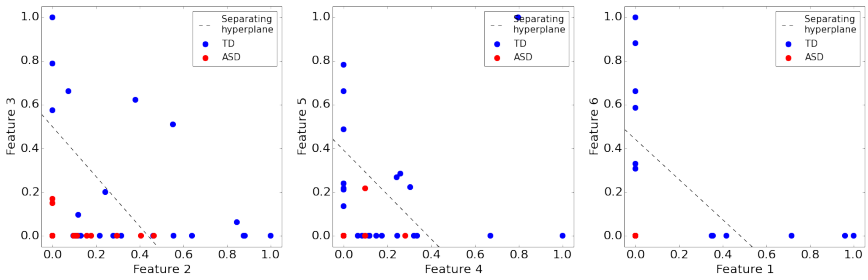

Fig. 7. Scatter plots of edges best discriminating between ASD and TD individuals. Separating hyperplane for each plot was fitted on the entire sample with only two features considered. The labels of features correspond to those explained in Table 1. Note that in some instances red dots overlay blue ones (this is the case in the plot on the right where the two subgroups actually cannot be perfectly separated)

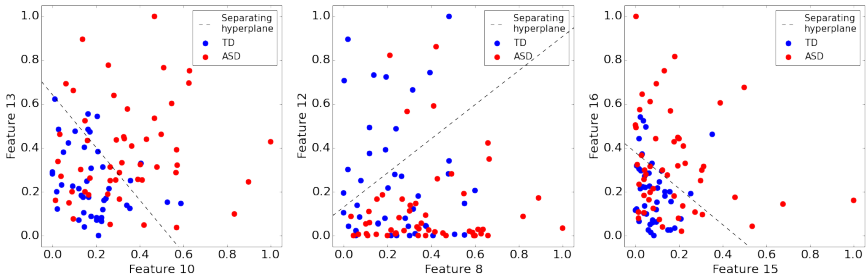

Fig. 8. Scatter plots of network metrics best discriminating between ASD and TD individuals. Separating hyperplane for each plot was fitted on the entire sample with only two features considered. The labels of features correspond to those explained in Table 1.

We employed statistical techniques of feature selection and cross-validation recommended for such situations, but larger sample size would be the best recipe to improve the analysis.

Second, there could be other approaches to dimensionality reduction and classification itself. Here, we only report results for χ^2-based feature selection which fails to account for any relations between the features. Lasso regularization is another alternative to be considered. Even more importantly, comparing results obtained by different classifiers is another work to be done in the future.

Third, there are certain methodological aspects of the data that should be noted. For example, parcelling brain volume into nodes was quite unusual for structural network analysis. DTI-based networks are commonly constructed on atlas-based zones or their partitions. Thus, results obtained for this functional-connectivity-based parcellation scheme need to be reproduced on networks with alternative parcellation of brain zones.

Finally, we intentionally left aside any neurological interpretation of our findings. Our study is purely exploratory in nature, and our analysis is blind to substantial meaning of the observed differences. We output the labels of brain zones for which significant differences in local network characteristics between ASD and TD groups are found, but do not go any further in interpreting our results. At this point, we also do not attempt to compare our findings with other group differences reported in literature on brain substrates of autism. This is another important work to be done in the future, especially if similar findings survive replication on larger samples.

Acknowledgements

First author (D.P.) was financially supported by the Russian Science Foundation (Grant No.14-50-00150). We gratefully acknowledge UCLA Multimodal Connectivity Database for making the dataset available to the research community. We also thank Mikhail Belyaev (Institute for Information Transmission Problems) for his insightful comments on machine learning techniques.

References

1. Craddock, R.C., Jbabdi, S., Yan, C.G., Vogelstein, J.T. Imaging human connectomes at the macroscale. Nature Methods 10, 6, 524-539 (2013)
2. Bassett, D.S., Bullmore, E.T. Human brain networks in health and disease. Curr. Opin. Neurol. 22, 340–347 (2009)
3. Bullmore, E., Sporns, O. The economy of brain network organization. Nat. Rev. Neurosci. 13, 336–349 (2012)
4. First, M. et al. Consensus Report of the APA Work Group on Neuroimaging Markers of Psychiatric Disorders (2012)
5. Kana, R.K, Uddin, L.Q., Kenet, T., Chugani, D., Mller, R.A. Brain connectivity in autism. Frontiers in human neuroscience 8, 349 (2014)

6. Hernandez, L.M., Rudie, J.D., Green, S.A., Bookheimer, S., Dapretto, M. Neural signatures of autism spectrum disorders: insights into brain network dynamics. Neuropsychopharmacology 40, 171–189 (2015)
7. Hastie, T., Tibshirani, R., Friedman, J. The elements of statistical learning. Springer (2001)
8. 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)
9. Brown, J.A., Rudie, J.D., Bandrowski, A., Van Horn, J.D., Bookheimer, S.Y. The UCLA multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. Frontiers in Neuroinformatics 6, 28 (2012)
10. Available online at: http://umcd.humanconnectomeproject.org
11. Mori, S., van Zijl, P.C. Fiber tracking: principles and strategies a technical review. NMR in Biomedicine 15, 468–480 (2002)
12. Available online at: http://trackvis.org/dtk
13. 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. Functional net- work organization of the human brain. Neuron 72, 665–678 (2011)
14. Jones, D.K., Knsche, T.R., Turner R. White matter integrity, fiber count, and other fallacies: the do's and don'ts of diffusion MRI. Neuroimage 73, 239–254 (2013)
15. 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, 1262–1279 (2011)
16. 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)
17. 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, 15684–15693 (2009)
18. 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, 3784–3804 (2012)
19. Rubinov, M., Sporns, O., Complex network measures of brain connectivity: uses and interpretations. Neuroimage 52, 3, 1059–1069 (2010)
20. Freeman, L.C. Centrality in social networks: conceptual clarification. Soc. Netw. 1, 215–239 (1978)
21. Bonacich, P. Power and Centrality: A Family of Measures. American Journal of Sociology 92, 1170–1182 (1987)
22. Wasserman, S., Faust, K. Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press (1994)
23. Newman, M.E.J. Assortative mixing in networks, Phys. Rev. Lett. 89, 208701 (2002)
24. Refaeilzadeh, P., Tang, L., Liu, H., Cross validation. Encyclopedia of Database Systems. Springer (2009)
25. Vapnik, V.N. Statistical Learning Theory. Wiley-Interscience (1998)
26. Pérez, F., Granger, B. E.: IPython: A System for Interactive Scientific Computing. Computing in Science & Engineering, 9, 21–29 (2007)
27. van der Walt, S., Colbert, S. C., Varoquaux, G: The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13, 22–30 (2011)

28. Hagberg, A. A., Schult, D. A., Swart, P. J.: Exploring network structure, dynamics, and function using NetworkX. Proceedings of the 7th Python in Science Conference 11–15 (2008)
29. 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.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830 (2011)
30. Hunter, J. D. Matplotlib: A 2D graphics environment. Computing In Science Engineering 9, 3. 90–95 (2007)