Graph Neural Network for Interpreting Task-fMRI Biomarkers

Xiaoxiao Li[1(✉)], Nicha C. Dvornek[5], Yuan Zhou[5], Juntang Zhuang[1], Pamela Ventola[4], and James S. Duncan[1,2,3,5]

[1] Biomedical Engineering, Yale University, New Haven, CT, USA
xiaoxiao.li@yale.edu
[2] Electrical Engineering, Yale University, New Haven, CT, USA
[3] Statistics and Data Science, Yale University, New Haven, CT, USA
[4] Child Study Center, Yale School of Medicine, New Haven, CT, USA
[5] Radiology and Biomedical Imaging, Yale School of Medicine, New Haven, CT, USA

Abstract. Finding the biomarkers associated with ASD is helpful for understanding the underlying roots of the disorder and can lead to earlier diagnosis and more targeted treatment. A promising approach to identify biomarkers is using Graph Neural Networks (GNNs), which can be used to analyze graph structured data, i.e. brain networks constructed by fMRI. One way to interpret important features is through looking at how the classification probability changes if the features are occluded or replaced. The major limitation of this approach is that replacing values may change the distribution of the data and lead to serious errors. Therefore, we develop a 2-stage pipeline to eliminate the need to replace features for reliable biomarker interpretation. Specifically, we propose an inductive GNN to embed the graphs containing different properties of task-fMRI for identifying ASD and then discover the brain regions/subgraphs used as evidence for the GNN classifier. We first show GNN can achieve high accuracy in identifying ASD. Next, we calculate the feature importance scores using GNN and compare the interpretation ability with Random Forest. Finally, we run with different atlases and parameters, proving the robustness of the proposed method. The detected biomarkers reveal their association with social behaviors and are consistent with those reported in the literature. We also show the potential of discovering new informative biomarkers. Our pipeline can be generalized to other graph feature importance interpretation problems.

Keywords: Graph Neural Network · Task-fMRI · ASD biomarker

1 Introduction

Autism spectrum disorders (ASD) affect the structure and function of the brain. To better target the underlying roots of ASD for diagnosis and treatment, efforts to identify reliable biomarkers are growing [8]. Significant progress has been made using functional magnetic resonance imaging (fMRI) to characterize the

D. Shen et al. (Eds.): MICCAI 2019, LNCS 11768, pp. 485–493, 2019.
https://doi.org/10.1007/978-3-030-32254-0_54

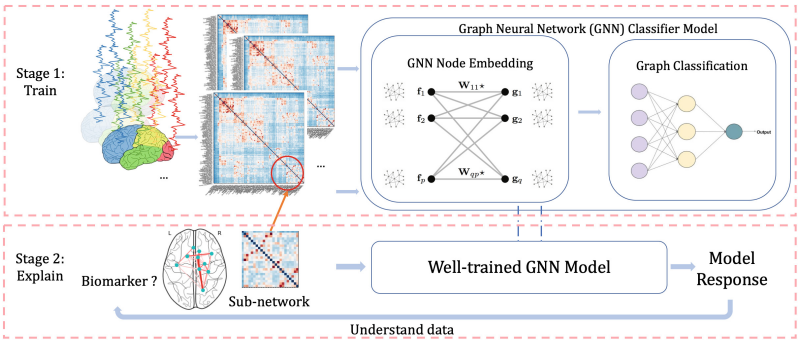

Fig. 1. Pipeline for interpreting important features from a GNN

brain remodeling in ASD [9]. Recently, emerging research on Graph Neural Networks (GNNs) has combined deep learning with graph representation and applied an integrated approach to fMRI analysis in different neuro-disorders [11]. Most existing approaches (based on Graph Convolutional Network (GCN) [10]) require all nodes in the graph to be present during training and thus lack natural generalization on unseen nodes. Also, it is necessary to interpret the important feature in the data used as evidence for the model, but currently no tool exists that can interpret and explain GNNs while recent CNN explanation algorithms cannot directly work on graph input.

Our main contributions include the following three points: (1) We develop a method to integrate all the available connectivity, geometric, anatomic information and task-fMRI (tfMRI) related parameters into graphs for deep learning. Our approach alleviates the problem of predetermining the best features and measures of functional connectivity, which is often ambiguous due to the intrinsic complex structure of task-fMRI. (2) We propose a generalizable GNN inductive learning model to more accurately classify ASD v.s. healthy controls (HC). Different from the spectral GCN [10], our GNN classifier is based on graph isomorphism, which can be applied to multigraphs with different nodes/edges (e.g. sub-graphs), and learn local graph information without binding to the whole graph structure. (3) The GNN architecture enables us to train the model on the whole graph and validate it on subgraphs. We directly evaluate the importance scores on sub-graphs and nodes (i.e. regions of interest (ROIs)) by examining model responses, without resampling value for the occluded features. The 2-stage pipeline to interpret important sub-graphs/ROIs, which are defined as biomarkers in our setting, is shown in Fig. 1.

2 Methodology

2.1 Graph Definition

We firstly parcellate the brain into N ROIs based on its T1 structural MRI. We define ROIs as graph nodes. We define an undirected multigraph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$,

where $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N)^T \in \mathbb{R}^{N \times D}$ and $\boldsymbol{E} = [\boldsymbol{e}_{ij}] \in \mathbb{R}^{N \times N \times F}$, D and F are the attribute dimensions of nodes and edges respectively. For node attributes, we concatenate handcrafted features: degree of connectivity, General Linear Model (GLM) coefficients, mean, standard deviation of task-fMRI, and ROI center coordinates. We applied the Box-Cox transformation [13] to make each feature follow a normal distribution (parameters are learned from the training set and applied to the training and testing sets). The edge attribute $\boldsymbol{e}_{ij}$ of node i and j includes the Pearson correlation, partial correlation calculated using residual fMRI, and $exp(-r_{ij}/10)$ where r_{ij} is the geometric distance between the centers of the two ROIs. We thresholded the edges under the 95th percentile of partial correlation values to ensure sparsity for efficient computation and avoiding oversmoothing.

2.2 Graph Neural Network (GNN) Classifier

The architecture of our proposed GNN is shown in Fig. 2 (node, edge attribute definition, kernel sizes are denoted). The model inductively learns node representation by recursively aggregating and transforming feature vectors of its neighboring nodes. Below, we define the layers in the proposed GNN classifier.

Convolutional Layer. Following Message Passing Neural Networks (NNconv) [7], which is invariant to graph symmetries, we leverage node degree in the embedding. The embedded representation of the lth convolutional layer $\boldsymbol{v}_i^{(l)} \in \mathbb{R}^{d^{(l)}}$ is

$$\boldsymbol{v}_i^{(l)} = \frac{1}{|\mathcal{N}(i)| + 1} \sigma(\boldsymbol{\Theta} \boldsymbol{v}_i^{(l-1)} + \sum_{j \in \mathcal{N}(i)} h_\phi(\boldsymbol{e}_{ij}) \boldsymbol{v}_j^{(l-1)}), \tag{1}$$

where $\sigma(\cdot)$ is a nonlinear activation function (we use `relu` here), $\mathcal{N}(i)$ is node i's 1-hop neighborhood, $\boldsymbol{\Theta} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$ is a learnable propagation matrix, h_ϕ denotes a Multi-layer Perceptron (MLP), which maps the edge attributes $\boldsymbol{e}_{ij}$ to a $d^{(l)} \times d^{(l-1)}$ matrix, and we initialize $\boldsymbol{v}_i^{(0)} = \boldsymbol{v}_i$.

Pooling Aggregation Layer. To make sure that down-sampling layers behave idiomatically with respect to different graph sizes and structures, we adopt the approach in [2] for reducing graph nodes. The choice of which nodes to drop is done based on projecting the node attributes on a learnable vector $\boldsymbol{w}^{(l-1)} \in$

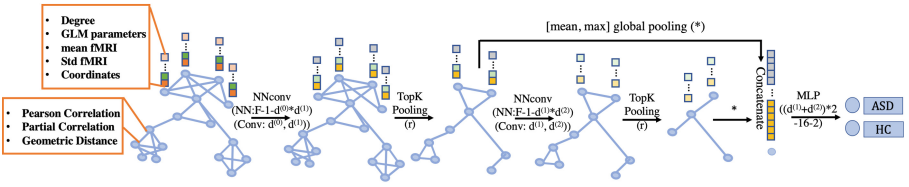

Fig. 2. The architecture of the GNN classifier

$\mathbb{R}^{d^{(l-1)}}$. The nodes receiving lower scores will experience less feature retention. Fully written out, the operation of this pooling layer (computing a pooled graph, $(\boldsymbol{V}^{(l)}, \boldsymbol{E}^{(l)})$, from an input graph, $(\boldsymbol{V}^{(l-1)}, \boldsymbol{E}^{(l-1)})$), is expressed as follows:

$$\boldsymbol{y} = \frac{\boldsymbol{V}^{(l-1)}\boldsymbol{w}^{(l-1)}}{\|\boldsymbol{w}^{(l-1)}\|} \quad \boldsymbol{i} = \text{top}k(\boldsymbol{y}, k) \quad \boldsymbol{V}^{(l)} = (\boldsymbol{V}^{(l-1)} \odot \tanh(\boldsymbol{y}))_{\boldsymbol{i},:} \quad \boldsymbol{E}^{(l)} = \boldsymbol{E}^{(l-1)}_{\boldsymbol{i},\boldsymbol{i}} \ . \quad (2)$$

Here $\|\cdot\|$ is the L_2 norm, top$k$ finds the indices corresponding to the largest k elements in vector $\boldsymbol{y}$, $\odot$ is (broadcasted) element-wise multiplication, and $(\cdot)_{\boldsymbol{i},\boldsymbol{j}}$ is an indexing operation which takes elements at row indices specified by $\boldsymbol{i}$ and column indices specified by $\boldsymbol{j}$ (colon denotes all indices). The pooling operation trivially retains sparsity by requiring only a projection, a point-wise multiplication and a slicing into the original feature and adjacency matrix. Different from [2], we induce constraint $\|\boldsymbol{w}^{(l)}\|_2 = 1$ implemented by adding an additional `regularization loss` $\lambda \sum_{l=1}^{L}(\|\boldsymbol{w}^{(l)}\|_2 - 1)^2$ to avoid identifiability issues.

Readout Layer. Lastly, we seek a "flattening" operation to preserve information about the input graph in a fixed-size representation. Concretely, to summarise the output graph of the lth conv-pool block, $(\boldsymbol{V}^{(l)}, \boldsymbol{E}^{(l)})$, we use

$$\boldsymbol{s}^{(l)} = (\frac{1}{N^{(l)}} \sum_{i=1}^{N^{(l)}} \boldsymbol{v}_i^{(l)}) \parallel \max(\left\{\boldsymbol{v}_i^{(l)} : i = 1, ..., N^{(l)}\right\}), \quad (3)$$

where $N^{(l)}$ is the number of graph nodes, $\boldsymbol{v}_i^{(l)}$ is the ith node's feature vector, max operates elementwise, and $\parallel$ denotes concatenation. The final summary vector is obtained as the concatenation of all those summaries (i.e. $\boldsymbol{s} = \boldsymbol{s}^{(1)} \parallel \boldsymbol{s}^{(2)} \parallel \cdots \parallel \boldsymbol{s}^{(L)}$) and submitted to a MLP for obtaining final predictions.

2.3 Explain Input Data Sensitivity

To explain input data sensitivity, we cluster the whole brain graph into sub-graphs first. Then we investigate the predictive power of each sub-graph, further assign importance score to each ROI.

Network Community Clustering. From now on we add the subscript to the graph as $\boldsymbol{G}_s = (\boldsymbol{V}_s, \boldsymbol{E}_s)$ for the sth instance, $s = 1, ..., S$, where S is the number of graphs. Concatenating the sparsified non-negative partial correlation matrices $(\boldsymbol{E}_s)_{:,:,2}$ for all the graphs, we can create a 3rd-order tensor τ of dimension $N \times N \times S$. Non-negative PARAFAC [3] tensor decomposition is applied to tensor τ to discover overlapping functional brain networks. Given decomposition rank R, $\tau \approx \sum_{j=1}^{R} \boldsymbol{a}_j \otimes \boldsymbol{b}_j \otimes \boldsymbol{c}_j$, where loading vectors $\boldsymbol{a}_j \in \mathbb{R}^N$, $\boldsymbol{b}_j \in \mathbb{R}^N$, $\boldsymbol{c}_j \in \mathbb{R}^S$ and $\otimes$ denotes the vector outer product. $\boldsymbol{a}_j = \boldsymbol{b}_j$ since the connectivity matrix is symmetric. The ith element of $\boldsymbol{a}_j$, a_{ji} provides the membership of region i in the community j. Here, we consider region i belongs to community j if $a_{ji} > mean(\boldsymbol{a}_j) + std(\boldsymbol{a}_j)$ [12]. This gives us a collection of community indices indicating region membership $\{\boldsymbol{i}_j \subset \{1, ..., N\} : j = 1, ..., R\}$.

Graph Salience Mapping. After decomposing all the brain networks into community sub-graphs $\left\{\boldsymbol{G}_{sj} = ((\boldsymbol{V}_s)_{\boldsymbol{i}_j,:}, (\boldsymbol{E}_s)_{\boldsymbol{i}_j,\boldsymbol{i}_j}) : s = 1, ..., S, j = 1, ..., R\right\}$, we use a salience mapping method to assign each sub-graph an importance score. In our classification setting, the probability of class $c \in \{0, 1\}$ (0: HC, 1: ASD) given the original network $\boldsymbol{G}$ is estimated from the predictive score of the model: $p(c|\boldsymbol{G})$. To calculate $p(c|\boldsymbol{G}_{sj})$, different from CNN or GCN, we can directly input the sub-graph into the pre-trained classifier. We denote c_s as the class label for instance s and define *Evidence for Correct Class (ECC)* for each community:

$$ECC_j = \frac{1}{S}\sum_s \tanh(\log_2(p(c = c_s|\boldsymbol{G}_{sj})/(1 - p(c = c_s|\boldsymbol{G}_{sj})))), \tag{4}$$

where laplace correction ($p \leftarrow (pS + 1)/(S + 2)$) is used to avoid zero denominators. Note that log odds-ratio is commonly used in logistic regression to make p more separable. The nonlinear tanh function is used for bounding *ECC*. *ECC* can be positive or negative. A positive value provides evidence for the classifier, whereas a negative value provides evidence against the classifier. The final importance score for node k is calculated by $\sum_{j:k\in\boldsymbol{i}_j} ECC_j/|\boldsymbol{i}_j|$. The larger the score, the more possible the node can be used as a distinguishable marker.

3 Experiments and Results

3.1 Data Acquisition and Preprocessing

We tested our method on a group of 75 ASD children and 43 age and IQ-matched healthy controls collected at Yale Child Study Center. Each subject underwent a task fMRI scan (BOLD, TR = 2000 ms, TE = 25 ms, flip angle = 60°, voxel size $3.44 \times 3.44 \times 4\,\text{mm}^3$) acquired on a Siemens MAGNETOM Trio TIM 3T scanner. For the fMRI scans, subjects performed the "biopoint" task, viewing point light animations of coherent and scrambled biological motion in a block design [9] (24 s per block). The fMRI data was preprocessed following the pipeline in [14].

The mean time series for each node were extracted from a random 1/3 of voxels in the ROI (given an atlas) of preprocessed images by bootstrapping. We augmented the ASD data 10 times and the HC data 20 times, resulting in 750 ASD graphs and 860 HC graphs separately. We split the data into 5 folds based on subjects. Four folds were used as training data and the left out fold was used for testing. Based on the definition in Sect. 2.1, each node attribute $\boldsymbol{v}_i \in \mathbb{R}^{10}$ and each edge attribute $\boldsymbol{e}_{ij} \in \mathbb{R}^3$. Specifically, the GLM parameters of "biopoint task" are: β_1: coefficient of biological motion matrix; β_3: coefficient of scramble motion matrix; β_2 and β_4: coefficients of the previous two matrices' derivatives.

3.2 Step 1: Train ASD/HC Classification Model

Firstly, we tested classifier performance on the Destrieux atlas [5] (148 ROIs) using proposed GNN. Since our pipeline integrated interpretation and classification, we apply a random forest (RF) using 1000 trees as an additional "reality check", while the other existing graph classification models either cannot

Table 1. Performance of different models (mean ± std)

Model	RF(V)	RF(E)	RF(V+E)	GNN(r = 0.3)	GNN(r = 0.5)	GNN(r = 0.8)
Accuracy	0.71 ± 0.05	0.66 ± 0.06	0.68 ± 0.06	0.67 ± 0.14	**0.76 ± 0.06**	0.73 ± 0.07
F-score	0.69 ± 0.06	0.68 ± 0.06	0.63 ± 0.12	0.68 ± 0.09	**0.79 ± 0.08**	0.71 ± 0.10
Precision	0.68 ± 0.06	0.61 ± 0.06	0.69 ± 0.12	0.65 ± 0.19	**0.76 ± 0.12**	0.68 ± 0.08
Recall	0.73 ± 0.12	0.76 ± 0.10	0.77 ± 0.09	0.74 ± 0.07	**0.82 ± 0.06**	0.75 ± 0.08

achieve the performance as GNN [2,7] or do not have straightforward and reliable interpretation ability [1]. We flattened the features to $\boldsymbol{V} \in \mathbb{R}^{1480}$ and $\boldsymbol{E} \in \mathbb{R}^{65712}$ ($65712 = 148 \times 148 \times 3$) and used this input to the RF. In our GNN, $d^{(0)} = D = 10$, $d^{(1)} = 16, d^{(2)} = 8$, resulting in 2746 trainable parameters and we tried different pooling ratios r ($k = r \times N$) in Fig. 2, which was implemented based on [6]. We applied `softmax` after the network output and combined `cross entropy loss` and `regularization loss` with $\lambda = 0.001$ as the objective function. We used the Adam optimizer with initial learning 0.001, then decreased it by a factor of 10 every 50 epochs. We trained the network 300 epochs for all of the splits and measured the instance classification by accuracy, F-score, precision and recall (see Table 1). Our proposed model significantly outperformed the alternative method, due to its ability to embed high dimensional features based on the structural relationship. We selected the best GNN model with $r = 0.5$ in the next step: interpreting biomarkers.

3.3 Step 2: Interpret and Explain Biomarkers

We put forth the hypothesis that the more accurate the classifier, the more reliable biomarkers can be found. We used the best RF model using $\boldsymbol{V}$ as inputs (77.4% accuracy on testing set) and used the RF-based feature importance (mean Gini impurity decrease) as a form of standard method for comparison. For GNN interpretation, we also chose the best model (83.6% accuracy on testing set). Further, to be comparable with RF, all of the interpretation experiments were performed on the training set only. The interpretation results are shown in Fig. 3, where the top 30 important ROIs (averaged over node features and instances)

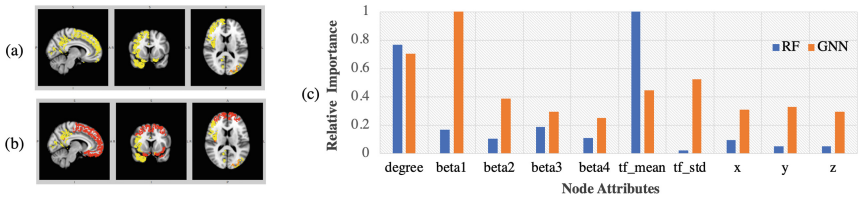

Fig. 3. (a) Top 30 important ROIs (colored in yellow) selected by RF; (b) Top 30 important ROIs selected by GNN ($R = 20$) (colored in red) laying over (a); (c) Node attributes' relative importance scores in the two methods. (Color figure online)

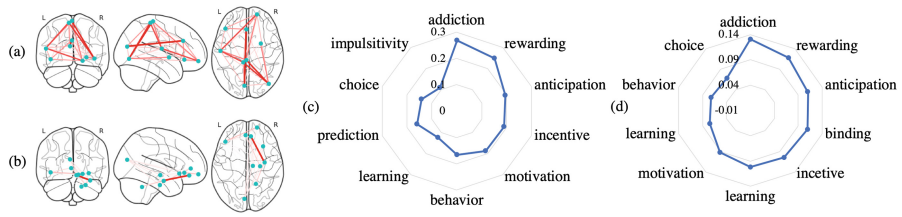

Fig. 4. (a), (c) Top scoring sub-graph and corresponding functional decoding keywords and coefficients. (b), (d) The 2nd high scoring sub-graph and corresponding functional decoding keywords and coefficients.

selected by RF are shown in yellow and the top 30 important ROIs selected by our proposed GNN in red. Nine important ROIs were selected by both methods. In addition, for node attribute importance, we averaged the importance score over ROIs and instances for RF. For GNN, we averaged *gradient explanation* over all the nodes and instances, i.e. $\mathbb{E}(\frac{1}{N}\sum_i |\frac{\partial y}{\partial v_{ij}}|)$, where $y = p(c = 1|\boldsymbol{G})$, which quantifies the sensitivity of the jth node attribute. From Fig. 3(c) we show the relative importance to the most important node attribute, our proposed method assigned more uniform importance to each node attribute, among which the biological motion parameter β_1 was the most important. In addition, similar features, mean/std of task-fMRI (tf_mean/tf_std) and coordinates (x, y, z), have similar scores, which makes more sense for human interpretation. Notice that our proposed pipeline is also able to identify sub-graph importance from Eq. (4), which is helpful for understanding the interaction between different brain regions. We selected the top 2 sub-graphs ($R = 20$) and used Neurosynth [15] to decode the functional keywords associated with the sub-graphs (shown in Fig. 4). These networks are both associated with high-level social behaviors. To illustrate the predictive power of the 2 sub-graphs, we retrained the network using the graph slicing on those 19 ROIs of the 2 sub-graphs as input. Accuracy on the testing set (in the split of the best model) was 78.9%, achieving comparable performance to using the whole graph.

3.4 Evaluation: Robustness Discussion

To examine the potential influence of different graph building strategies on the reliability of network estimates, the functional and anatomical data were registered and parcellated by the Destrieux atlas (*A1*) and the Desikan-Killiany atlas (*A2*) [4]. We also showed the robustness of the results with respect to the number of clusters for $R = 10, 20, 30$. The results are shown in Fig. 5. We ranked *ECC*s for each node and indicated the top 30 ROIs in *A1* and top 15 ROIs in *A2*. The atlas and number of clusters are indicated on the left of each sub-figure. Orbitofrontal cortex and ventromedial prefrontal cortex are selected in all the cases, which are social motivation related and have previously been shown to be associated with ASD [9]. We also validated the results by decoding the neurological functions of the important ROIs overlapped with Neurosynth.

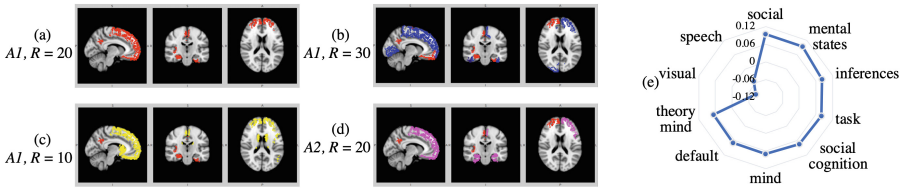

Fig. 5. (a) The biomarkers (red) interpreted on *A1* with 20 clusters; (b)–(d) The biomarkers interpreted by different R and atlas laying over on (a) with different colors; (e) The correlation between overlapped ROIs and functional keywords. (Color figure online)

4 Conclusion and Future Work

In this paper, we proposed a framework to discover ASD brain biomarkers from task-fMRI using GNN. It achieved improved accuracy and more interpretable features than the baseline method. We also showed our method performed robustly on different atlases and hyper-parameters. Future work will include investigating more hyper-parameters (i.e. suitable size of sub-graphs communities), testing the results on functional atlases and different graph definition methods. The pipeline can be generalized to other feature importance analysis problems, such as resting-fMRI biomarker discovery and vessel cancer detection.

Acknowledgment. This work was supported by NIH Grant R01 NS035193.

References

1. Adebayo, J., et al.: Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems, pp. 9505–9515 (2018)
2. Cangea, C., et al.: Towards sparse hierarchical graph classifiers. arXiv preprint arXiv:1811.01287 (2018)
3. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika **35**(3), 283–319 (1970)
4. Desikan, R.S., et al.: An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. NeuroImage **31**(3), 968–980 (2006)
5. Destrieux, C., et al.: Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. NeuroImage **53**(1), 1–15 (2010)
6. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch geometric. CoRR abs/1903.02428 (2019)
7. Gilmer, J., et al.: Neural message passing for quantum chemistry. In: ICML 2017, pp. 1263–1272. JMLR.org (2017)
8. Goldani, A.A., et al.: Biomarkers in autism. Front. Psychiatry **5**, 100 (2014)
9. Kaiser, M.D., et al.: Neural signatures of autism. Proc. Nat. Acad. Sci. **107**(49), 21223–21228 (2010)

10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
11. Ktena, S.I., et al.: Distance metric learning using graph convolutional networks: application to functional brain networks. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10433, pp. 469–477. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66182-7_54
12. Loe, C.W., Jensen, H.J.: Comparison of communities detection algorithms for multiplex. Physica A: Stat. Mech. Appl. **431**, 29–45 (2015)
13. Nishii, R.: Box-Cox Transformation. Encyclopedia of Mathematics. Springer, New York (2001)
14. Yang, D., et al.: Brain responses to biological motion predict treatment outcome in young children with autism. Transl. Psychiatry **6**(11), e948 (2016)
15. Yarkoni, T., et al.: Large-scale automated synthesis of human functional neuroimaging data. Nat. Methods **8**(8), 665 (2011)