Modelling the brain with a multilevel Random Geometric Graph

Bachelor Thesis

Ilse Meijer

6125670

Supervisor: Prof. dr. R.H. Bisseling

Mathematics
Utrecht University
15/01/2021

Abstract

The brain is a complex network. Recently multiple studies have tried to model complex networks using graph theory. In this thesis we have developed a multi-level Random Geometric Graph to model the human brain. We have included the extra property that the probability of a connection declines proportionally to the distance. Based on our analysis we conclude that this graph has a similar level of segregation, but a higher level of integration compared to the real brain. This graph also predicts the properties of deeper levels in the brain, where data of these levels is not yet available.

Contents

1 Introduction

We are still far from understanding how the human brain works. We know that certain connections or damage to these connections can influence or change the way the brain functions. To learn more about this, researchers investigate lesioned patients. With new techniques such as functional magnetic resonance imaging (fMRI) we are now also able to make models of the human brain [14]. If we can successfully do this we can change parameters or make lesions in the artificial brain to analyze what happens to the brain under these circumstances. We can use this to understand for example which damages lead to changes in the brain's function. One of the challenges is that the brain consists of many small parts (e.g. the neurons and synapses) that have specific locations [9]. This makes the use of computer memory of an algorithm modelling the brain high. Our goal is to create a model of the brain that uses less computer memory by using random configurations. We will do this by modelling a Random Geometric Graph (RGG) and making some adjustments to make the fit with the brain better. An RGG is a model that generates random points, which we call nodes and connects them if they are within a certain radius of each other [12]. These connections are called edges.

Some of the previous studies that compared an RGG to the brain include that of Smith et al. [16]. In this study they constructed an RGG in a cubic volume. They concluded that there were similar high levels of segregation between the RGG and the brain MRI, meaning that in both networks there is a lot of clustering of nodes. In the article of Carlson, Mucha, Klimm, and Bassett [6] the RGG was created in a rectangular volume. They use a slightly different implementation in which they connect a node to the M closest nodes according to the Euclidean distance. Their results were that the RGG was more segregated than the brain data and that it had a large network diameter and long path length. They also compared the brain to the Distance Drop-Off (DD) model. In the DD-model nodes are located at certain predetermined regions in the brain. Two nodes are then connected with a probability that depends on the distance d between these nodes [6]. In Ajazi's PhD thesis he described a model that had some properties of RGGs and some properties of the DD-model [1]. We also want to take properties from both these models.

The brain consists of multiple levels. Because of this we want to create a multilevel network, so that the brain can be analyzed at the specific level we are interested in. Most of the time the brain is divided in the macroscale, the mesoscale and the microscale. On the macroscale, large areas of the brain are connected through white matter tracts [9]. The brain is separated in two highly connected parts: the left and the right hemisphere. The fiber tract between these two parts is the Corpus Callosum (CC). Each hemisphere can be divided in six parts, that differ in functionality and structure: the frontal lobe, the occipital lobe, the parietal lobe, the temporal lobe, the cerebellum and the limbic system. The mesoscale lies somewhere between the macroscale and the mi-

croscale and is still hard to investigate in humans due to the invasive nature of currently available methods.

On the microscale, the network consists of neurons and synapses [9]. On the finest scale of a brain network, the nodes represent neurons and the edges represent synapses. The brain consists of approximately $86.1 \pm 8.1 \cdot 10^9$ neurons [3]. The number of synapses in the brain is a debated subject and ranges somewhere between 10^{14}–10^{15} synapses [18, 20]. The human brain consists of different kinds of synapses, electrical and chemical synapses. Electrical synapses are faster than chemical synapses. Furthermore electrical synapses are bidirectional, while chemical synapses are directed. These two ways of synaptic transmission often interact with each other [13]. The weights of chemical synapses change over time. On the microscale weight can be defined as a measure of the amount of postsynaptic response after an action potential. This can be excitatory (positive) or inhibitory (negative). It is a function of the number of neurotransmitters fired and the number of active receptors for these neurotransmitters. There exist multiple kinds of neurotransmitters and receptors [22].

With new techniques such as fMRI and diffusion-weighted magnetic resonance imaging (DW-MRI) we can analyse the brain both functionally and anatomically. The brain is functionally segregated, which means that the nodes are functionally clustered. Thanks to this characteristic the brain is able to perform specialized processing. The brain is also functionally integrated, which means that nodes from different parts in the brain are connected. This characteristic makes sure that the brain can combine the specialized information from the segregated brain regions. In this thesis we focus on anatomical brain networks and not on functional brain networks. Anatomical brain networks incorporate the physical connections between brain regions [15]. We will build a network as similar as possible to the human brain based on graph theory, the basics of which will be discussed in the next chapter.

2 Graph theory

In this thesis we limit ourselves to graphs embedded in $\mathbb{R}^3$. A graph $G = (V, E)$ consists of a set of vertices $V(G) \subseteq \mathbb{R}^3$ (also called nodes) and a set of edges $E(G) \subseteq V \times V$, lines that represent connections between two nodes. The number of nodes in a graph is called the order of the graph and is denoted by $|V|$. Let $I = \{1, ..., |V|\}$ be the index set of V and let $i, j, k \in I$. A node is denoted by $x_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in V(G)$. An edge $\{x_i, x_j\} \in E(G)$ is often written as $x_i x_j$. In figure 1 we have drawn an example of a graph that is not connected and we have labeled an edge and a vertex/node. A graph can be represented by

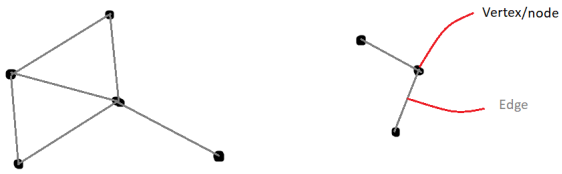

Figure 1: Example of a graph that is not connected.

an adjacency matrix A. We say that $x_i, x_j \in V$ are adjacent if $x_i x_j \in E$. The adjacency matrix A is a $|V| \times |V|$-matrix that stores the weight w_{ij} of an edge between the nodes x_i and x_j. In the case of a weighted graph this means that:

$$A_{ij} = \begin{cases} w_{ij} \in \mathbb{R}_{>0} & \text{if } x_i x_j \in E, \\ 0 & \text{otherwise.} \end{cases}$$

In an unweighted graph $w_{ij} = 1$. We say that $G' = (V', E')$ is a subgraph of G if and only if $V' \subseteq V$ and $E' \subseteq E$. We denote this by $G' \subseteq G$. A graph can be directed or undirected. A directed graph is a graph that has two maps: init: $E \longrightarrow V$ and term: $E \longrightarrow V$, where the edge is directed from init(e) to term(e).
A path is a graph P that connects node x_0 to node x_n. It consists of nodes $\{x_0, x_1, x_2..., x_n\}$ and edges $E = \{x_0x_1, x_1x_2, ..., x_{n-1}x_n\}$ [7]. Two nodes $x_i, x_j \in V$ are connected if there exists a path between x_i and x_j.
A graph can consist of a single connected component, which means that for every $x_i, x_j \in V$ there exists a path P such that x_i is connected to x_j. The brain consists of such a connected component.
The degree k_i of a node x_i is defined as the sum of the weights of the edges connected to x_i [15]:

$$k_i = \sum_{j \in |V|} w_{ij}.$$

This equation holds for both unweighted graphs, where $w_{ij} \in \{0, 1\}$, and weighted graphs, where $w_{ij} \in [0, 1]$. The (weighted) geometric mean of tri-

angles t_i that include node x_i is defined as:

$$t_i = \frac{1}{2} \sum_{j,h \in |V|} (w'_{ij} w'_{ih} w'_{jh})^{\frac{1}{3}},$$

where $w' = \frac{w}{\max_{i,j \in |V|} w_{ij}}$ [2]. Note that the factor $\frac{1}{2}$ is needed because each pair of triangles is counted twice. Consider for example node $i = 1$, that forms a triangle with node 2 and 3. This triangle is counted when $j = 2$ and $h = 3$, but also when $j = 3$ and $h = 2$.

Our goal is to compare the human brain to a random graph. There are several random graphs, one of which is the Erdös-Rényi random graph. This graph takes a random configuration of nodes V and connects two nodes with probability p. Thus for $c_{ij} \in [0, 1]$ randomly chosen

$$A_{ij} = \begin{cases} 1 & \text{if } c_{ij} \leq p \text{ and } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

Another type of graph is the geometric graph. This is an undirected graph $G(V; r)$ where V is the vertex set and where $x_i x_j \in E(G)$ when $|x_j - x_i| \leq r$. In order to determine the distance between two nodes $x_i, x_j \in V(G)$ we use the Euclidean norm:

$$|x_i - x_j| = \sqrt{(x_{i_1} - x_{j_1})^2 + (x_{i_2} - x_{j_2})^2 + (x_{i_3} - x_{j_3})^2}.$$

A geometric graph based on random point configurations is called a Random Geometric Graph (RGG).

We use three measures to test our network. First the number of connected components: the brain consists only of one component, so the model should be connected. The connectivity threshold of an RGG can be determined based on theorem 1, which is a special case of theorem 6(ii) in [8]. In this theorem we take $p = 2$, since we use the Euclidean norm ($l_p = l_2$) and $d = 3$, since we consider graphs in $\mathbb{R}^3$. We then obtain the following theorem:

Theorem 1. Let G be an RGG in $\mathbb{R}^3$ using the Euclidean norm. Suppose that $r = \left(c \cdot \frac{\ln n}{n}\right)^{\frac{1}{3}}$ and $c > \frac{4}{3}$. Then, almost always, G is connected.

The second characteristic of the brain is that it has functional segregation. A measure for this is the clustering coefficient, defined below, which should be high (see Figure 2). Another characteristic of the human brain is its functional integration. This can be measured by the global efficiency, which is the inverse of the average shortest path length. Those two properties combined make the anatomical brain network a small-world network [15]. The clustering coefficient C is calculated based on the geometric mean of triangles t_i and the degree of a node k_i. It is defined as:

$$C = \frac{1}{n} \sum_{i \in |V|} \frac{2 \cdot t_i}{k_i(k_i - 1)}. \tag{1}$$

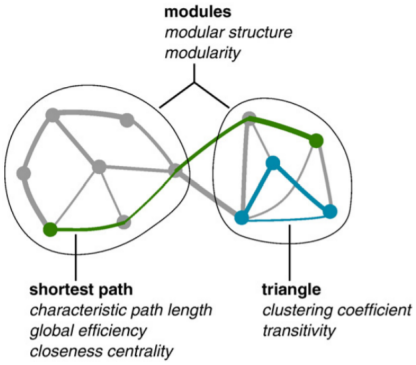

Figure 2: Clustering coefficient and global efficiency [15].

If d_{ij} is the shortest path length between x_i and x_j then the global efficiency E is given by:

$$E = \frac{1}{n} \sum_{i \in |V|} \frac{\sum_{j \in |V|} d_{ij}^{-1}}{n-1}. \tag{2}$$

The benefit of an RGG compared to e.g. Erdös-Rényi random graphs is the triangle property of RGGs. Given three points $x_i, x_j, x_k \in V$ with $|x_i - x_j| \leq r$ and $|x_i - x_k| \leq r$, then $|x_j - x_k| = |x_j - x_i + x_i - x_k| \leq |x_j - x_i| + |x_i - x_k| = |x_i - x_j| + |x_i - x_k| \leq 2r$. So if two points are connected to the same point, there is a higher chance that those two points are connected too. This means that the fraction of triangles around a node will be high, which implies that the clustering coefficient will be high too.

3 Brain Network Generator

The brain network generator we developed depends on seven parameters as described in table 1. For the first level, we generate n random nodes in the unit

Parameter	Meaning
n	the number of nodes
l	the number of levels
b	the branching factor
t	the threshold
r	the cutoff radius
p	the probability of a connection
F_w	the weight distribution

Table 1: The parameters of the model.

sphere, the coordinates of which are stored in the matrix 'RandomNodes'. We then create the empty $n \times n$-adjacency matrix 'weights'. Next we will change each value in this matrix to the corresponding weight of this connection.

To find out if two nodes x_i and x_j are connected we first determine the distance d between these two nodes with the Euclidean norm. We then compare this distance to the cutoff radius r.
Suppose $d < r$, then the value of the matrix 'weights' at position i, j changes according to the weight distribution.
Suppose now that $d \geq r$. If a random number c_0 is smaller than p, the value of the matrix 'weights' at position i, j changes according to the weight distribution. If however $c_0 \geq p$, the value of the matrix 'weights' at position i, j becomes 0. In both cases we check whether the value of the matrix 'weights' at position i, j is smaller than the threshold t. If this is the case, the value is changed to zero. We do this to reduce noise.

If $l > 1$, the algorithm creates bn new nodes for each new level, since every node x_i in RandomNodes is replaced by b new nodes whose coordinates lay in the sphere of radius r around x_i. After replacement of the old nodes by the new ones, the number of nodes n is higher. This causes the cutoff radius r to change too, since we take $r = \left(c \cdot \frac{\ln n}{n}\right)^{\frac{1}{3}}$ and $c > \frac{4}{3}$. The algorithm now connects the nodes in the same way as described for the first level. This will be repeated until there are l levels.

We visualize this network with the python package NetworkX. This reads the adjacency matrix and the coordinates of the network and prints it in 2D.

In one of our first trials we implemented spatial hashing to make our algorithm faster, because we still looked at the modelling of the brain as a hard-spheres problem (see the first panel in figure 3). This means that there can't be a con-

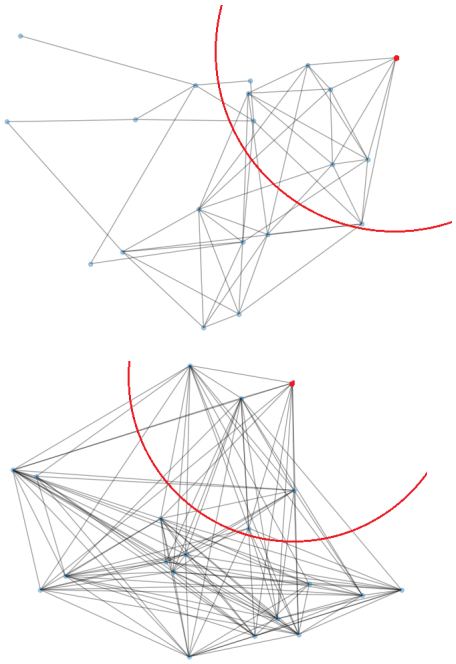

Figure 3: Difference soft-spheres problem (bottom) and hard-spheres problem (top).

nection when the distance between two nodes is larger than the cutoff radius. Spatial hashing could make such a problem faster, as it would only consider nodes that are in neighbouring bins, thereby lowering the number of times the distance between two nodes should be calculated. Later we realised that the brain is better comparable to a soft-sphere problem, since there also exist long connections (see the second panel in figure 3). In soft-sphere problems, the probability of a connection depends on the distance. Therefore spatial hashing would not benefit our algorithm.

4 Analysis

Diffusion tensor imaging (DTI) is the preferred method to study white matter [21]. One could also use voxel-based morphometry, which estimates the amount of tissue at a specific point. This estimation is given in voxels and appears to be more specific than DTI. However the sensitivity of morphometry for white matter is limited, since there only exist small changes in intensity in white matter regions [11]. For the analysis of our data we use the USC Multimodal Connectivity Database [5]. This database provided by the MGH/UCLA Human Connectome Project has stored the connectivity matrices of brains using fMRI, DW-MRI and DTI.

We use the NKI/Rockland data (id 1759), in which they averaged the connectivity matrices of 196 individuals in the age range of $6 - 89$. The data is measured using the Siemens Trio 3T MRI scanner. In the data the diffusion tensor is estimated with the diffusion toolbox and the fiber assignment by continuous tracking (FACT) algorithm is used to do the tractography. The brain is partitioned into 188 regions of interest (ROIs). The data is thresholded to make sure each voxel was assigned to only one ROI, the one for which the likelihood of membership is the highest. The number of fibers that have at least one voxel in both the source and the target ROI are determined and summarized in the structural connectivity matrix [5].

As mentioned before, our code has multiple parameters (see table 1). We first base our choice for the values of these parameters on the literature. In the results section we will vary one of these parameters and keep the rest constant to explore whether this gives improvements of the properties of the RGG.

We choose

$$r = \left(c \cdot \frac{n}{\ln n}\right)^{\frac{1}{3}}$$

and vary the value of $c > \frac{4}{3}$, such that G is almost always connected. We choose $n = 188$, since the data we use to compare our algorithm to also has 188 nodes. As mentioned before the brain consists of approximately $86.1 \pm 8.1 \cdot 10^9$ neurons. At this moment it is not (yet) possible to model this number of nodes and the even higher number of edges that come with these nodes. If this were possible there would be a dependent relationship between the branching factor b and the number of levels l, since

$$8.61 \cdot 10^{10} = n \cdot b^{l-1},$$

for $l > 1$. If we assume that $n = 188$ and rewrite this equation we obtain:

$$b = \sqrt[l-1]{\frac{8.61 \cdot 10^{10}}{n}} = \sqrt[l-1]{4.58 \cdot 10^8},$$

for $l > 1$. To make this clearer we have calculated the relation between b and l for some values in table 2. However for now, it is not yet possible to calculate

Number of levels (l)	Branching factor (b)
2	$2.14 \cdot 10^4$
3	$7.71 \cdot 10^2$
4	$1.46 \cdot 10^2$
8	12

Table 2: The relation between b and l.

all the levels with these branching factors, so we consider only the first couple of levels.

To determine the weight distribution we have written the code weights.py, in which we use the connectivity matrix of the NKI/Rockland-dataset. Since the weight w_{ij} of an edge in this connectivity matrix represents the number of voxels in both nodes (ROIs), we know that $w_{ij} \in \mathbb{N}$. If there exists a connection we add the weight of this connection to an array, where we take into account that $w_{ij} = w_{ji}$, which we only add once. We then count the number of times each weight exists in the whole matrix. We plot the weights on the x-axis, the occurrence of each weight on the y-axis and draw a line through these points (see the first panel of figure 4). This seems to be a power law. In order to further analyse this distribution we plotted the same data on a logarithmic scale (see the second panel of figure 4), which should follow a linear distribution for a power law. Since this is approximately the case we fitted a power law on our data and obtained the equation:

$$f(w) = 67.84 \cdot w^{-0.81}.$$

The resulting graph is shown in the third panel of figure 4. We base our weight distribution on this power law.

To estimate the probability p of a connection when the distance is larger than the cutoff radius, we wanted to find a distribution of the fiber lengths. Using post-mortem histology it was determined that the total length of nerve fibers is 118000 km and that the length density is 249 m/mm^3 [19]. In the preceding literature study we have not been able to find information about the precise distribution of the length of nerve fibers. Therefore we assume this distribution to decline proportionally to the distance d, where the chance of a connection should be 1, if $d = r$ and should be approaching 0, when $d = 2$, since we consider our graph in the unit sphere. We will consider the candidate probabilities $p_0 = \frac{2-d}{2-r}$, $p_1 = e^{r-d}$, $p_2 = \sqrt{\frac{2-d}{2-r}}$ and $p_3 = \ln(3-d)$.

The threshold t is used to reduce noise. We should thus choose it in the interval $[0, 1]$.

The properties we use to analyse our network are the clustering coefficient C (1), the global efficiency E (2) and the number of connected components. By analysing the NKI/Rockland data we can find the values that these properties should approach. The clustering coefficient should be approximately 0.6199, the global efficiency 0.6415 and there should be one connected component.

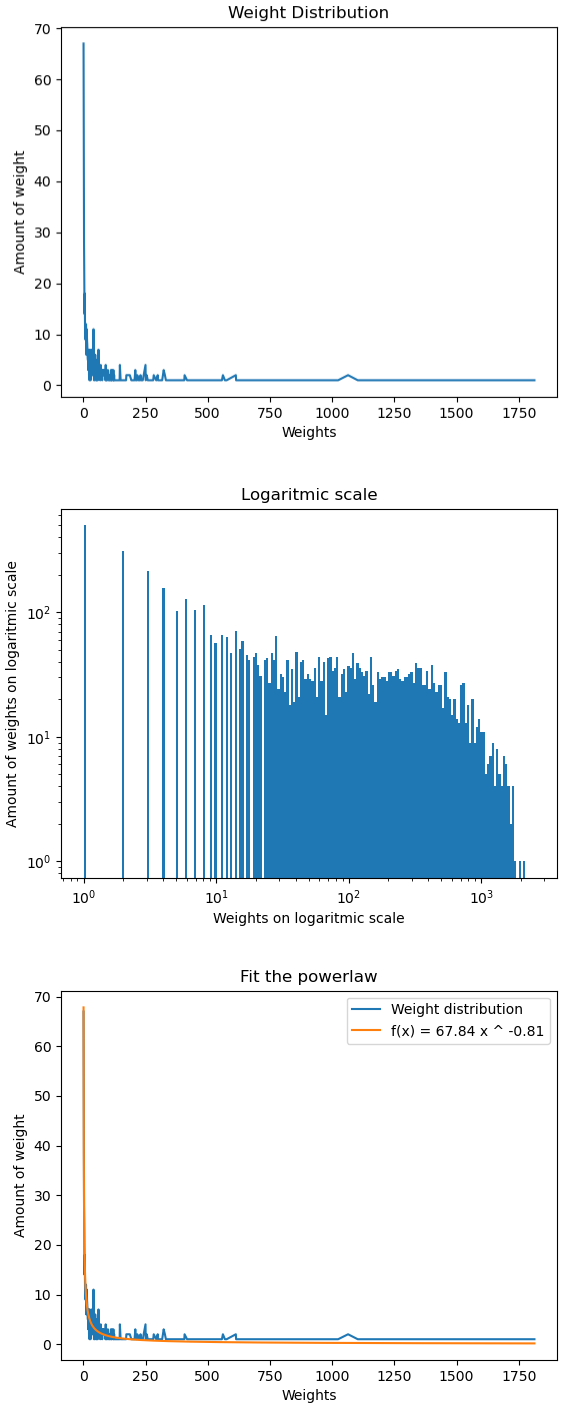

Figure 4: Weight distribution

5 Results

We test our model based on the difference between the properties of the NKI/ Rockland data and the properties of our data. Specifically we will consider the difference in clustering coefficient (ΔC) and in the global efficiency (ΔE). We combine these results by taking the geometric mean ($D = \sqrt{\Delta C \cdot \Delta E}$), see table 4. An advantage of the geometric mean compared to the arithmetic mean is that it takes into account different scales. It is used for instance to compare results of different methods in parallel computing for problem instances of widely different scales [4].

For now, we consider the case where the number of levels $l = 1$. The data we use for comparison only has one level, so we use this data to determine the best values for p_k, t and c. When those values are determined, we can make predictions about lower levels in the brain, which can be tested when more detailed data becomes available.
We could not predict the optimal values for p_k based on the literature. Therefore we have tried several values for t, c and n, for every value of k. In every case the value $p_k = p_2$ had the lowest D and thus we take $p_k = p_2$ as default.
For the threshold t, we could only determine that $t \in [0, 1]$. To determine the best value of t, we considered $t \in \{0.01, 0.05, 0.1, 0.5, 0.9\}$. From these values $t = 0.05$ had the lowest value of D. We checked if $\exists t \in (0.01, 0.05) \cup (0.05, 0.1)$, such that D is even lower. This turned out to be the case for $t = 0.03$, which is why we chose this as default value.
For the cutoff-radius, we know that $c > \frac{4}{3}$. We thus tried $c \in \{2, 5, 6, 7, 9\}$ and noticed the lowest value of D for $c = 6$, hence we chose this as the default value for c. In figure 5 we have plotted our model for the default values.

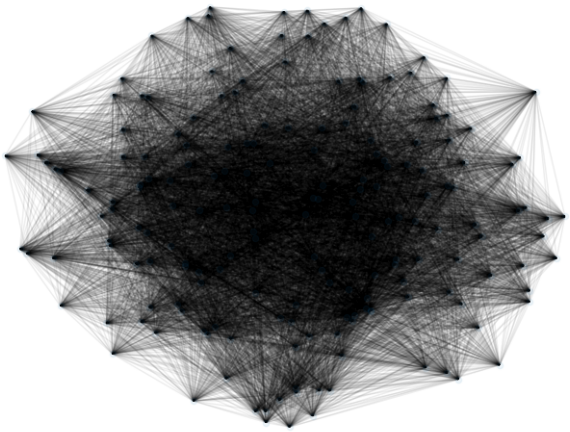

Figure 5: Model with default values $n = 188$, $t = 0.03$, $c = 6$ and $p_k = p_2$.

In table 3 we have summarized the values of the clustering coefficient (C), the global efficiency (E) and the number of connected components (m), the latter as expected equals one, since $c > \frac{4}{3}$ for different values of the variables. Each time we have varied one variable and kept the remaining variables as the default value.

Variables	C	E	m
Default	0.62	0.81	1
$t = 0.01$	0.65	0.82	1
$t = 0.1$	0.57	0.78	1
$c = 2$	0.59	0.79	1
$c = 9$	0.63	0.81	1
$p_k = p_0$	0.53	0.75	1
$p_k = p_1$	0.51	0.75	1
$p_k = p_3$	0.60	0.79	1

Table 3: Default: $n = 188$, $t = 0.03$, $c = 6$ and $p_k = p_2$. The columns represent the variables of the model that are not the default value, the clustering coefficient (C), the global efficiency (E) and the number of connected components (m).

Variables	ΔC	ΔE	Geometric mean D
Default	0.00	0.16	0.015
$t = 0.01$	0.03	0.18	0.079
$t = 0.1$	0.05	0.14	0.081
$c = 2$	0.03	0.15	0.068
$c = 9$	0.01	0.17	0.050
$p_k = p_0$	0.09	0.11	0.099
$p_k = p_1$	0.11	0.11	0.11
$p_k = p_3$	0.02	0.15	0.048

Table 4: Default: $n = 188$, $t = 0.03$, $c = 6$ and $p_k = p_2$. The columns represent the variables of the model that are not the default value and the difference between our data and the NKI-dataset. Specifically the difference in clustering coefficient (ΔC), the global efficiency (ΔE). These differences are combined in the geometric mean D.

In table 4 we have summarized the differences in clustering coefficient, global efficiency and the average difference. These values are based on table 3 and the NKI/Rockland data ($C = 0.6199$ and $E = 0.6415$).

We can use these default values to make an estimation about deeper levels in the brain. Suppose we divide the brain into 8 levels, then, according to table 2, $b = 12$. Since we have a multilevel graph, we can analyse multiple levels of the brain. We analyse the highest two levels for $b = 12$ and thus $n = 12$ for level 1 and $n = 144$ for level 2. According to our model, the clustering coefficient C

and the global efficiency E of the second level is: $C = 0.59$ and $E = 0.79$. The geometric mean $D = 0.091$, which is still not that large. At the moment, to the best of our knowledge, there is no data available to further test this data.

6 Discussion

From table 4 we can see that the chosen values are in a (local) optimum. Thus we take $n = 188$, $t = 0.03$, $p_k = p_2$ and $c = 6$ to be the optimal values for our model. This means that the optimal values are $C = 0.62$ for the clustering coefficient, which is extremely close to the clustering coefficient of the real brain, and $E = 0.81$ for the global efficiency, which is larger than the global efficiency of the real brain. From this we can conclude that the segregation is the same and the integration of the model is higher than the integration of the brain.
This analysis is not perfect though. First of all because there is no data available about the deeper levels of the brain (over 200 nodes), which makes it impossible to test all the features of our model. Unfortunately it is not yet possible to run this program for the deepest levels or for a large number of nodes either, because of the long runtime and large amount of memory needed.
Second because of the data we used, the DT-imaging. This data can contain noise. Another disadvantage of this data is that the streamlines (paths between nodes) are not quantitative, which makes it hard to define the weights. This can result in larger weights, when the number of streamlines is higher. This can also lead to low weights of long connections, because it is more difficult to recover the path of a long white matter tract than the path of a short tract [10]. Follow-up research could reduce this problem reduced by using the SIFT2-technique [17].
Another problem with the lack of a definition for weights in the brain, is that different datasets consider different definitions and therefore obtain different connectivity matrices. In our model we chose a dataset where weights are determined by the number of tracts that have voxels in both regions of interest. Suppose we chose a different dataset to compare our data with, which has a different definition of weight. Then we would have gotten a different weight distribution and thus a different model. In future research it could be investigated if a different definition for weights would give us a better model.
It is also possible to compare our model to data from voxel-based morphometry. This can give a slightly more detailed representation of the brain, but since we analyse the white matter of the brain and voxel-based morphometry is less sensitive to white matter, this choice is not optimal either. In next studies it could be possible to compare our model to both DT-imaging and voxel-based morphometry.
Another uncertain factor is that we could not find information about the distribution of the length of tracts, so we had to make an estimation based on the outcomes of our model. If this distribution becomes available the model could be made more accurate.
We based our choice for the variables on the geometric mean of the difference in the clustering coefficient C and the global efficiency E. We could also have taken into account whether the difference in clustering coefficient was approximately as large as the difference in global efficiency. We now have a model where the clustering coefficient is the same, but the global efficiency differs a lot from the real brain. This would not have been the optimal solution when taking into account the ratio of both differences.

7 Conclusion

First we have looked into basic properties of and methods to investigate the brain. We have analysed the basic definitions of graph theory, that we later used to create our model and analyse the optimal values of our model. We have created a weighted multilevel Random Geometric Graph, to which we have added the extra property that the probability of a connection between two nodes depends on the distance, when the distance between these nodes is larger than the cutoff radius. We made the simplification in the multilevel property that the branching factor is uniform. According to our analysis the optimal values to compare this graph to the brain are $n = 188$, $p_k = p_2 = \sqrt{\frac{2-d}{2-r}}$, $r_c = r_6 = \left(6 \cdot \frac{n}{\ln n}\right)^{\frac{1}{3}}$, $t = 0.03$ and $b = \sqrt[l-1]{4.58 \cdot 10^8}$. This gave us the values $C = 0.62$ for the clustering coefficient and $E = 0.81$ for the global efficiency. Thus the segregation in the brain is modelled accurately, but the integration is of the model is higher than the integration in the brain.

A Python Code of our Brain Generator

```
import numpy
import math
import networkx as nx
import matplotlib.pyplot as plt
import random as rnd
import decimal as dec

def Probability(d, r, k): #candidate probability distributions
    if k == 0:
      return (2− math.sqrt(d))/(2 − r)
    if k == 1:
      return math.exp(r− math.sqrt(d))
    if k == 2:
      return math.sqrt((2− math.sqrt(d))/(2 − r))
    if k == 3:
      return math.log(3− math.sqrt(d))

def BrainGenerator(n, l, b, t, c, k): #nodes n, levels l, branching factor b, threshold t,
    constant factor cutoff radius c, probability distribution k
    RandomNodes = numpy.empty([n, 3])
    for level in range(l):
        if level == 0:
          i = 0
          while (i < n): #create n random nodes in the unit sphere at level 1
                node = (rnd.uniform(−1, 1), rnd.uniform(−1, 1), rnd.uniform(−1, 1))
                if node[0]**2 + node[1]**2 + node[2]**2 <= 1:
                  RandomNodes[i] = node
                  i+ = 1
        else: #create n random nodes in the unit sphere at the higher levels
          size = numpy.size(RandomNodes, 0)
          n = b*size
          RandomNodes = numpy.empty([n, 3])
          for j in range(size):
              i = 0
              while (i < b):
                    node = (dec.Decimal(rnd.uniform(RandomNodes[j, 0] −r,
                          RandomNodes[j, 0] +r)), dec.Decimal(rnd.uniform(
                          RandomNodes[j, 1] −r, RandomNodes[j, 1] +r)),
                          dec.Decimal(rnd.uniform(RandomNodes[j, 2] −r,
                          RandomNodes[j, 2] +r)))
                    if(node[0]**2) + (node[1]**2) + (node[2]**2) <= 1:
                      RandomNodes[i + b ∗ j] = node
                      i+ = 1
```

```
r = (c*math.log(n)/(n))**(1/3) #cutoff radius
weights = numpy.empty([n, n])
for j in range(n):
    i = 0
    while (i <= j): #make connections between nodes
        dsquared = (float) (RandomNodes[i, 0] - RandomNodes[j, 0]) ** 2+
                   (RandomNodes[i, 1] - RandomNodes[j, 1]) ** 2+
                   (RandomNodes[i, 2] − RandomNodes[j, 2]) ** 2
        if dsquared < r ** 2 and i ! = j:
           weights[i, j] = weights[j, i] = 67* numpy.rnd.power(0.19)
           if weights[i, j] < t:
              weights[i, j] = weights[j, i] = 0
           i + = 1
        elif dsquared >= r ** 2 and i ! = j:
           p = Probability(dsquared, r, k)
           c = rnd.uniform(0, 1)
           if c < p and i ! = j:
              weights[i, j] = weights[j, i] = 67*numpy.rnd.power(0.19)
              if weights[i, j] < t:
                 weights[i, j] = weights[j, i] = 0
           else:
              weights[i, j] = weights[j, i] = 0
              i + = 1
        else:
           weights[i, j] = 0
           i + = 1

G = nx.from_numpy_matrix(weights)
nx.draw (G, pos = {i: (RandomNodes[i, 0], RandomNodes[i, 1]) for i in
        range(len(RandomNodes))}, node_size = 20, alpha = 0.07)
plt.show()
```

References

[1] F. Ajazi. *Random geometric graphs and their applications in neuronal modelling*. PhD thesis, Lund: Lund University, Faculty of Science, Centre for Mathematical Sciences, 2018.

[2] D. A. S. Aric A. Hagberg and P. J. Swart. Exploring network structure, dynamics, and function using networkx. *Proceedings of the 7th Python in Science Conference*, page 11–15, 2008.

[3] F. A. C. Azevedo, L. R. B. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. L. Ferretti, R. E. P. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513(5):532–541, 2009.

[4] R. H. Bisseling. *Parallel Scientific Computation: A Structured Approach Using BSP*. Oxford University Press, 2020.

[5] J. Brown, J. Rudie, A. Bandrowski, J. Van Horn, and S. Bookheimer. The UCLA multimodal connectivity database: a web-based platform for brain connectivity matrix sharing and analysis. *Frontiers in Neuroinformatics*, 6:28, 2012.

[6] J. M. Carlson, P. J. Mucha, F. Klimm, and D. S. Bassett. Resolving structural variability in network models and the brain. *PLOS Computational Biology*, 10(3), 2014.

[7] R. Diestel. *Graph Theory*. Springer Nature, 5th edition, 2017.

[8] R. B. Ellis, J. L. Martin, and C. Yan. Random geometric graph diameter in the unit ball. *Algorithmica*, 47:421–438, 2007.

[9] A. Fornito, A. Zalesky, and E. T. Bullmore. Chapter 1 - An introduction to brain networks. In A. Fornito, A. Zalesky, and E. T. Bullmore, editors, *Fundamentals of Brain Network Analysis*, pages 1–35. Academic Press, San Diego, 2016.

[10] D. Jones. Challenges and limitations of quantifying brain connectivity in vivo with diffusion MRI. *Imaging in Medicine*, 2:341–355, 2010.

[11] F. Kurth, E. Luders, and C. Gaser. Voxel-based morphometry. In A. W. Toga, editor, *Brain Mapping*, pages 345–349. Academic Press, Waltham, 2015.

[12] M. Penrose. *Random Geometric Graphs*. Oxford University Press, 2007.

[13] A. E. Pereda. Electrical synapses and their functional interactions with chemical synapses. *Nature Reviews Neuroscience*, 15:250–263, 2014.

[14] C. Rorden and H.-O. Karnath. Using human brain lesions to infer function: a relic from a past era in the fMRI age? *Nature Reviews Neuroscience*, 5:812–819, 2004.

[15] M. Rubinov and O. Sporns. Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059–1069, 2010.

[16] K. Smith, M. E. Bastin, S. R. Cox, M. C. Valdés Hernández, S. Wiseman, J. Escudero, and C. Sudlow. Hierarchical complexity of the adult human structural connectome. *NeuroImage*, 191:205–215, 2019.

[17] R. E. Smith, J.-D. Tournier, F. Calamante, and A. Connelly. SIFT2: enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography. *NeuroImage*, 119:338–351, 2015.

[18] O. Sporns, G. Tononi, and R. Kötter. The human connectome: A structural description of the human brain. *PLoS Comput Biol*, 1(4), 2005.

[19] Y. Tang and J. R. Nyengaard. A stereological method for estimating the total length and size of myelin fibers in human brain white matter. *Journal of Neuroscience Methods*, 73(2):193–200, 1997.

[20] Y. Tang, J. R. Nyengaard, D. M. De Groot, and H. J. Gundersen. Total regional and global number of synapses in the human brain neocortex. *Synapse*, 41(3):258–271, 2001.

[21] W. G. Webb. 3 - Organization of the nervous system II. In W. G. Webb, editor, *Neurology for the Speech-Language Pathologist*, pages 44–73. Mosby, 6th edition, 2017.

[22] R. Wells. Synaptic weight modulation and adaptation part I: Introduction and presynaptic mechanisms. 2003.