Adaptive Rewiring on Logistic Maps with Heterogeneous Parameters

# Method

This section treats three main technical steps taken in this study. First, the models used in this study are described in detail; the oscillator model used for neural masses (i.e., coupled logistic maps) are introduced and the birth of network models and their maturation through adaptive rewiring are discussed. Then, we discussed the qualitative and quantitative methods of network characterization. Finally, we discuss the methods of quantifying and reporting the structural resemblance of our models.

## Making and growing brains

### Mathematical description of networks

In mathematics, a graph is a set of 3-tuples that define the relation (called edge or connection) between pairs of objects and (called vertices or nodes). The set of edges and vertices are represented by E and V, respectively. The value of characterizes the strength of association between vertices and and hence is often called edge weight. Conventionally, the edges with zero values (i.e., no connection between the vertices) are omitted from the set E. If , the graph is called unweighted or binary. In the case where , the network is called an undirected network.

The nodes of a graph can be depicted by circles and association between the nodes are represented graphically by circles as nodes and (directed) lines connecting them. A graph can be formally described by a set of 3-tuples, an edge list, or an adjacency matrix. Adjacency matrix is a square matrix of the size |V| where each element is the edge weight of the connection between the nodes i and j. If the graph is undirected, the matrix is symmetrical around its main diagonal, and if it is unweighted all the elements take values of zero or one.

### Coupled logistic maps

Brain activity in neural masses (elaboration on neural masses?) can be described as chaotic oscillations governed by the attractor in Fig 1A (Breakspear, Terry, & Friston, 2003). Dimensional reduction via Poincare section yields the relationship in 1B that can approximately be described by a logistic map 1C).

The logistic maps is of the form shown in Equation 1.

EQ1

Logistic maps are well known to exhibit universal dynamical properties (van Strien, 1987). For certain regimes of the chaos parameter, the behavior of the logistic map converges to one or more attractors, but otherwise it exhibits chaotic behavior. In these regimes, logistic maps produce deterministic stochastic bounded time series that give reasonable approximation of the activities of neural masses. The values of the 200 draws of logistic maps (after a burn-in period of 4000 iterations) are shown in Feigenbaum plots in Figure 2. The parameters of the logistic maps in the current study were chosen in the range of 1.7-1.9, as it grants stochastic signals bounded in the range of [-1, 1] with properties suitable for the adaptive rewiring algorithm[[1]](#footnote-1). Each neuron in each brain was assigned a fixed value for \alpha, and the initial x\_1 values of activations were drawn uniformly from the range of [0, 1]. This random draw was based on the same seeds used in constructing the random graphs. Since the models are mathematically described (and simulated) by matrix algebra, the same notation will be used from now on.

Whereas the logistic map could be considered as an abstract representation of a neural mass oscillators, systems of coupled neural mass oscillators may be represented by coupled logistic maps. Because of the universal dynamics of logistic maps, networks of such simple maps may capture generic properties of interacting nonlinear systems. (Kaneko, 1992). The logistic maps are coupled according to Equation 2 as such the activity of each neuron is moderated by the activity of its neighbors. In a matrix notation, for a network with N nodes, the activity of neurons at time t+1 is calculated via

EQ2

In the right-hand side of Equation 2, the vector form of logistic maps is Hadamard-multiplied[[2]](#footnote-2) by a coupling term. In the coupling term, is the vector of coupling strengths and is the connectivity matrix at time t. in the denominator is a vertical unit vector of size N. Division of by normalizes the former (i.e., the effect of of node i) by sum of the weights of the edges connected to each node. Since our graphs are binary, the term in the denominator counts the number of connections for each node.

We model binary networks of chaotic coupled logistic oscillators with different parameterizations and let them evolve by means of adaptive rewiring algorithm. In this model, every node is characterized by two parameters, coupling strength and chaos, and a global rewiring algorithm lets the network structure to evolve. There have been five different sets of parameters, for which a total of 50 models (10 model per parameter set) are analyzed. In order to ease the discussion, we use metaphors to describe our models: each model is assumed to be a “brain” where the oscillators at the nodes are called, loosely speaking, “neurons”. Moreover, the brains have been assigned a randomly generated name, and each parameter set is called a “family”. Brains within a family have identical parameters and the only difference between family members is the random seed used to assign the initial values at their birth. In what follows, we go through the elements of our models: birth of the brains; characteristics of the oscillators; the rewiring algorithm; making of families; and the qualitative and quantitative measures of network structures. All the simulations and analyses are conducted in R programming language version 3.5.0 (R Core Team, 2018).

### Creating the networks

In this study, building on previous literature (citation), we modeled an undirected graph with 300 nodes and 5200 unweighted edges. The models (“brains”) simulated in this study are undirected graphs of |V| = 300 “neurons” and |E| = 5200 unweighted connections. The initial graph was constructed by randomly imputing matrices of zeros with ones, as such the matrix becomes symmetrical (undirected) and has zeros on diagonal (without any loops). The random numbers were generated based on a seed which depended on |V|, |E|, and a generation parameter `round`.

### Adaptive rewiring of the network

At each rewiring attempt, the connections of a random node is optimized locally as such the node is disconnected from its most dissimilar neighbor and is connected to the most similar nodes to which it was not connected. The dissimilarity of two neurons at a given time is defined as the absolute value of the difference in the value of their activity. To do so, at time t, a node i is selected randomly from V, and a vector[[3]](#footnote-3) of its distance from other nodes is calculated as , and another vector of similarities is defined as .

The most dissimilar neighbor and the most similar non-neighbor of the neuron, denoted as and , respectively, are found by finding the index of the maxima of the following vectors:

The matrix multiplication of M and (1-M) with , respectively, ensures that the search for the edges subject to rewiring happens in the right subset of edges. The rewiring is then changing the corresponding elements of the adjacency matrix:

In this study, a rewiring attempt takes place after every 20 updates of the activities of the logistic maps. In this one million rewirings were done, meaning that each neuron fired 20 million times.

### Setting the parameters and making families of brains

As we have seen, each brain can be characterized by three constant parameters: `round`, which determines the seed used to generate the initial random graph and set the starting values of the logistic maps; chaos parameter for each node; and , the vector of coupling strengths of each neuron. Previous research (citation) and preliminary exploratory simulations suggest that adaptive rewiring on coupled logistic maps realize best for and in the ranges of [1.7-1.9] and [0.3-05], respectively. In this study, the midpoints of the ranges are assumed the baseline values for the parameters. The nodes with lower and higher values in the desired ranges of and are assumed to be hypo- and hyper-chaotic and hypo- and hyper-coupled, respectively.

To answer the research question, i.e., to what extend … (phrasing? Should write the introduction), the neurons were assigned to two partitions of minority (the first 50 nodes) and majority (the remaining 250 nodes). While keeping the parameters of the majority at the baseline () five different combinations of parameters were assigned to the minorities, each combination called a “family”: The homogeneous family (), and the families with hypo-chaotic minority (), hyper-chaotic family (), hypo-coupled minorities (), and hyper-coupled minority ().

In each family, ten generation of brains are simulated to increase the robustness of the outcomes. The only difference between the generations are in the `round` parameters used to initialize the brains. The generations across families are matched in that parameter, i.e., the initial network structure and activation values of generation k are identical in all families. With this choice, it is possible to match brains starting with the same conditions to see the effect of different parameterizations more clearly.

## Characterizing network structures

As mentioned earlier, each brain is characterized by the set of abovementioned parameters that remain constant over the course of simulation. The state of the network at any given time t can be described by the connectivity matrix and the neuron activities, i.e., , respectively. Since this study seeks to answer questions about the structure of the networks under certain conditions (i.e., different parameterizations of and ), we focus here on describing the structural (or anatomical) connectivity of the brains and the evolution of these measures over time as the brains mature. Furthermore, as we need to draw general conclusions about the effect of different parameterizations, we also define measures to quantify (dis)similarities among the matured brains.

### Qualitative description of network structures.

The network structure can be qualitatively assessed by means of visual investigation of the graph diagram or by visualizing the adjacency matrix. Since the neurons belonging to a given partition are not distinguishable, the raw adjacency matrix can only give an estimate of the edge density within and between partitions but fails to provide intuitions into the structure of the network. Hence, the adjacency matrix is serialized using … algorithm, implemented in the package `seriation` (citation), which orders the rows and columns of the matrix to maximize visibility of modules within the network (better/more precise phrasing?). The raw (unserialized) and ordered (serialized) adjacency matrices, and the graph diagrams are plotted using `seriation` and `igraph` packages. In the plots, the minority and majority neurons are colored sky blue and pink, respectively. In both matrix visualizations and graph diagrams the within-minority and within-majority edges are colored blue and red, respectively, while the inter-partition edges, connecting neurons of minority subset to neurons of majority, are colored green.

### Quantitative measures of the structure.

In network science, a wide range of structural measures of connectivity, also known as network (summary (?)) statistics, have been proposed (Costa, Rodrigues, Travieso, & Villas Boas, 2007). After each rewiring attempt, we calculate eight network connectivity measures for the whole network and three subsets of edges, namely, intra-minority, intra-majority, inter-partition. Furthermore, we calculate three additional measures for the matured brains at the last rewiring iteration. These measures are explained in what follows.

#### Clustering coefficient.

This measure gives an indication of the tendency of nodes to form clusters and can be defined either locally or globally. We use global clustering coefficient, which is defined as the number of closed triplets of nodes (the triplets of nodes that are all connected) divided by the number of connected triplets, either open (paths of length 2) or closed (triangles). The numerator is equal to three times the number of triangles in the graph. Using linear algebra (cite StackExchange RMurphy?), it can be shown that the global clustering coefficient can be calculated formally from the adjacency matrix as shown in EQ3.

EQ 3

#### Modularity.

Modularity of a graph, as proposed by Newman (2006) and denoted by Q, is a measure of how (and to what degree), for a certain labeling of nodes, the nodes tend to form communities with the nodes of the same label and tend to not connect to other nodes of the graph. This measure requires a priori labeling of nodes that defines the communities to which they are believed to belong. The labeling can be done manually based on theoretical knowledge or arbitrary decisions.

There has been a line of research on algorithmic discovery of optimal modules (also known as communities) within graphs such that the measures of modularity is maximized (for a review of the proposed methods, cf. Zhang, Ma, Zhang, Sun, & Yan, 2018). The communities discovered by these algorithms can thus be used as labels for calculating modularity of the network. In this study, in line with (Clauset, Newman, & Moore, 2004), using `igraph` package, the fast greedy algorithm is used to oqptimally detect communities and calculate the modularity based on community membership of the nodes.

#### Average path length.

Average path length is the mean value of lengths of shortest path between all pairs of nodes. This measure, calculated using `igraph` package, gives an indication of how closely the nodes of a network are located from each other.

#### Efficiency.

The efficiency of a graph quantifies the efficiency of information exchange within the network and is defined as sum of inverses of the distances between nodes. In order to make the measure comparable across graphs of different size, the average efficiency is often used which is defined in Equation 4.

EQ 4

In EQ4, N is the size of the network and are the distances between unidentical nodes i and j. The denominator is twice the number of all possible edges in a graph of size N.

#### Small-worldness.

Small-worldness is a measure of the degree to which the graph shows properties akin to the structures known as small world (citation). It is defined as the multiplication of normalized clustering coefficient and efficiency of the network, i.e.. are the expected clustering coefficient and efficiency of a random graph of the same as the graph in question. Since all networks modeled in this study start off with random networks of the same size and density, for computational reasons, a non-normalized version of small-worldness coefficient (i.e., ) is calculated and reported.

#### Assortativity.

Assortativity coefficient indicates the preferences of nodes to connect to “similar” nodes by summarizes the probability of connections between the similar nodes. The similarity is can be imposed externally, e.g., by assigning categories to the nodes using labels (known as nominal assortativity), or by internal criteria like the node degrees (degree assortativity). Degree assortativity, calculated using `igraph` package in this study, measures the probability that nodes of similar degree (i.e., number of connections) are connected. (citation?)

#### Rich Club coefficient.

This coefficient quantifies the tendency of nodes with higher than a certain degree to connect to each other. More formally, as EQ5 shows, it is equivalent to the edge density (see below) of the subgraph of the network where the nodes with lower degrees than the cut-off value k are removed. Since this coefficient is a function of club size k, it is hard to visualize its evolution over time for all possible values of k. Hence, the values of this coefficient were once plotted for k=XXX to show network maturation and once as a function of k for the final network. In this study, we used `brainGraph` package (citation) to calculate rich club coefficient.

EQ 5 {Citation}

It is worth noting that the absolute value of the rich club coefficient is not comparable among networks of different sizes and densities and thus is hard to interpret in isolation. Hence, this coefficient is often normalized by the average rich club coefficient of Erdős–Rényi random network with similar number of nodes and edges. In doing so, one can generate multiple Erdős–Rényi networks of appropriate size and compare the absolute rich club coefficient of the network in question with the distribution of the same coefficient over the set of randomly generated networks by performing 1-sample t-test. However, the Erdős–Rényi model can generate networks with certain edge densities and fails to generate such networks for the minority, majority, and interpartition subsets of our brains. Hence, for these subsets, only the absolute values of the rich club coefficients are reported which are sufficient for relative comparison of the coefficient across the brains. For the whole networks, however, the normalized rich club coefficient is reported, and the significant values are marked by black diamonds in the plots.

#### Edge density.

For a subset of nodes, this coefficient is the proportion of existing edges to the maximum number of edges possible in that subset. It is used to quantify the normalized density of edges labeled as within-minority, within-majority, and inter-partition, as well as the whole network. Since the total number of edges remains the same during the adaptive rewiring, this coefficient gives an indication of how strongly each partition has attracted the new nodes into itself at every rewiring step. One can visually estimate the value of this coefficient by the density of each color in the subsets of the unserialized adjacency matrix.

#### Vertex and edge betweenness.

Vertex betweenness is a measure of centrality, quantifying to what degree a vertex (i.e., node) is placed in a central position in the network. The betweenness of a vertex $v$ can be defined as the sum of the proportion of shortest paths among all pairs of vertices of the network that go through it. Similarly, betweenness of edge $e$ can be defined as the proportion of shortest path that include that edge. Eq 6 and 7 show how these measures can be calculated. In these equations, is the number of shortest paths between nodes i and j, and are the number of path between i and j that include, respectively, node $v$ or edge $e$.

Eq 6 V

Eq 7 E

Vertex and edge betweenness are local measures; they are defined for particular nodes or edges. To characterize the networks globally, one can look at the distribution of their values for all nodes in the network. These distributions were calculated and plotted for all subsets of the brain networks for all our models.

## Looking for resemblance.

The quantitative measures of network structures discussed so far yield a dozen of plots for each brain, all of which with four curves for the ­­­­­­subsets of the networks. These outcomes are very informative when describing the structures of the networks in isolation. For the ultimate goal of this research (i.e., studying the effect of heterogenous parameterization of the models on the structures), however, we need to draw conclusions based on comparison of the structures of families with each other. Since we have five families of brains (with ten members each), it is not feasible to use these measures to draw such conclusions. Moreover, as we will shortly discuss in the Results section, it is hard to find common themes and consistent patterns specific to each family to base our comparisons upon. Consequently, we need to find measures allowing us to compare families of brains with each other.

The strategy taken in this study was to make pairwise network comparisons of 1225 unique pairs of brains using the methods discussed below. Then, having quantitative measures for (dis)similarities among the networks, we quantified the within-family resemblances and between-family contrasts among the brains. Finally, by defining a measure for family distinction (?) we compared how families vary in the … (?)

Dozens of methods have been proposed to be used in network comparison (for a review, see citation needed). The well-known methods are adapted to cases where the nodes are labeled (i.e., distinguishable) a priori. These methods are not always suitable for non-labeled networks (wherein the nodes are not assigned to any labels, which is the case for our models), as they require a primary step of inferring (or estimating) node labels or matching the nodes for the networks being compared (also known as node correspondence problem), which is computationally very costly (citation needed). Here, we use two methods suitable for our case that are inspired by NetSimile algorithm (Berlingerio, Koutra, Eliassi-Rad, & Faloutsos, 2012).

Each brain model consists of two sub-models; a network model for its anatomical connectivities and a logistic map for the activities of its neural masses. Hence, one can compare any pair of brains with regards to their anatomical structure and node activities. The anatomical connectivity of a brain model at each time is reflected in its adjacency matrix (M) and is subject to adaptive rewiring. We can also define another network for “functional connectivity” of neural masses (A) by calculating pairwise distances of node activities at each time. From a mathematical point of view, M and A are both undirected networks, one with binary edges and one with weighted edges. Thus, we can—and did—use the same methods of network comparisons for both anatomical and functional connectivities. Note that the term “network structures” applies to both M and A.

The remaining of this section discusses these methods, the rationale behind choosing them in this study, and the assumption they rest upon. Then, within- and between-family similarities are defined and a measure of family distinction is proposed. Finally, we discuss the method used in visualizing the results.

### Networks as distributions.

As we have seen, there are numerous network statistics that can describe network structures either locally or globally. The local measures are suitable for node-wise (or clique-wise) comparisons, while the global measures are aggregates of some local properties that provide “summary statistics” for the structure. The local measures hardly lead to holistic description of networks as the nodes are usually described in isolation from other nodes. On the other hand, valuable structural information is sacrificed along aggregation taking place in deriving global structural measures. Therefore, neither local nor global measures are optimal for comparison of networks. A solution to this issue would be taking an intermediary (?) approach by putting the distribution of various local measures under the spotlight. This way, we get to keep all[[4]](#footnote-4) the information we can harvest from structure. Then, we can decide how to use this information in network comparison.

Berlingerio et al. (2012) suggest characterizing each node i of the network with a seven-dimensional feature vector consisting of the following local measures that capture characteristics of the node and its surrounding subset of the network: , degree (i.e., number of neighbors); , local clustering coefficient (i.e., the number of triangles connected to node i over the number of connected triples centered on node i); , average degree of node’s neighbors; , average clustering coefficient of node’s neighbors; , the number of edges in the egonet[[5]](#footnote-6) of node i; , the number of outgoing edges from the egonet of node i[[6]](#footnote-7); and , the number of neighbors of egonet of node i. Although one can add more local features to this vector, Berlingerio and colleagues have shown that these features suffice for decent comparison of networks in their algorithm, i,e, NetSimile. Having the feature vectors of all nodes, we reach a 7-variate distribution which can be used to compare networks.

### The NetSimile method.

Given the distributions of local features, one can compare the distributions by means of comparing their summary statistics. In NetSimile, the feature distribution (which is a nodesXfeatures matrix) is aggregated into a 35-dimensional signature vector consisting of five summary statistics for each feature: median, mean, standard deviation, skewness, and kurtosis. The comparison of networks is thus reduced to calculating distances (or similarities) of the signature vectors. NetSimile is superior to other methods of inferring network similarity as its computational complexity grows linearly with the size of the networks, and more importantly, it allows comparison of networks of different sizes. One now must define an appropriate method of calculating the distances.

As we know (citation needed?), the ranks of summary statistics characterize the overall shape of distributions thus is a highly discriminative metric in their comparison. Then, the values of these summary statistics provide information about the similarities among the distributions on top of their overall shapes. Hence, the signature vectors are akin to ranked lists. It has been shown that the Canberra distance, defined in EQ8, is an appropriate measure of dissimilarity for ranked lists (Jurman, Riccadonna, Visintainer, & Furlanello, 2009) as it is sensitive to small distances from zero and normalizes the pairwise distances of features by their absolute values. Moreover, Berlingerio and colleagues (2012) report high discriminative power of Canberra distance in comparison of signature vectors, a good property of a dissimilarity for the task at hand.

EQ 8

In this study, we used this dissimilarity metric in the pairwise comparison of the signature vectors derived from NetSimile algorithm. However, NetSimile does not allow hypothesis testing to infer significance levels for the distances. Berlingerio et al. (2012) suggest hypothesis testing for independence of the distributions by pairwise comparison of the univariate distributions of the features and aggregating their p-values through averaging or choosing the maximum values. They report that neither Mann-Whitney nor Kolmogorov-Smirnov tests—which are nonparametric tests without any assumption for the distributions being compared—yield amply meaningful discrimination among the networks being compared. Their approach of hypothesis testing ignores the multivariate dependencies of the features. Hence, we use another method to test independence of distributions that is discussed below.

### Hypothesis testing for similarities of network distributions.

As mentioned earlier, the significance tests used by Berlingerio and colleagues (2012) posits multivariate independence among features and lacks what they call “discrimination power”. To tackle this issue, one needs use multivariate dependence tests. Since parametric dependence tests rely on assumption for the distributions being compared, we used HHG nonparametric permutation test of multivariate dependence (Heller, Heller, & Gorfine, 2013) implemented in `HHG` R package (Brill, Heller, & Heller, 2018). HHG is a consistent omnibus test for the null hypothesis that the joint distribution of two multivariate random variables X and Y is equal to the multiplication of the marginal distributions of those variables. Eq 9 shows the null and alternative hypotheses:

EQ 9

HHG has a reasonable computational complexity and uses norm distance matrices of the samples taken from X and Y separately. The technical details of this method are beyond the scope of this paper. In short, HHG iteratively forms hyper spheres in the joint space of and based on the implications of the null hypothesis, quantifies evidence against by likelihood ratio or Pearson’s Chi-square tests statistics over the contingency tables. From these tests, one can drive permutation p-values that can be interpreted as evidence against null hypothesis of the independence of the distributions. Hence, the lower the p-value, the more evidence favoring the dependence of the distributions being compared. Loosely speaking, one can treat the p-values derived from HHG methods as a form of distributional distance; the lower the value, the more “similar” the distributions are to each other. This interpretation (is it an interpretation, or some other term?) is rather unorthodox (?) in hypothesis testing, yet it allows us to compare non-significant p-values as relative measures of resemblance.

The `hhg.test()` function in `HHG` package runs the test for a number of permutations on distance matrices of the samples in X and Y and outputs four different permutation p-values based on sums or maximum values of likelihood ratio or Chi-square test scores of al 2X2 contingency tables (elaborate? Clarify?). In this study, we let HHG run for 2000 permutations for each pairwise comparison and extracted permutation p-value for the maximum of likelihood ratio score statistics as it yielded higher discriminative power compared to other test statistics.

### Family resemblances and differentiations

Both HHG and NetSimile, precisely speaking, provide measures for dissimilarity; higher values of these measures entails smaller resemblance between the networks being compared. The outcomes of pairwise comparison of the networks using these methods were stored in four 50X50 matrices of form for networks N (either M or A) based on method m (either HHG or NetSimile). To ease the parallel interpretation of these measures, the matrices of NetSimile distances, i.e., and , were normalized by the highest value in each matrix so their values range from zero to one. Since it is more intuitive to talk about similarities, the dissimilarity matrices were transformed into similarity matrices by subtracting them from one (Eq 10). Consequently, higher values in the similarity matrices denote higher pairwise resemblance.

Eq 10

The within- and between-family resemblance aggregate scores were calculated by averaging the elements similarity matrices that belong to the families being compared as shown in Equation 11.

Eq 11

Finally, a differentiation score was calculated for each family to quantify the degree to which brains belonging to family f resemble each other and, at the same time, diverge from the members of other families. Equation 12 shows this score is calculated.

Eq 12

The numerator in Equation 12 is the within-family resemblance of networks for family f. The denominator is the mean of between-family resemblance of f to other families. This proportion is subtracted from one so that this score is positive if the family f differentiates from other families and negative otherwise.

### Visualizing family measures

To get a better intuition into how NetSimile and HHG similarity measures diverge, they were put together in new matrices wherein the lower triangle belongs to and the upper triangle belongs to , i.e., . These similarity matrices are plotted in Figure 3 as heatmaps using `ComplexHeatmap` R package (citation). Moreover, the matrices of where also plotted in the same figure.

The differentiation scores of families based on NetSimile and HHG for M and A are plotted in Figure 4. Positive differentiation value for family f means that its within-family resemblance of network structures is higher than the average resemblance of its members to the members of other families.

Finally, in order to have both family resemblance and family differentiation in a single frame, we summarized the values in graphs shown in Figure 5. For each method and network, a graph is built wherein nodes are families of brains. The edge colors and sizes represent between-family resemblances. The node colors capture within-family resemblance scores. The size of each node is proportional to the absolute value of the differentiation score of its corresponding family. The nodes with positive differentiation score are marked with asterisks.

1. The values were chosen in line with (citation), and the range has been narrowed down by exploratory inspection of preliminary outcomes of the simulations. [↑](#footnote-ref-1)
2. Also known as elementwise multiplication of matrices where the corresponding elements of matrices are multiplied. [↑](#footnote-ref-2)
3. While programming the analyses, a matrix of distances was calculated to increase the versatility of the code for other rewiring algorithms. [↑](#footnote-ref-3)
4. Note that, since the nodes are not labeled and are indistinguishable, the order of node-specific values for local measures hold no information. [↑](#footnote-ref-4)
5. Egonet of node i, referred to as ego(i), is the subset of the network including node i, its first-order neighbors (N(i)), and the edges among N(i). [↑](#footnote-ref-6)
6. Since our networks are undirected, this value would be equal to . [↑](#footnote-ref-7)