Graph Analysis For Simulated Neural Networks With STDP

Snigdha Singh

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Computer Science and Software Engineering

University of Washington

2021

Committee:

Michael Stiber

Erika Parsons

Wooyoung Kim

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Graph Analysis For Simulated Neural Networks With STDP

Snigdha Singh

Chair of the Supervisory Committee:
Professor Michael Stiber
Computing and Software Systems

Many real-world systems can be represented as networks and studied using graph theory. Brain graphs are widely used to analyze brain connectomes using graph theory. Electrophysiological data, tract-tracing, and MRI data have been used to extract functional brain graphs. This study analyzes the properties of brain graphs generated using a neural network simulator. Using a simulator solves the problems related to pre-processing, data acquisition, and length of time series which exist in extracting brain graphs using other data collection methods. Synaptic plasticity is an important part of the functioning and growth of a neural network, and spike-time-dependent plasticity (STDP) has emerged as one of the most widely used plasticity mechanisms due to its physiological realistic induction and evidence of its presence in vivo. This thesis presents the graphical analysis for a spatiotemporal neural dataset and compares the properties of the connectome with a random graph model of similar size. We implement different STDP algorithms, use STDP to refine a simulation equivalent to neuronal growth for 28 days in vitro, and analyze the effect of STDP on the network's connections and structural properties.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my respected advisor, Professor Michael Stiber, for his constant support and guidance throughout my research. Without his persistent support, guidance, incredible patience, immense knowledge, and insightful feedback, this thesis would not have been possible.

I would like to thank my committee members, Professor Wooyoung Kim and Professor Erika Parsons, for their insightful feedback and time. In addition, I would like to extend my gratitude to the members of the UW Bothell Biocomputing Lab (BCL) for their helpful weekly discussions and moral support throughout my endeavor. I would also like to acknowledge Sinchai Delong and Maxwell Wenger for their timely technical support. I am grateful to the CSS department for providing financial support though Gradate Student Teaching Assistantship. My warmest thanks and admiration to my supervisor, Professor Johnny Lin, for sharing his knowledge and making me a better teacher.

Lastly, I am very thankful to my family and friends for encouraging me and believing in me even when I didn't.

DEDICATION

to my grandmothers, Baldev and Krishna, who inspire me everyday

Chapter 1

INTRODUCTION

Growth and refinement are the two major phases of neuronal development. Synaptic plasticity is an important part of the functioning and growth of a neural network, and is used for tuning neural networks in the refinement phase. Spike-time-dependent plasticity (STDP) has emerged as one of the most widely used plasticity mechanisms due to its physiological realistic induction and evidence of its presence *in vivo*. During the refinement phase, connections are adjusted and removed [6]. In the last few decades, graph theory methods have been used to study brains [1, 21, 22], and can be used to analyze the changes in neuron connections and pathways after refinement phase.

Graph analysis of the human connectome has become an increasingly popular way to analyze neuroimaging data. Brain graphs represent the brain as a set of nodes and edges connecting them where nodes denote anatomical regions, and edges denote structural and functional connections [22]. Brain graphs have been used to discover biomarkers for predicting and classifying neurological disorders [7, 10, 47]. Structural hubs, which are neurons that play a central role in the network connectivity and brain function, are present in the human brain from birth [73]. Recognizing and studying the impact of hubs on brain graphs has been getting a lot of attention by neuroscientists due to the critical role of these nodes in cognitive functions of the brain [33, 73].

Functional magnetic resonance imaging (fMRI) [2, 3, 76], electroencephalography (EEG) [70], and magnetoencephalography (MEG) [8, 10] have been used by network neuroscientists to construct functional brain graphs [22]. Using these methods to construct brain graphs have various challenges involved including developing methods to measure nodes and edges, comparing graphs, data acquisition and deciding the appropriate length of time series [1, 22].

Neural simulators have been used by researchers to generate large-scale neural networks; simulation results can be accessed easily and don't have the problems associated with data collection from the human brain [15, 56]. Due to this data from neural simulators can be used to generate brain graphs, and analyze the connectivity of the brain to better understand cognitive functions in the nervous system. Most simulators implement either the growth phase or the refinement phase sepa-

rately. In this study, we attempted to implement the refinement phase using STDP after generating a neural network using a cortical growth model.

We used the BrainGrid neural simulator to produce a network of 10,000 neurons [60]. Previous work on BrainGrid includes analysis of spatiotemporal bursting behavior, prediction of bursts using pre-burst and burst precursors using machine learning algorithms, serialization-deserialization of the network data, and implementation of STDP mechanisms [50, 60, 62]. The connectivity of the neural network generated using BrainGrid has not been studied in detail Hsu [50]. This study aims to bridge that gap and focus on the structural and functional connectivity of the brain graph constructed from the simulation results. In this thesis, we serialized the results from a growth simulation and deserialized them to refine the network using STDP. We studied the spatiotemporal neural dataset as a brain graph, analyze the effect of STDP on the network, and recognize hubs in the network. The brain graph generated using the growth simulation is an undirected weighted graph with similar connection pattern throughout the network. Tuning the network using STDP leads to drastic changes in the architecture of the network, and the graph after refinement is a directed graph with an irregular connection pattern.

This thesis is part of the BrainGrid project, under the direction of Dr. Michael Stiber at the University of Washington Bothell. This thesis makes the following major contributions:

- We implemented different types of STDP models
- We simulated the refinement phase in neuronal development after the growth model using STDP
- We generated brain graphs for the network after the growth simulation and for the network after tuning by STDP, and analyzed the changes that happen due to STDP

Chapter 2

BACKGROUND: NEUROSCIENCE BASICS

The following sections in this chapter provide a brief introduction to the neuroscience basics required to have a better understanding of the research presented in this thesis.

2.1 The Nervous System

The nervous system is a complex system which transmits signals and information through neurons to different parts of the body. It is the controlling, regulatory, communicating and learning system of the body. Sensory receptors detect change, called *stimuli*, inside and outside our body and transit this information as electrical signals to the brain. Based on the sensory input, decisions are made and the brain sends back signal to our body indicating which actions to take. The nervous system has two types of cells: *nerve cells or neurons* and *glial cells or glia*.

The nervous system consists of two components: the *central nervous system* (CNS) and the *peripheral nervous system* (PNS). The brain and the spinal cord make the CNS, they work as a control center for the body; they process the incoming signals and send outgoing signals to do the necessary tasks. The PNS includes a large network of nerves and ganglia. These act as the receptors and transfer signals throughout the body. Neurons can be classified into three types based on their role in the body: sensory neurons that take external stimuli and send signals to the brain, motor neurons that receive signals from the CNS and transfer commands to different body parts, and interneurons which act as connections between neurons in the CNS.

2.2 Neurons and Synapses

A neuron can be divided into four main parts as we can see in figure 2.1 : *soma*, *dentrites*, *axon*, and *presynaptic terminals*. Soma is the cell body which contains the nucleus, which has the genes of the cell, endoplastic reticulum, and cell proteins. The cell body extends to dendrites and axon [58]. Dendrites are branch like structures used to receive signals [58]. Multiple signals can be intercepted at the same time due to the various dendrite branches. The soma determines whether an output

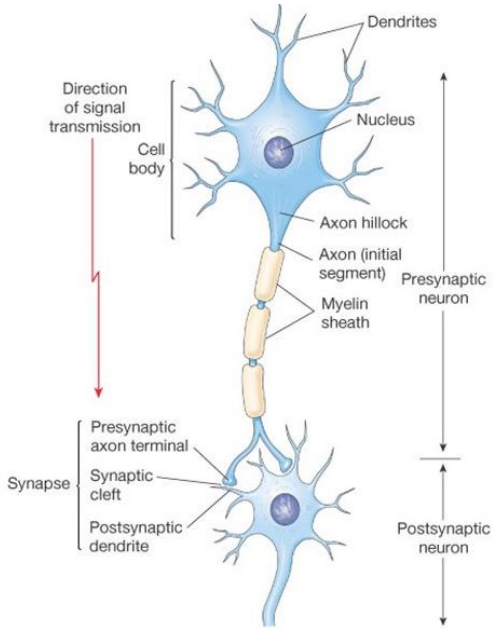


Figure 2.1: The structure of a neuron which are the basic functional unit of the nervous system from Mandira [66]. The main parts of a neuron are soma, axon, dendrites and presynaptic terminals. Dendrites receive incoming signals. The axon is responsible for transmitting outgoing signals. Soma or the cell body is the metabolic center of the cell and contains the nucleus.

signal should be sent based on the input signals received by dendrites. The output signal is passed through the axon which is long tubular part responsible for sending signals to other neurons through presynaptic terminals. The branches of an axon can send the signal to multiple neurons. Signals can be transmitted to over 1000 postsynaptic neurons [58].

The synapse is the point that connects two neurons between which signals are being transmitted. The sending neuron is called the *presynaptic or source neuron* and the receiving neuron is called *postsynaptic or destination neuron*. The direction of the signal transmission is from the presynaptic neuron to postsynaptic neuron.

2.3 Spikes

The signals that are transmitted between neurons and are used in communication are called *spikes*. A spike is generated if the neuron's *membrane potential*, which is the difference in the electrical potential between the inside and outside of the cell, exceeds a certain limit called threshold. Typically

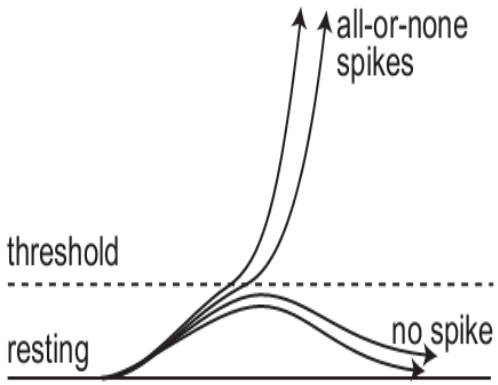


Figure 2.2: The structure of a neuron which are the basic functional unit of the nervous system from Mandira [66]. Schematic representation of firing neurons based on the all-or-none law, diagram taken from Gürcan [42]. A spike is generated only when the membrane potential reaches the threshold value.

the membrane potential is about -65 mV at rest [58]. The membrane potential of a cell can change based on incoming signals. There are two kinds of input signals: *excitatory input signals* which cause an increase or positive change in the action potential, and *inhibitory input signals* which cause a decrease or negative change in the action potential. An increase in the membrane potential causes the neuron to be more likely to spike and a decrease in the membrane potential causes the neuron to be less likely to spike. When the summation of the input signals causes the membrane potential reaches above the threshold, the neuron sends out an action potential or fires a spike. *Synaptic strength* is the amount of impact a neuron has on another when a spike is produced.

After a spike is fired, the membrane potential drops from the resting potential and the neuron is unable to fire any more spikes for a small period of time. This period of time is called *refractory period*. The membrane potential gradually recover to the resting state and then the neuron can fire another spike. An increase in the membrane potential does not necessarily cause a spike. A spike obeys the *all-or-none law* which means that a neuron does not fire a spike of reduced potential if the membrane potential does not reach the threshold and it fires a full spike when the threshold is generated (Figure 2.2).

Due to the stereotypical nature of the membrane voltage magnitude and waveform, they are often considered to carry minimal information. It is considered that this information is encoded with the

frequency of spikes or spike rate and timing of spikes or spike timing. Neuronal activity consists of spikes or action potentials. Neuronal firing rates represent the average activity of a neuron. Neuronal network behaviors are activities that are repeated in the network. These include stochastic spiking, whole network bursting and avalanches. *Network Burst* is a synchronized spiking event that involves rapid spiking activity throughout the network [87]. Network bursts have been found to be a major component of activities in developing networks of cortical neurons [87]. Neuronal avalanches are runs of sequential spikes that occur more often than the network average and exhibit power-law distributions of avalanche sizes probability [13].

2.4 Spike-Timing-Dependent Plasticity

Refinement and maintenance of the central nervous system (CNS) is a crucial part in its development process which allows humans to learn new skills and behaviors, and store memories [6]. These changes are accomplished by rewiring and forming new connections between neurons using synaptic plasticity processes. Several spike- and rate-based learning rules have been introduced to explain the learning rules in CNS after the introduction of Hebbs rule, which says that correlated firing activity between two neurons leads to strengthening of synaptic connections [49]. STDP combines both synaptic modification and competition which makes it suitable as a Hebbian learning mechanism [80].

The precise timing of of firing of presynaptic and postsynaptic neurons can affect the efficacy and induce Long-term depression (LTD) and Long-term potentiation (LTP), which are the processes that weaken and strengthen synaptic connections respectively. This phenomenon is called spike-timing-dependent plasticity (STDP) [14, 17, 67]. When presynaptic spikes precede postsynaptic spike within a window of 20 ms, the strength of the synapse between the neurons is strengthened and when postsynaptic spikes precede presynaptic spikes by 0 to 20-100 ms, the synapse strength is weakened [17, 67].

Over the past few decades, a number of researchers have shown experimental evidence for STDP [14, 17, 67, 80, 79]. Bi and Poo [17] found out that persistent potentiation and depression of glutamatergic synapses were induced by correlated spiking of presynaptic and postsynaptic in cultures of dissociated rat hippocampal neurons, and highlighted the important of precise spike timing, synaptic strength, and postsynaptic cell type in STDP. Markram et al. [67] discovered that the magnitude and sign of potentiation and depression depended on the order and timing of

presynaptic and postsynaptic spikes. Studies have demonstrated that STDP is responsible for the development and refinement of neuronal circuits during brain development [16] and the selective pruning that leads to the shape a newborn's neuronal connections [51, 52]. STDP presents benefits like network stability, competition, sequence learning and prediction [27].

Analyzing the effect of STDP on the topology and the role of STDP on firing patterns are among the many questions that have raised interest in this area [79]. Most investigations have tried to use neural network modeling to look into these questions [15, 53, 56] . Some research has been done to investigate the problem from a theoretical viewpoint using mathematical frameworks based on graph theory [84]. The majority of studies done on STDP involve excitatory neurons. Haas et al. reported spike-timing-dependent plasticity of inhibitory synapses in the entorhinal cortex, and demonstrated its role in network modeling and network behavior efficiency [45].

2.4.1 Basic Spike-Timing-Dependent-Plasticity Model

In its standard form, STDP weight updates are defined for each pre-post spike pair and calculated using a double-exponential form. After calculating the synaptic modification, $F(\Delta t)$, the total amount of weight change can be calculated using different strategies. These strategies include: considering only the nearest neighbor spike pair, calculating the sum of synaptic modifications $F(\Delta t)$ for all pairs, or considering only the nearest postsynaptic spike. These are explained in detail in Section 2.4.3.

In the basic STDP model, the synaptic modification for a single presynaptic and postsynaptic spike pair is calculated using equation 2.1 [80]. Slightly varied forms of this equation have also been used to define various STDP models.

$$F(\Delta t) = \begin{cases} A_+ \exp(\Delta t/\tau_+) & \text{if } \Delta t < 0 \\ -A_- \exp(\Delta t/\tau_-) & \text{if } \Delta t > 0 \end{cases} \tag{2.1}$$

where A is the scaling factor which controls the amplitude of synaptic modification and, $+$ and $-$ represent potentiation and depression, respectively. Δt is the time difference between the two spikes, calculated by subtracting the presynaptic spike time from the postsynaptic spike time. Different experiments have shown the range of τ to lie between 20 to 100 ms. Song et al. used $\tau_+ = \tau_- = 20$ms. Figure 2.3 shows the plot of this model.

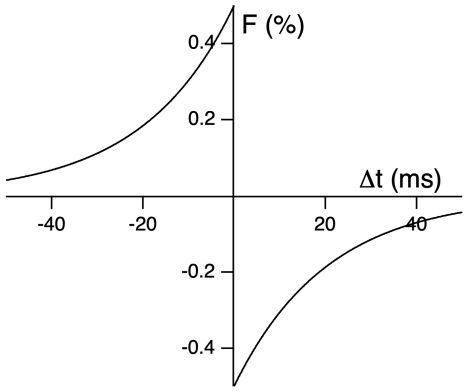


Figure 2.3: The STDP modification function from Song et al. [80]. The change of the peak conductance at a synapse due to a single pre- and postsynaptic action potential pair is F(t) times the maximum value gmax. t is the time of the presynaptic spike minus the time of the postsynaptic spike. In this figure, the synaptic modification, F, is expressed as a fractional change. After calculating F, the final weight can be calculated using the methods specifies in section 2.4.3. The values of constants are: $A_+ = 0.005$, $\tau_+ = 0.02s$, $A_- = 0.00525$, $\tau_- = 0.02s$.

2.4.2 Weight dependence: hard bounds and soft bounds

Synaptic weights range from a minimum value to maximum values in natural systems. Therefore, computational models must have bounds for synaptic modification. If left unchecked the synaptic weights could assume very high or low values which are not feasible. A common way to do is to establish hard bounds i.e set an upper and lower limit for the weights based on previous experimentation [61, 80]. Most models set A_- as 0.

Another way to set limits is to set soft bounds. In soft bounds, for large weights, synaptic depression is significantly more than potentiation [26]. Equations 2.3 and 2.2 demonstrate soft bounds. Some models use an amalgamation of soft and hard bounds [43].

$$A_+(w_j) = (w^{\max} - w_j)\eta_+ \tag{2.2}$$

$$-A_-(w_j) = w_j\eta_-| \tag{2.3}$$

where η_+ and η_- are fixed parameters, w_j is the synaptic weight being changes, A_+ and A_- are

the scaling factors for synaptic modification wher + and − represent potentiation and depression, and $w_{\max}$ is the maximum synaptic weight.

2.4.3 Different Implementations of STDP

When a single pre- and post-synaptic pair exists, the final weight of the synapse can be calculated by simply adding or multiplying the synaptic modification to the original synaptic weight. Calculating the final weight when there are multiple pre- and post- spikes involves using more complicated strategies. These strategies try to achieve a stable equilibrium for the network and using a different strategy results in a different STDP implementation. In each of these, the synaptic modification is calculated using equations similar to equation 2.1. Some commonly implemented STDP models are described below.

- **Classical Additive STDP:** In an additive model, synaptic modification, $F(\Delta t)$, for all pre- and post-synaptic spike pairs are added to the synaptic weight to calculate the final weight. This is the most basic model and considers the effect of all pairs equally regardless of the time difference between them. Since the weight is calculated as a summation, implementing upper and lower bound is extremely important to ensure that the weight values are feasible [56].

$$w = \sum_{i,j} w_{ij} \tag{2.4}$$

- **Suppression Additive STDP:** In this type of STDP, the final weight is calculated by the summation of suppression weight for each spike pair which is calculated using equation 2.5 and 2.6. Unlike the classical additive model, the suppression additive model does not give equal weightage to all synaptic modifications. This is due to the fact that the suppression weight changes based on the time elapsed since the presynaptic spike. This is the model used by Froemke and Dan [35], and shown in figure 2.4.

  In this model, each presynaptic and postsynaptic spike is assigned an efficacy, which depends only on the interval from the preceding spike in the same neuron [35]:

$$\epsilon_i = 1 - e^{(t_i - t_{i-1})/\tau_s} \tag{2.5}$$

where ϵ_i is the efficacy of the ith spike, t_i and t_{i-1} are the timings of the ith and $(i-1)$th spike, respectively, and τ_s is the suppression time constant.

$$w = \sum_{i,j} {\epsilon_i}^{\text{pre}} {\epsilon_j}^{\text{post}} w_{ij} \tag{2.6}$$

where ${\epsilon_i}^{\text{pre}}$ and ${\epsilon_j}^{\text{post}}$ are the efficacies of the presynaptic and postsynaptic spikes, respectively.

- **Nearest-neighbor additive STDP:** The nearest-neighbor additive STDP takes only two postsynaptic into consideration for each presynaptic spike: the postsynaptic spike that occurs just before and just after the presynaptic spike. The resulting weight is calculated by averaging the change calculated by the two spikes.

- **Multiplicative STDP:**

  In a multiplicative model, the synaptic modification, $F(\Delta t)$,is calculated as a percentage or fractional change and is then multiplied with the current synaptic weight to get the final synaptic weight. For multiple spikes, these contributions are done until the weight approaches the upper or lower limit [23].

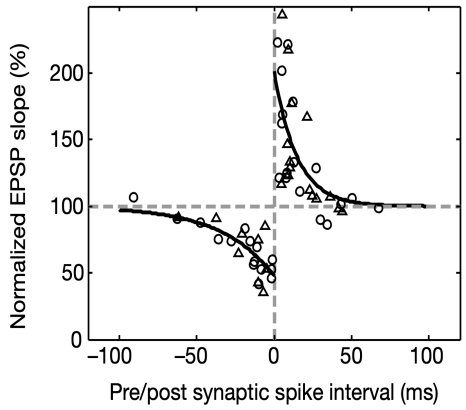


Figure 2.4: The STDP model, graph taken from Froemke and Dan [35]. The x-axis is the spike interval defined as the postsynaptic spike time minus the presynaptic spike time. The y-axis is the fractional change in excitatory postsynaptic potential (EPSP), a measure of the membrane potential in the postsynaptic neuron. Each point represents one experiment's data Each point represents one experiment. Circles, normal ACSF ($A_+ = 103 \pm 10\%$, $\tau_+ = 13.3 \pm 1.7ms$, $A_- = -51 \pm 1\%$, $\tau_- = 34.5 \pm 1.6ms$). Triangles, high divalent ACSF ($A_+ = 102 \pm 12\%$, $\tau_+ = 15.5 \pm 3.2ms$, $A_- = -52 \pm 6\%$, $\tau_- = 33.2 \pm 5.3ms$). The drawing curves are the math model defined in equations 2.5, 2.6.

Chapter 3

BACKGROUND: GRAPHICAL ANALYSIS OF THE BRAIN CONNECTOME

The patterns of connections between neurons can be used to learn a great deal about how the nervous system produces behavior. Scientists have tried to analyze this using brain graphs. Brain graphs provide a simple and effective way to represent the human brain connectome, which is a comprehensive map of neural connections in the brain. Different properties of the brain have been studied, and many structural and functional brain graphs have demonstrated topological properties like small-worldness, modularity, community structure and heterogeneous degree distribution [22, 38]. These properties are consistent with properties shown by specific random graphs, social networks, powergrids, Worldwide Web and various other networks [38]. Topological properties of brain connectivity networks has been used to study the effects of breaks on brain activity, divergent thinking ability and affects of aging on the brain [12, 36, 82].

In this chapter, we provide a brief overview of some concepts about random graphs and brain graphs to have a better understanding of the work done in this thesis on the graphical analysis of the human brain connectome.

3.1 Random Graph Theory

Pául Erdős and Alfréd Rényi introduced the theory of random graphs to give probabilistic construction to large-scale graphs [30, 31]. The two basic random graph models define the graph as $G = (n, e)$ and $G = (n, p)$. In random graph model $G(n, p)$, where an edge exists between two nodes with probability p, which is independent of all the other edges [37]. The random graph model $G(n, e)$ has n nodes and e edges; all such graphs have uniform distribution [30]. Unlike $G(n, p)$, the edges in the model $G(n, m)$ are not independent of each other. The existence of an edge decreases the likelihood of all other edges because of the fixed number of edges.

Various other models of random graphs have been proposed that illustrate specific properties. Aiello et al. proposed a random graph model which satisfies the power law and has behavior consistent with a telecommunications data [4].

Random-graph theory focuses on studying properties of random graphs. There are two kinds of graph properties or features: global and nodal. Global features consider the complete graph and nodal features only focus on a particular node. These properties are used to study both directed and undirected graphs, and are calculated in similar ways for both. The following subsections summarizes some metrics and structural properties of a random graph.

3.1.1 Node Degree, Degree distribution and Connection Density

The degree of a node is the number of connections the node has with other nodes in the network. Most of the other measures use or derive some part from nodal degree. The degrees of all nodes in the network make up the degree distribution. Connection density is defined as the ratio of the number of edges in a graph to the number of possible edges in a graph i.e. $n(n-2)/2$.

3.1.2 Subgraphs, connected component and giant component

One of the first properties studied was the appearance of subgraphs which is a subset of the set of nodes and edges of the original graph [5, 30]. A connected component of the graph is a subgraph in which every node has a path to every other node in the component and the subset is not part of a larger connected component [28]. A connected component that contains a significant portion of the nodes of the graph is called a giant component [28]. There is a critical value of p at which a random graph starts forming giant components.

3.1.3 Clustering Coefficient

The clustering coefficient of a node i.e. local clustering coefficient is the ratio of the number of shared edges between a node's neighbors to the number of possible edges between the neighbors. The global clustering coefficient is defined as the number of triplets or triangles in the graph to the total number of possible triplets as described in equation 3.1 [63]. It is used to measure the degree to which nodes in a network tend to cluster. Denser connections result in high clustering coefficient values. This is because the concentration of the connections is directly proportional to nodes forming clusters. Figure 3.1 shows the variation of a graph with low and high degree and clustering coefficient values. The graphs with high clustering coefficient has higher average degree and more triangle formation than the graphs with a lower clustering coefficient.

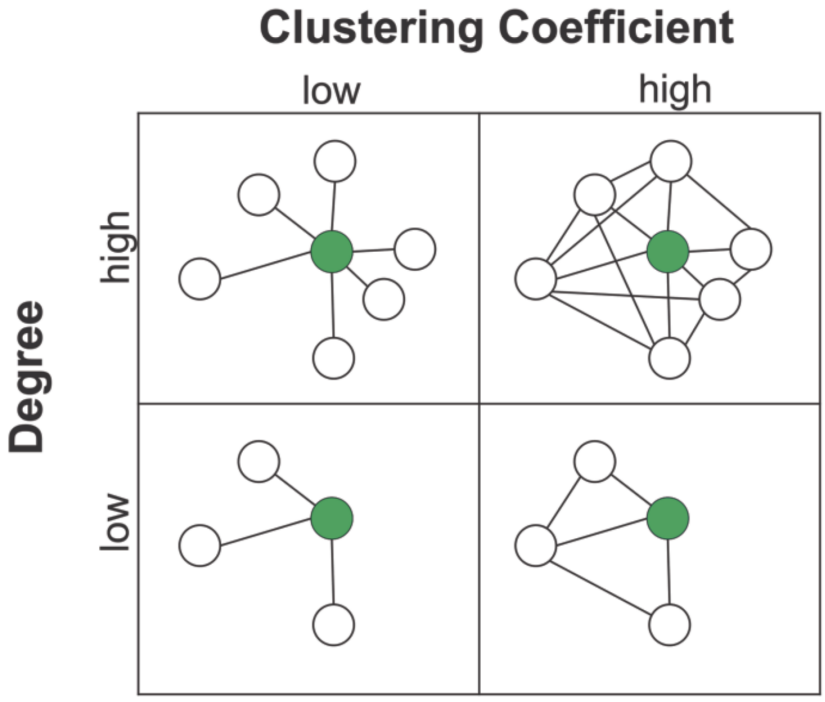


Figure 3.1: Representation of networks with different values of degree and clustering coefficient from Teppa et al. [83].

$$C(G) = 3\frac{\#\text{ of triangles in the network}}{\#\text{ of connected triples of vertices}} \tag{3.1}$$

3.1.4 Path length

Path length is the summation of the weights of the edges that are traversed to go from one node to another. In an unweighted graph, this is the number of edges that are in the path from one node to another as the weight of each node is considered to be the same or 1. In a weighed graph, the sum of the weights of the edges in the path is calculated. The lower the path length, the higher the efficiency of information transfer. The average path length (L) in a network is the average shortest path where shortest path is calculated for all possible pairs of network nodes. Average path length is also called characteristic path length.

3.1.5 Centrality, hubs and modularity

Centrality of a node is used to define the importance of a node in the network based on connectivity relative to other nodes. Nodes with high degree of connectivity or high centrality form hubs. Commonly, centrality measures how many shortest distance paths in the network pass through that node. It is the property of a network in which a subgraph is tightly connected to one another and weakly connected to other subgraphs or nodes in the network.

Different types of centrality measures are listed below:

- **Degree Centrality:** Degree centrality of a node is the degree of the node, which is the number of edges connecting to the node. In directed graphs, the number of incoming edges is called indegree and the number of outgoing edges is called outdegree. A high degree value means that a node is connected to a lot other nodes, which further implies that the node is more central to the network [39]. The degree centrality measures importance of a node in the network based on the quantity of connections, and does not pay any heed to the location of the node in the graph. Betweenness centrality, which is discussed next, overcomes this caveat and also focused on the location of a node with respect to all other nodes in the network.

- **Betweenness Centrality:** Betweenness centrality measures how many times a node appears on the shortest paths between any two nodes in the graph. It measures a node's importance in information passing through the network. This can be better understood by taking the example of a network connecting cities by road. If a city lies in between the shortest paths connecting a large number of cities, more people are likely to pass through the city on the way their way to their destination. The city will likely to develop as a hub and has a location advantage. Similarly, if you wanted to pass information between any two nodes, if a node lies in the paths for a large number of node pairs, that node is central to allowing information to pass from one part of the network to another. We calculate the betweenness for a node i by summing the fraction of shortest paths for all possible node pair in the network that pass through node i [34]. The formula of betweenness centrality for a node i is shown in equation 3.2.

$$b_i = \sum_{s,t} \frac{n^i_{s,t}}{n_{s,t}} \tag{3.2}$$

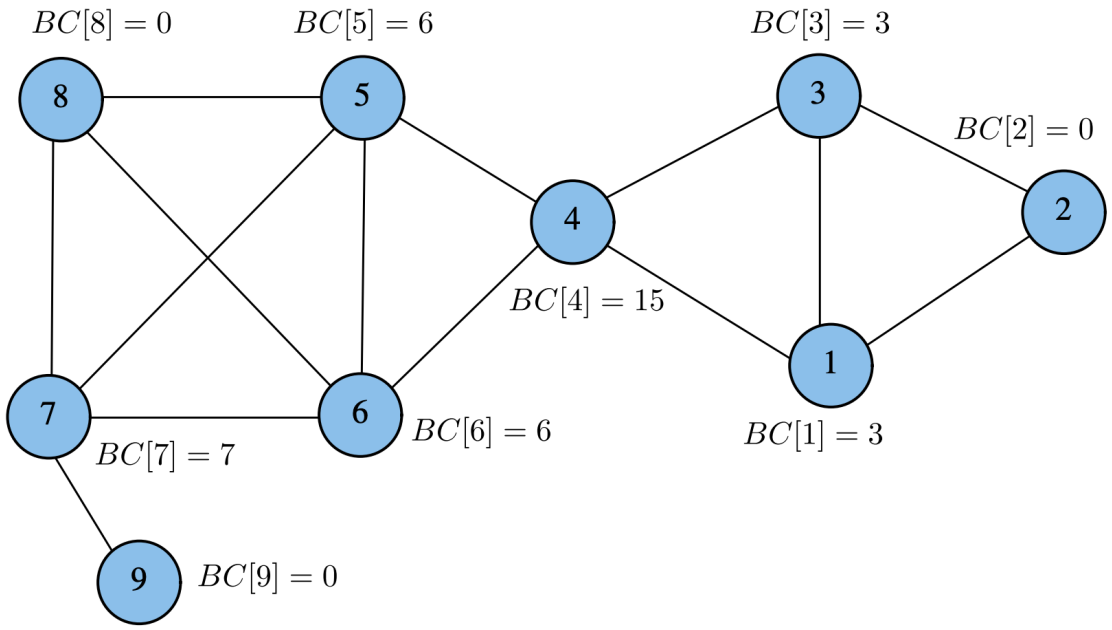


Figure 3.2: Example betweenness scores for a small graph from McLaughlin and Bader [69].

where $n^i_{s,t}$ e the number of shortest paths from s to t that pass through i, $n_{s,t}$ is the total number of shortest paths from s to t, and b_i is the betweenness of the node i.

An example of betweenness centrality is shown in figure 3.2. Vertex 4 connects the left part of the network to the right, and thus, lies on the shortest paths from all the vertices on the left (5 to 9) to all the vertices on the right (1 to 3) and vice versa. This results in a high betweenness centrality value for vertex 4. Whereas, vertex 8 lies in the corner and does not lie on any of the shortest paths, and thus, has zero betweenness centrality [69]. If we block vertex 4, no information can pass through left and right, and network connectivity will be affected adversely. Vertex 8 has virtually no effect on the information passing through the network and blocking information passing through it does not affect network connectivity that much.

- **Closeness Centrality:** Closeness centrality is the inverse of the average distance between a

node and all other nodes in the network [57]. Incloseness considers paths that from all other nodes to node i and outcloseness considers paths from node i to all other nodes. The formal definition is given in equation 3.3 below:

$$C^i = \frac{N-1}{\sum_{n \neq i} d_{ni}} \tag{3.3}$$

- **Eigenvector Centrality:** Eigenvector centrality depends on the degree and centrality values of a node's neighbors. It is the first eigenvector of the adjacency matrix, which has the largest eigenvalue λ_1 (principal eigenvalue). Degree centrality only measures the number of nodes that a node is connected to, and it doesn't consider if those nodes are important or not. For example, if two people in a social network have the same number of connections, but one of them has more famous connections then this person will have a higher impact on the network. Eigenvector centrality takes this into account the importance of the adjacent nodes. PageRank centality is a variant of this.

- **PageRank Centrality:** A modified form of the original PageRank is used to measure importance of nodes based on connectivity [20]. The PageRank centrality of a node depends on the number of incoming edges or links, the edge or link propensity, and the centrality of the connecting node or linkers [72]. It incorporates the direction of a connection which is extremely helpful in networks where the direction of information flow is relevant to importance of a node. The PageRank centrality, x_i, of a node i is given in equation 3.4.

$$x_i = \alpha \sum_k \frac{a_{k,i}}{d_k} x_k + \beta \tag{3.4}$$

  where $(a_{k,i})$ is an outgoing edge from node k to i, d_k is the out-degree of node k, if k has no outgoing nodes then the values of d_k is 1, x_k is the centrality value of node k, and α and β are constants. α is the damping factor whose value lies between 0 and 1, and β is the personalization vector.

- **Hubs and Authorities Centrality:** These are linked to each other. The hubs centrality score is the sum of the authority values of its neighbors and vice versa. This centrality

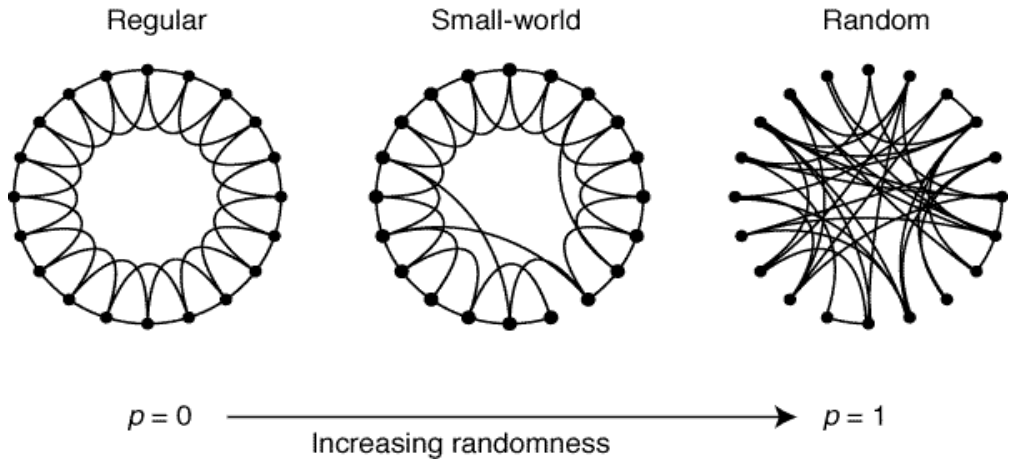


Figure 3.3: The small-world network generated by Watts-Strogatz using random rewiring from Watts and Strogatz [92]. The network interpolates between a regular ring lattice and a random network and does not alter the number of nodes or edges. They started with $N = 20$ nodes, each connected to its $k = 4$ nearest neighbors. For $p = 0$ the original ring is unchanged; as p increases the network becomes increasingly disordered until for $p = 1$ all edges are rewired randomly. For intermediate values of p, the graph is a small world network with high clustering value and small characteristic path length [92].

measures tries to evaluate the importance of a node based on the nodes its directly connected with.

3.2 Small-World Networks

A graph has small-world property if it has a high clustering coefficient and small average path length. It is the ratio of the clustering coefficient (C) and the average path length (L) of the network. Watts and Strogatz proposed a small-work network using one-parameter that interpolated between a regular ring lattice and a random network [92]. Watts-Strogatz (WS) model uses a rewiring procedure to generate small-world network. From a ring with n vertices where each node is connected to k nearest neighbors, an edge is reconnected with probability p to a vertex chosen uniformly at random as shown in figure 3.3. When $p = 0$ the ring remains unchanged; a random graph is generated when $p = 1$; all intermediate values of p result in a small-world graph.

3.3 Brain Graphs

A brain graph is defined as a model of the nervous system which has a set of nodes connected via edges. Graph analysis is applicable to any scale, modality, or volume of neuroscience data which allows researchers to use a wide range of existing analysis tools and properties [9, 21]. Representing the data collected from the brain as graphs provides a common language for analysing complex systems and comparing brain networks to other real-life networks like World Wide Web, road network etc. There are two kinds of brain networks: structural which demonstrate the inter-neuronal connectivity to inter-regional connectivity, and functional which demonstrate how the structural architecture supports neurophysiological dynamics. Figure 3.4 demonstrates the adjacency matrix generated from a single MRI dataset by [22].

Different imaging techniques like MRI, diffusion tensor imaging etc. can be used to collect data from the brain, but the basic steps for exploring brain networks using graph theory remain the same. The first step is to define the network nodes and edges. The definition varies depending on the imaging technique used. Some common approaches to define node and edges are anatomical parcellation and defining size of a node using the number of voxel [94, 95]. The next step is generating an association or adjacency matrix. At last, structural and functional brain graphs can then be explored by using graph theory by calculating the network parameters and comparing them to the metrics calculated of random graphs of the same size [21]. Figure 3.5 demonstrates the steps for graph theoretical analysis.

3.4 Measures on Brain Graphs

The brain is a high-performance network and there are various metrics used to analyze its architecture. There are two main classes of measures are: topological and geometric. Topological measures evaluate the relationship between the nodes based on the network's topological architecture. These measures don't take into consideration the physical location. Rather than using the distance between the nodes as the edge weight, they use synapse strength. Some popular topological measures are defined in section 3.1 and shown in figure 3.6. Geometric measures evaluate the physical distance between nodes and try to find the relationship between the network function and geometry. These allow us to analyze the wiring length of connections [24].

3.5 Application of centrality measures and correlation between them

Centrality measures have been used to identify structurally core regions, which have high degree and betweenness centrality, within the brain, and evaluate study network efficiency [40, 47, 89]. These measures have been used to find connectivity patterns common for various disorders [64]. Previous studies have tried to determine if the centrality measures are correlated to each other since they are calculated by doing different mathematical computations on the same adjacency matrix. A study by Bolland [19] analyzed the performance and correlation between degree, closeness, betweenness, and flow centrality measures. The study found out that degree, closeness, and flow centrality has a high correlation coefficient, but betweenness centraity had a low correlation coefficient with the other measures [19]. Another study by Valente et al. [85] found out that the average correlation coefficient between different centrality measures was 0.53 indicating that each measure is associated with specific outcomes but because they're all caluculated using the same thing i.e. adjacency matrix. The study also showed that symmetric centrality measures are dependent on network size and density whereas asymmetric measures are independent of these features.

Zuo et al. [96] studied the resting state fMRI of 1003 subjects from 21 centers and investigated network centrality measures to provide insights about the connectivity of the whole-brain functional network. They summarized their findings by creating a mean functional brain network. This network was formed by averaging the Fisher z-values of the correlation matrices from all participants and then back-transforming to correlation matrices. Figure 3.7 shows the distribution of degree, eigenvector and PageRank centrality.

3.6 Pitfalls of graph analysis of human connectome

There are various methodological issues with data collection using techniques like Functional magnetic resonance imaging or functional MRI (fMRI). Some of these questions include: how do you define a node in the images collected, whether the scans should be collected in resting-state or when the subject is performing a cognitive task, what should be the appropriate time of the time-series of an fMRI scan, and what are the differences in each region's connectivity profile [22]. Brain maps representing structural connections commonly use diffusion-weighted-imaging (DWI) to measure anatomical connectivity. This technique is unable to differentiate between afferent and efferent connections, and is also known to include some spurious connections [1].

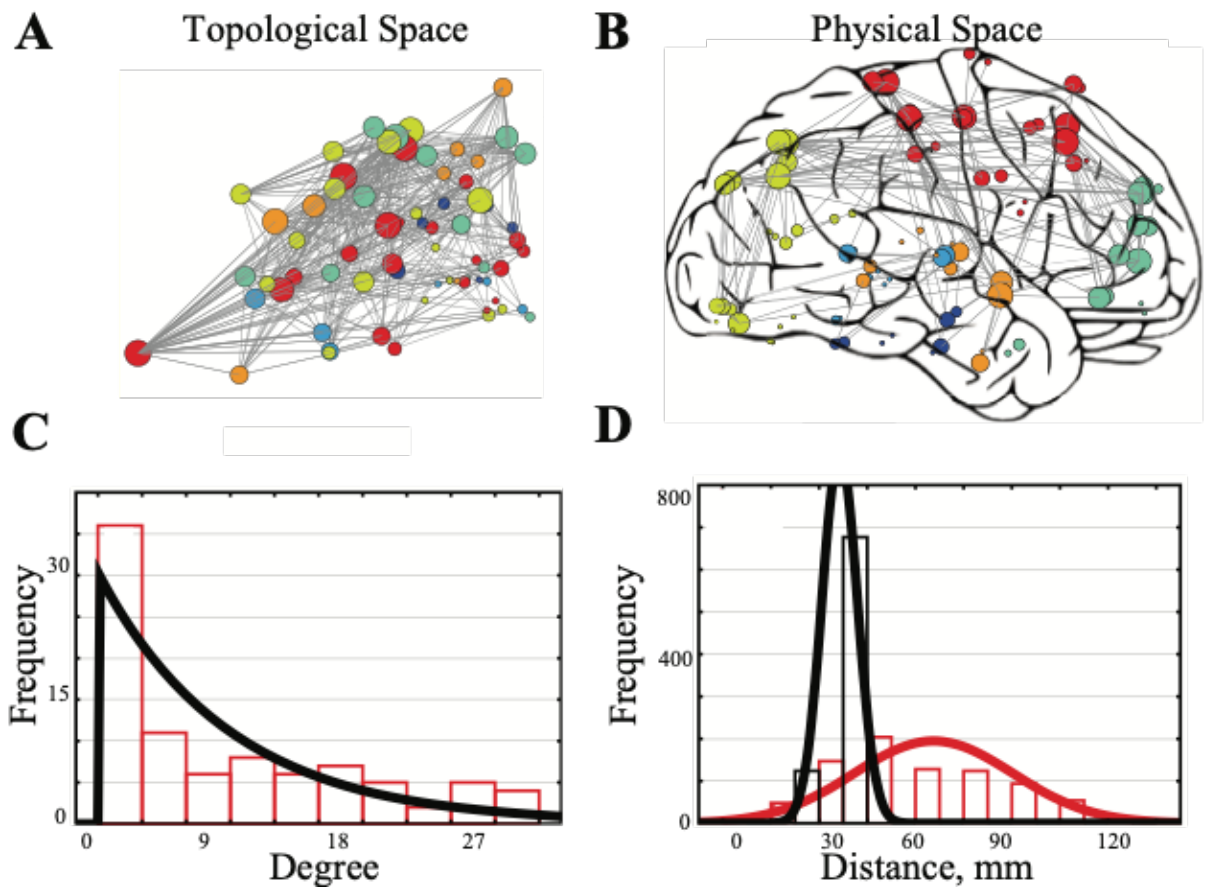


Figure 3.4: Topological and geometrical properties of functional brain graphs as shown in Bullmore and Bassett [22]. (A) Adjacency matrix for a single MRI dataset calculated at a cost of $\kappa = 0.05$. The size of nodes indicates nodal degree whereas the color of the node indicates its lobar identity. (B) Brain graph plotted in physical space where the distance between nodes is the Euclidean distance between regional centroids in the anatomical space of the real brain. The size of nodes indicates nodal degree whereas the color of the node indicates its lobar identity. (C) Plot of the degree distribution (red) and distribution fit (black) of the brain graph, showing the predominance of low-degree nodes and the presence of a few high-degree hubs. (D) Plot of the distribution of physical distance of connections of the brain graph (red) in comparison to the distance distribution in a minimally rewired network (black).

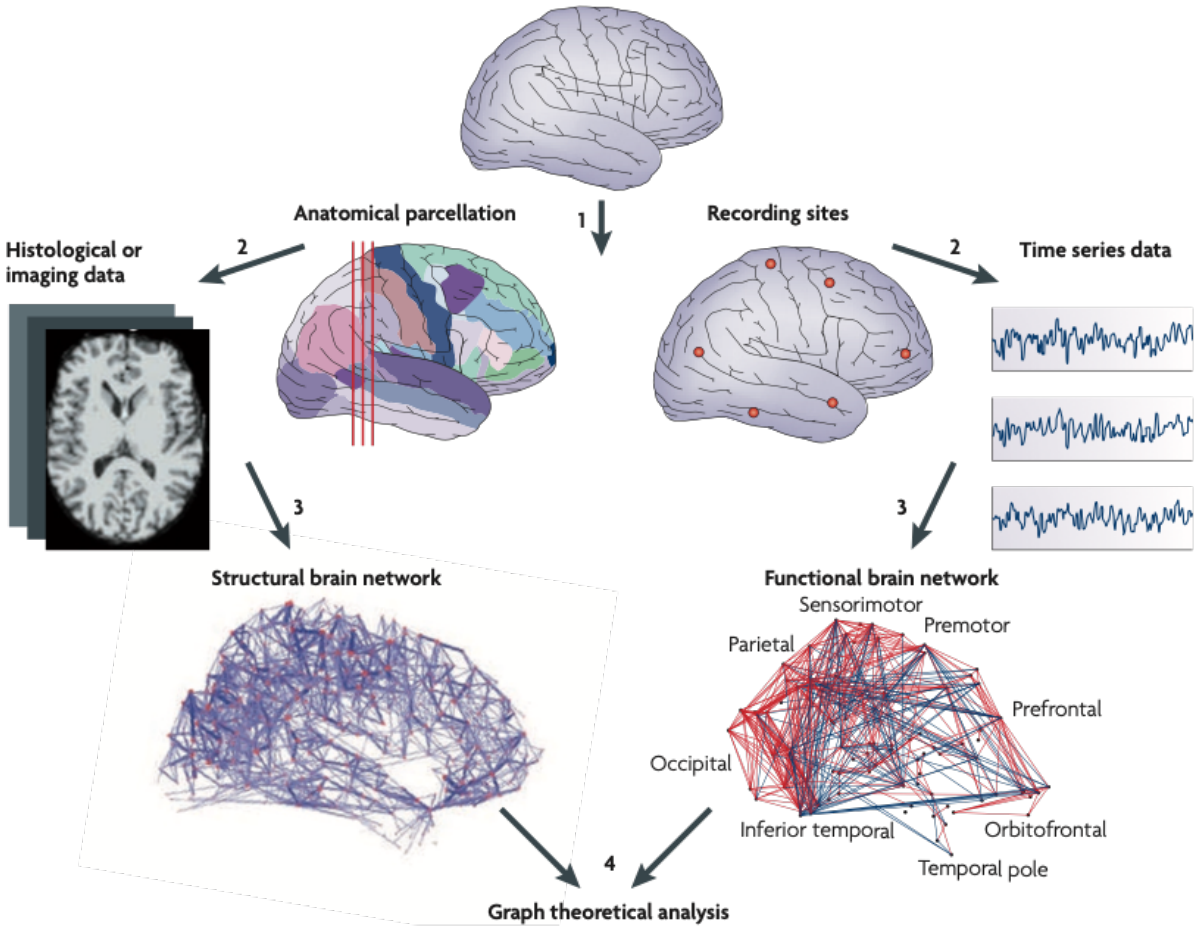


Figure 3.5: Steps for structural and functional analysis of the brain graph as shown in Bullmore and Sporns [21]. The first step is defining nodes. Then estimating a continuous measure of association between nodes i.e. edges. The third step is generating the adjacency matrix and the last step is calculating network parameters required for analysis.

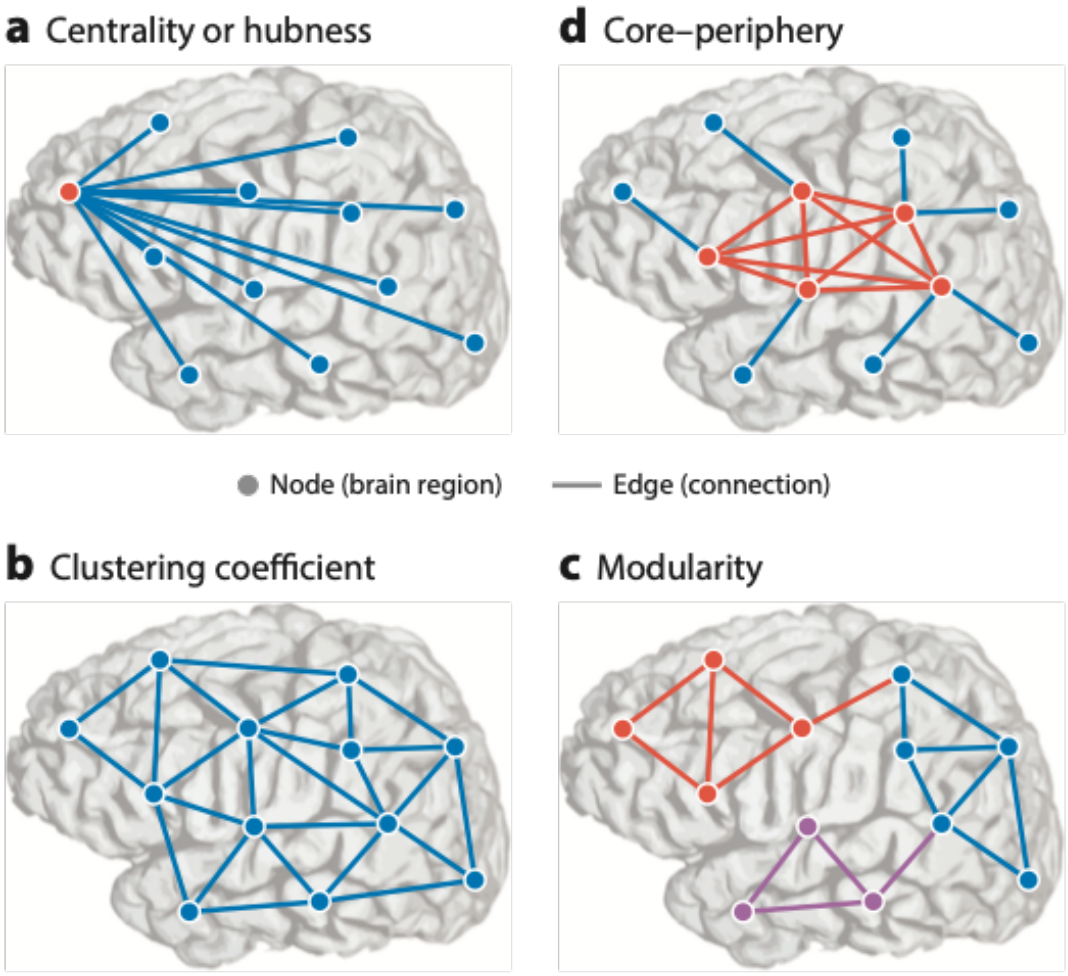
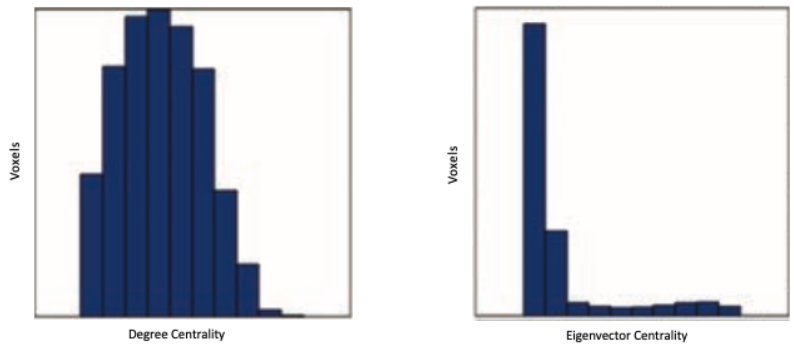
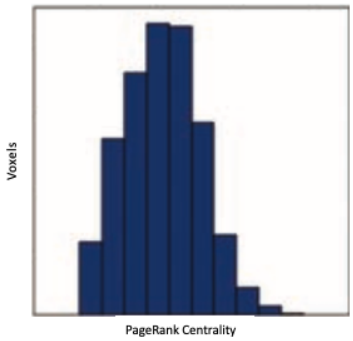


Figure 3.6: Multiscale topology in brain networks from Bassett et al. [11]. (a) Nodes with more connections or stronger edges tend to be hubs (red), whereas nodes with fewer connections tend to be isolated (blue). (b) Networks with strong clustering coefficient demonstrate a high density of triangles that is believed to facilitate local information processing. (c) Nodes of different communities are colored red, blue, or purple. (d) Networks with core–periphery structure exhibit a set of tightly connected nodes (core, red) sparsely connected to a set of isolated nodes (periphery, blue).



(a) Degree Centrality

(b) Eigenvector Centrality



(c) PageRank Centrality

Figure 3.7: The histograms of each voxel-wise network centrality measure score from Zuo et al. [96]. Each voxel in images generated by fMRI is treated as a node. This figure shows the distribution of degree, eigenvector and pagerank centrality.

Chapter 4

METHOD: SPIKE TIMING INDEPENDENT PLASTICITY IMPLEMENTATION

Synaptic plasticity is an important feature in spiking nervous networks, and is responsible for refinement and maintenance of the CNS. This has led to a rise of interest in plasticity mechanisms, and STDP has emerged as a leading candidate for synaptic and neural plasticity during development and learning [75]. Simulating plasticity in cell cultures requires repetitive simulations and detailed cell level information cannot be gathered due to physiological barriers. We wanted to collect data about the neural network from large neural population over period of development and refinement. For doing this, we used BrainGrid [81], a neural network simulator. BrainGrid uses a simple model of cortical culture growth using a leaky-integrate-and-fire-model proposed by Kawasaki and Stiber [60]. This model reduces network complexity while reproducing network behaviors in living systems.

In this thesis work, we reproduced simulations using the leaky integrate-and-fire model from [60] to simulate cortical culture growth and extended the work to implement synaptic plasticity to the network. Previous work done on BrainGrid included implementing STDP, and serialization and deserialization using the cereal library [50]. The serialization feature can be used to store the connections after the growth simulation and then refine the network by using STDP. The ability of our simulator to remove the extra connections after the growth phase allowed us to explore the behavior of the network in the pruning phase.

This chapter is organized as follows: the next section gives a brief introduction about the BrainGrid simulator. Section 4.2 describes the methodology for the STDP implementation, section 4.4 describes the simulation workflow for the STDP model, and section 4.5 specifies how the data is collected for analysis. The experimental setup and simulator configuration are mentioned in section 4.6.

4.1 BrainGrid Simulator

The BrainGrid simulator takes in an input XML parameter file which allows the user to modify the simulation parameters including number of neurons, epochs, layout of the neurons, type of model

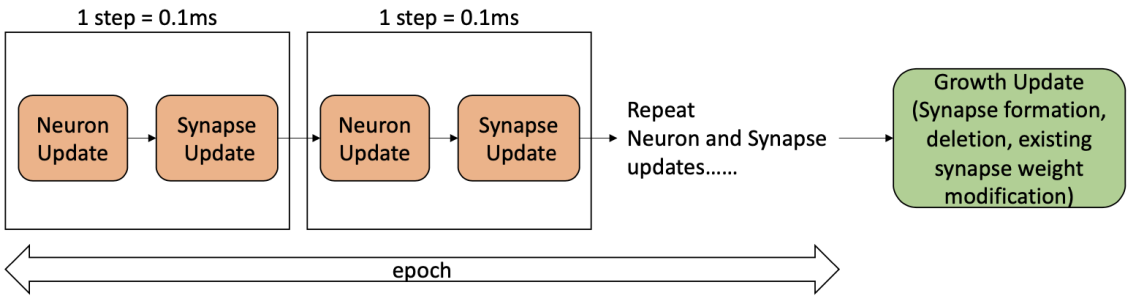


Figure 4.1: Workflow of each epoch in the growth simulation from Hsu [50]. Synapses and neurons are updated within each epoch at every step, and synapses are formed, deleted and updated at the end of the epoch in the growth update phase.

etc. There are two types of the models available in BrainGrid: the cortical culture growth model and the STDP model.

The *cortical culture growth model* simulated growth of a network in dissociated cortical cell cultures and is able to simulate growth equivalent to 28 *days in vitro (DIV)*. In the growth simulation, neuron and synapse states are updated within each epoch every 0.1ms, but no new synapses are formed or deleted until the end of each epoch, as shown in figure 4.1. The growth update phase happens at the end of each epoch, and during this phase, synapses are formed, deleted, and synaptic weights are adjusted. An epoch in this model is defined as the period where synapses and neurons are updated in steps and the end of an epoch is marked by the growth update phase.

The growth model does not include the synaptic modification development factor. The *STDP model* overcomes this shortcoming and incorporates a spike-timing based adjustment to the simulation. The basic STDP model implemented is explained in section 2.4 and the algorithm used is explained in section 4.2. According to the model, the synaptic weights are updated if the time interval for pre- and post-synaptic spikes is within the set limit. Unlike the growth model where the weights are updated at the end of each epoch, in the STDP model, the weights are updated immediately after a valid spike pair is found. Thus, the epoch definition for the STDP model is different than the growth model. In the STDP model, the end of an epoch is marked by the phase where we record the state of the network. This is explained in section 4.5.

4.2 STDP Implementation

The STDP synapse classes were already implemented in BrainGrid, but the classes were still in a nascent stage. We followed the STDP models implemented by Song et al. [80] and Izhikevich et al. [56] and modified the existing class. The two key components for STDP implementation is transmission of signals from a presynaptic neuron to a postsynaptic neuron and adjusting the synaptic weights based on STDP model. In this thesis, we worked on the latter and extended the capability of the simulator by adding different STDP implementations.

In the STDP model, the stdpLearning() method is called for each synapse for adjusting the synaptic weights. When a spike pair enters the stdpLearning() method, the algorithm calculates the time interval and if it is less than the STDP gap set, the function synapticWeightModification() is called to calculate the scaling ratio for weight. The synapticWeightModification() allows us to change the STDP implementation without changing the rest of the program. The two methods implemented are classical additive and suppression additive, these are described in section 2.4.

This fraction is added to 1.0 so that we have the scaling ratio. The scaling ratio has to be greater than zero because negative the range of scaling ratio is from zero to one. If the scaling ratio is less than zero, it is reset to zero. The scaling ratio is then returned to the stdpLearning() method. The final weight is calculated by multiplying the weight with the scaling ratio. At the end, we check whether the final weight is within the upper and lower bounds. If weight exceeds the upper bound, we set it equal to the upper bound, and if the weight is less than the lower bound, we set it equal to the lower bound. The algorithms for both the functions are specified in algorithm 1 and 2.

4.3 STDP Simulation Workflow

The STDP simulation can be run in two ways. The first way is to run an independent simulation. The workflow for this is shown in figure 4.2. In order to do this, we need to initialize weights of synapses to any value between the upper and lower bound. The results produced by this are not that interesting since the pruning is done on a random network. The second method allows us to study the effects of STDP on a growth simulation. The workflow for this involves two simulations, a growth simulation and the STDP simulation as shown in figure 4.3. Each of these represent different phases in network development. For the first run, we grow a network using

Algorithm 1: stdpLearning(): Calculate final synaptic weight after STDP modification

```
Result: weight after synaptic modification
/* values of all constants are loaded from the configuration file        */
if time_interval ¡ STDP_Gap then
    Function: synapticWeightModification();
    scaling_ratio = synapticWeightModification();
/* //negative scaling_ratio means weight is adjusted to zero and will never
   be changed                                                            */
/* This means the synapse is deleted.  No connection (synapse) anymore   */
if scaling_ratio < 0 then
    scaling_ratio = 0;
/* same final weight calculation classical and suppression additive      */
weight = weight * scaling_ratio;
/* check if weight is within bounds                                      */
if weight > upper_bound then
    weight = upper_bound;
if weight < lower_bound then
    weight = lower_bound;
```

the cortical culture growth model. In the growth simulation, the growth model and the dynamic synapse model were used. At the end of the simulation, we store the network state by using serialization, a programming technique for storing objects. Next, in the second simulation, we used deserialization, a programming technique to read objects from byte stream, and then ran the simulation. The STDP model was used for weight modification. In the STDP simulation, the static model and the STDP synapse model were used. As the synaptic weight modification is based on STDP learning rule, the impacts of STDP can be observed on the growth network. A brief explanation of serialization/deserialization is given in the next section. These frameworks can be used to explore new network models by implementing different STDP models and varying network parameters like number of neurons, epochs, etc.

Algorithm 2: synapticWeightModification():Calculate scaling ratio for synaptic weight modification

Result: scaling ratio for weight change

```
/* values of all constants are loaded from the configuration file        */
if presynaptic spike precedes postsynaptic spike then
    /* potentiation                                                      */
    fractional_change = Apos * exp(-(time_interval)/taupos);
else if postsynaptic spike precedes then
    /* depression                                                        */
    fractional_change = Aneg * exp(-(time_interval)/tauneg);
/* calculating values of epre and epost                                  */
if spike is preSpikeHit then
    epre = 1.0 - exp(time_interval/tauspre);
else if spike is postSpikeHit then
    epost = 1.0 - exp(time_interval/tauspost);
/* calculating scaling ratio depending on type of STDP                   */
if classical additive STDP method then
    scaling_Ratio = 1.0 + fractional_Change;
if suppression additive STDP method then
    scaling_Ratio = 1.0 + fractional_Change * epre * epost;
return scaling_ratio
```

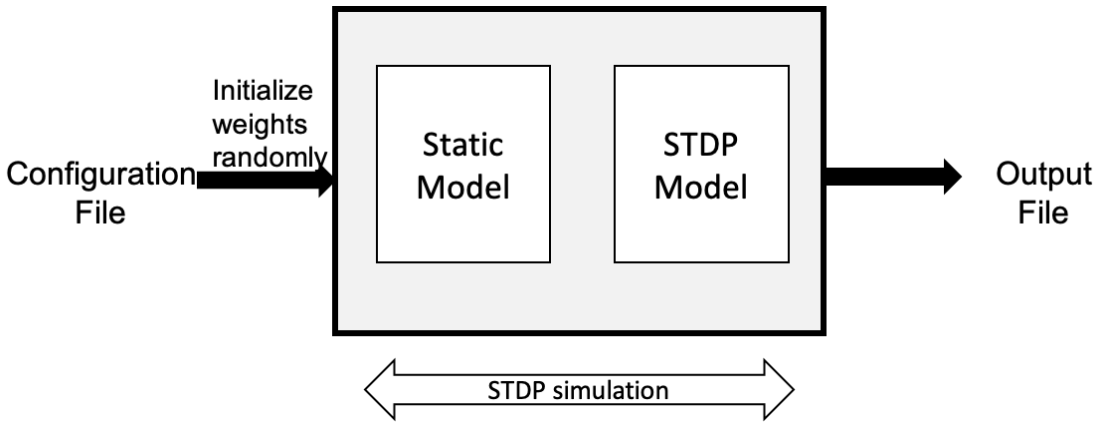


Figure 4.2: The design workflow of an independent STDP simulation.

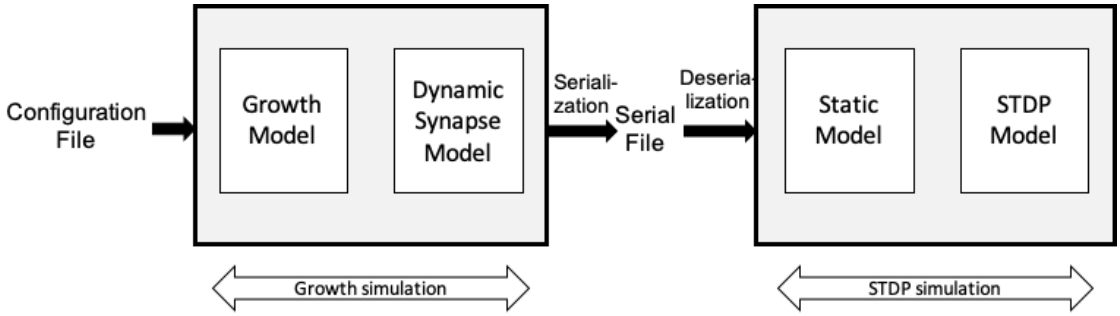


Figure 4.3: The design workflow of a two-step STDP simulation. At first, a growth simulation is conducted. Once the network is grown and serialized, the network state is regenerated by deserializing data, and the STDP simulation is started.

4.4 Serialization-Deserialization in BrainGrid

Serialization is a mechanism which transforms a class object to byte stream, and store it. Deserialization works the other way around, and converts byte stream to objects. These mechanisms can be used to store or transmit objects for later use. In BrainGrid, the Cereal library, which is an open-source C++ 11 serialization library, was used to implement serialization and deserialization [41]. The detailed explanation of the serialization implementation can be found in [50]. Serialization is done at the end of a simulation and information about the synapses including the synaptic weight, source and destination neurons are written into a file. These elements are the most representable elements of the network and carry enough information to recreate the state of the network. This output file can be used to initialize connections of another simulation by deserializing it at the beginning of the simulation. During deserialization, synapses are re-created by using synapse weights, destination and source neuron index. BrainGrid allows the users to choose whether to perform serialization, or deserialization, or both.

We wanted to analyze how the network changes after implementing STDP to a growth simulation. In order to do this, we serialized the final weights of growth simulations and then deserialized the serial file to create synapses, and performed the simulation using the STDP model. The values of the parameters used in these simulation runs and the experiment setup is explained in the next section.

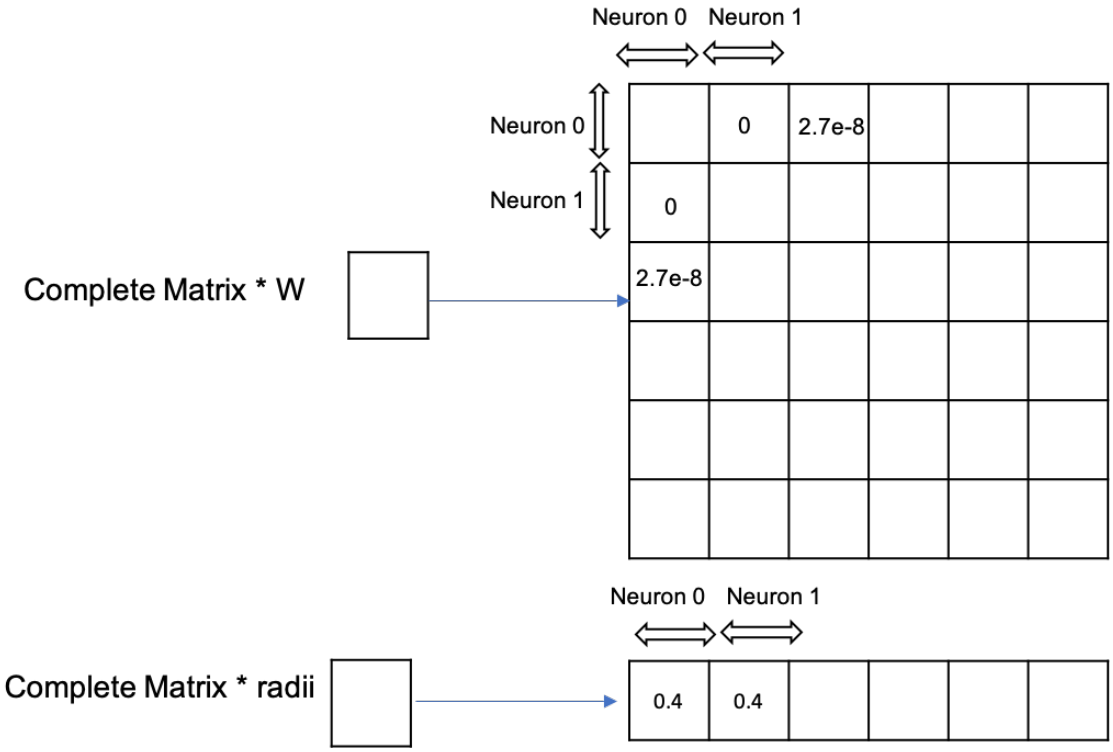


Figure 4.4: Matrix representing weight of each synapse and radii of connectivity for each neuron. In a growth simulation, weights are data is stored in a adjacency matrix format, and rates and radii of connectivity is stored in a one-dimensional array.

4.5 Data Acquisition

Network data was output in xml format for further analysis. The existing Xml Recorder classes, which are used to store the final simulation file in xml file format, were extended to implement recorder classes for storing results generated from an STDP simulation.

At the end of each growth simulation, the simulator outputs the following information:

- simulationEndTime: Total simulation time in seconds (60,000 s)
- probedNeurons: Unique ID for each neuron (0 - 9999)
- xloc: x location for each neuron (0 - 99)
- yloc: y location for each neuron (0 -99)

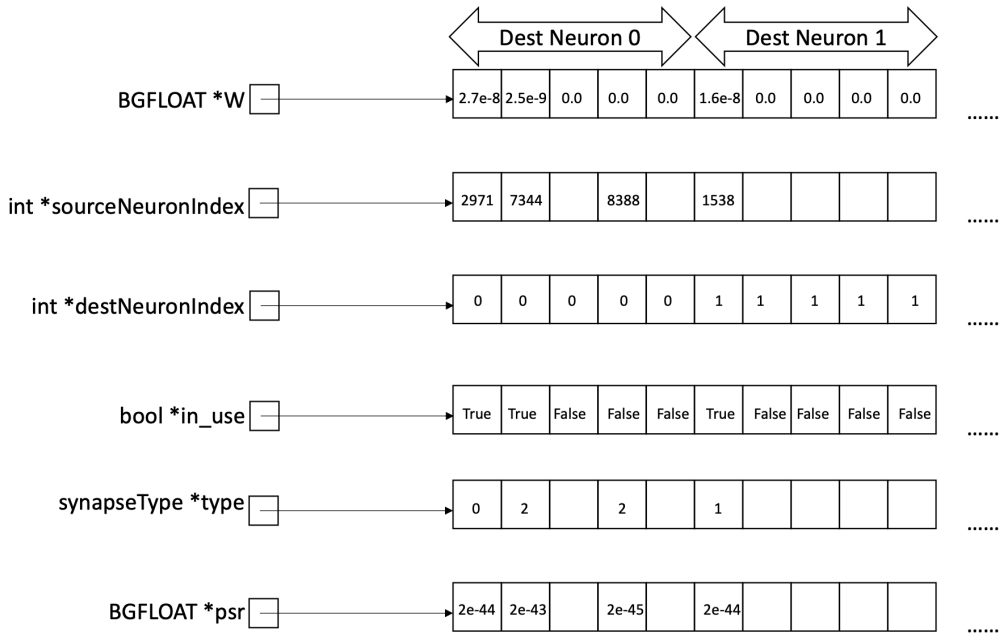


Figure 4.5: Synapse properties as shown in Hsu [50]. Synapse properties are stored in dynamic allocated arrays from. Each property is one array (from top to bottom, synaptic weight (W), source neuron (sourceNeuronIndex), destination neuron (destNeuronIndex), in_use array (in_use), synaptic type (type), and postsynaptic response (psr)). The current states of one existing synapse are located in the same index position in each array.

- spikeshistory: Total spike count for every 10 ms time bins
- spikesProbedNeuron: Spiking timing and location for every spike
- radiiHistory: Radii of connectivity of each neuron
- ratesHistory: Spiking rate of each neuron

In the growth simulation, the dynamic arrays store data for each neuron as shown in figure 4.4. The data is stored in an adjacency matrix format for network parameters like weights and radii, but properties of individual neurons are stored in a one dimensional matrix format. Neuron properties are written down for every epoch at the end of the simulation. Therefore, the size of the final radii

and spike rates array written to file is $(numberOfEpochs + 1) * (numberOfNeurons)$. In order to extract synapse weights further calculation needs to be done as explained in section 5.2.

At the end of the simulation, the simulator outputs the following information, in addition to the outputs specified for the growth simulator above, for the STDP simulation:

- sourceNeuronIndexHistory: Index of the source neuron for a synapse
- destNeuronIndexHistory: Index of the destination neuron for a synapse
- weightsHistory: Weight of a synapse

In the STDP simulation, dynamic arrays store information for each synapse as shown in figure 4.5. No new connections are added and only the ones that exist are updated based on the STDP model. The arrays should be able to store all the possible synapses for a network, and thus, the size of the final array output in the file is $(numberOfEpochs + 1) * (maxSynapsePerNeuron * numberOfNeurons)$. The synapse number is used to index the arrays and access the corresponding property value. If for an index n which is less than the total synapses possible, a connection exists if the synapse weight corresponding to n is greater than 0. The source and destination neuron can be found using index n in their respective arrays.

4.6 Simulation Configuration and Experiment Setup

The BrainGrid neural simulator [81] was used to simulate a network of neurons using the growth model and model refinement was done using the STDP model. The network consists of 10,000 neurons in a 100x100 rectangular arrangement. In this thesis work, the simulation had 90% excitatory cells was chosen as a representative dataset for graphical analysis.

The growth simulation is equivalent to 28 days development, as described in detail in [1]. Each time step is 0.1 microsecond and each epoch consists of one million time-steps. As there are a total 600 epochs in the growth simulation, the whole simulation is run for 600 million time-steps. The values of the parameters used in the growth simulation run are mentioned in table 4.1.

After the growth simulation, the results were serialized and deserialized to run STDP simulation of varying time intervals. The values of the parameters used in the STDP simulation run are mentioned in table 4.2.

Table 4.1: Graph parameters for BrainGrid Growth Simulation

Parameter	Value
Number of neurons	10,000
Epochs	600
Epoch Duration	100 sec

Table 4.2: Graph parameters for BrainGrid STDP Simulation

No.	Simulation Type	Neurons	Epochs	Epoch Duration (sec)	Total Duration (sec)
1	Growth	10,0000	600	100	600,00
2	STDP	10,0000	1	5	5
3	STDP	10,0000	1	10	10
4	STDP	10,0000	1	20	20
5	STDP	10,0000	1	50	50
6	STDP	10,0000	1	100	100
7	STDP	10,0000	3	100	300

4.7 Hardware and Software Environment

These simulations were run using the GPU version of the simulator on an 2.4GHz Intel Xeon E5-2620v3 system running CentOS Linux version 7.7.1908 and kernel version 3.10.0-1062.4.3.el7.x86_64 using an NVIDIA Tesla K80 GPU with CUDA 10.2 libraries.

Chapter 5

METHOD: BRAIN GRAPH ANALYSIS

Along with the advancement in quantitative analysis of the results of complex network imaging techniques, brain graphs have become an increasingly popular way to study the human brain connectome. Brain graphs can help us answer questions related to the evolution of network structure, the relationship between state, structure, function, and behavior of the network. Due to the various limitations and barriers to accurate imaging the of the human brain by using diffusion-weighted imaging, fMRI, and other techniques [90], having a complete understanding of structural and functional connectivity is extremely difficult. Global network metrics calculated on networks generated using these techniques are sensitive to noise in functional connectivity [88].

We wanted to analyze the topology of a cortical growth network and the stabilized network after pruning the outgrowth using the STDP model. In order to do this, we set up experiments and ran simulations using the BrainGrid simulator. The experimental setup and the parameters for the simulations is provided in section 4.6. This chapter provides the methodology used to investigate the network structure and the effects of STDP on the network connectivity.

5.1 Generating Brain Graphs

Before analyzing the network, we need to generate graphs from the simulation output.The first step to generate a brain graph is to define nodes and edges. In our dataset, each neuron and synapse data is recorded separately which allows us to distinctively represent each neuron as a node and each synapse as an edge. This also ensures that we are not missing any data. We simulated a network of 10,000 neurons in a 100 x 100 rectangular arrangement. We used a standardized layout of endogenously active neurons, inhibitory neurons and excitatory neurons. The layout shown in figure 5.1 represents a 10 x 10, this is repeated to fill the 100 x 100 Kawasaki [59]. We employed slightly different approaches to generate graphs for the growth and STDP simulations as we output different parameters in these simulations. These approaches are discussed in the subsequent subsection.

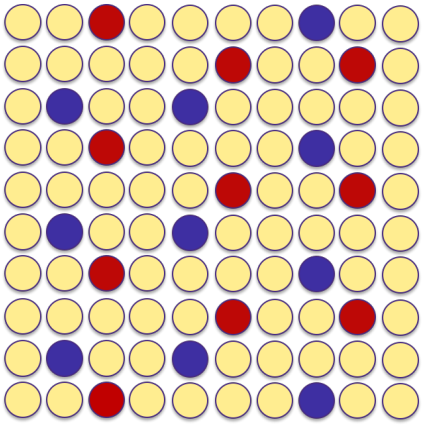


Figure 5.1: Layout for endogenously active neurons (blue dots), inhibitory neurons (red dots) and excitatory neurons (yellow dots). Redrawn from Kawasaki [59].

5.1.1 Growth simulation

For a growth simulation, the simulator outputs the radii of connectivity for each neuron but not the weight of each synapse. Therefore, the weight for each connection is calculated using the area of overlap using the radii of connectivity for each neuron. We calculated the area of overlap from the simulation data by using the radii the connectivity and the location of the neuron. If the area of overlap between two neurons is greater than zero, then a connection or edge exists between those two neurons, as illustrated in figure 5.2. The weights of the connection are calculated by normalizing the area of overlap by a factor of 10^7. These weights are then used to create an adjacency matrix. As the area of overlap is shared between the two nodes, the edge costs the same in both directions and is undirected. Hence, the brain graph generated is an undirected weighted graph. The graph is generated using inbuilt MATLAB functions. As the data is collected for each epoch, we studied the graph as it grew. In the growth simulation, more connections are added at the end of each epoch as discussed in section 4.1.

5.1.2 STDP simulation

In the STDP simulation, we output the weight, the source neuron and destination neuron of a synapse, as discussed in section 4.1. We use this data to generate a graph using MATLAB functions. Unlike, the growth simulation, the graph generated is a directed graph. After generating the graph,

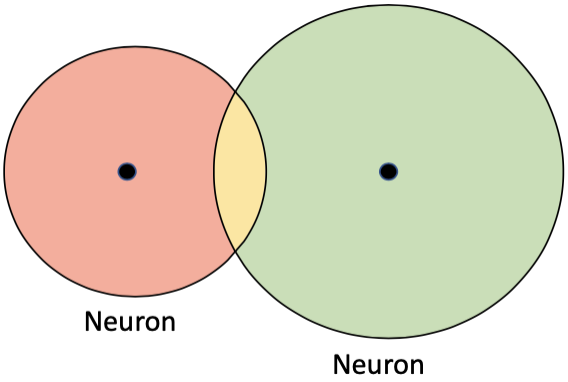


Figure 5.2: Overlapping regions of two neurons indicating that a connection exists between them (re-drawn from Hsu [50]). The overlapping yellow region is used to calculate the synaptic weight by multiplying by a normalizing factor.

Figure 5.3: The workflow for generating brain graphs and analyzing the network.

we calculate different feature values to study the structure and connectivity of the network. The feature values from the growth simulation and the STDP simulation are compared to analyze the changes in the network. The workflow for the analysis is shown in figure 5.3.

5.2 Feature Extraction

For graph analysis of the brain graph both global features and nodal features were used. Global features were used to compare changes in each time step, and nodal features were used to analyze the brain graph extracted at each time step in more detail. Parameter values like location of neuron on the x and y axis, which were used in feature calculations were read from the simulation output.The following section discusses the features used in the analysis and the rationale behind

using these specific features. All the calculations were done in MATLAB. A brief description of these is already provided in section 3.1.

- **Connection Density and average degree:** These are some of the basic features of a graph which we used to check if our graph is a connected graph, and generate a random graph of the same average degree.

- **Clustering Coefficient and Average Path Length:** Clustering coefficient and average path length have been used in brain networks to understand function-structure relationships as they used to calculate small worldness property. Small worldness is defined as the ratio of the clustering coefficient and the average path length. We wanted to measure the small worldness property of our network as brain networks are usually small-world networks [92].

- **Degree Centrality:** Degree centrality measures the importance of a node by measuring its connections. In a social network, a high degree centrality would imply a higher influence in the social circles. In a neural network, a high degree centrality implies that a neuron is capable of forming synaptic pathways with a large number of neurons and has access to more areas of the network. We wanted to calculate this property to see if the location of a neuron in our network has any impact on the centrality values.

- **Betweenness Centrality:** Betweenness centrality measures the importance of a node in information transfer within the network. As information passing is one of the most important functions of the human nervous system, we wanted to measure the impact of a node in the network's message passing system.

- **PageRank Centrality:** We wanted to measure the transitive influence of a node in the network and take into account the direction of the edge as we generate a directed graph in the STDP simulation. We used the PageRank centrality for this purpose. We calculated page rank using MATLAB libraries and the algorithm is a little different from the one discussed in section 3.1. In MATLAB implementation, page Rank centrality is calculated by a random walk of the network and calculating the time spent at each node during the walk. The next node is chosen from the neighbors of a edge with a follow probability 'p', if a node has no

edges then the next node is chosen from all nodes. The average time spent at each node is the centrality score [68].

- **Distance from Endogenously active neurons and Inhibitory neurons:** Due to the layout of our neural network, the distance of each neuron from inhibitory and endogenously active neurons is different. We wanted to analyze how the distance from a certain type of neuron affects other network parameters like betweenness.

5.3 Representing and analyzing network features

Due to the large size of the data collected, it was impossible to study or compare it directly. We used graphical data representation to convert the data into a more comprehensible format. We used the location of a neuron on x and y axis as a primary identifier while comparing the values of different features. In some feature values like the synaptic weight, there was huge disparity in the maximum and minimum value. This made visualizing data difficult while using images with scaled colors as the smaller values were barely visible. We normalize weights using log transformation. This allowed us to better visualize connections with minuscule weight values without changing the network pattern. As $log0$ is undefined, we updated the value of the weights by adding the minimum weight value greater than zero. The results of the network analysis are presented in the next chapter.

5.4 Correlation between different centrality measures

We wanted to find out the correlation between different centrality measures to investigate whether these metrics are redundant and if they measure different aspects of connectivity. We also wanted to find out the effect of local measures on global measures. Degree centrality is an example of a local measure as it only takes into account the adjacent nodes. Betweenness centrality is an example of a global measure as it takes into account the paths from all the nodes to the goal node. A high correlation would indicate that the different measures are redundant and a low correlation would indicate that they represent different properties, and hence, serve different purposes. We used the work for Valente et al. [85] for comparison of our correlation coefficient values and they used the Pearson coefficient. We used Pearson coefficient because we wanted to check if the coefficient values for the centrality measures of our dataset were within the permissible range and find out the linear

correlation between them. Pearson correlation coefficient between datasets x and y is calculated using equation 5.1.

$$\rho(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5.1}$$

where $\bar{x}$ is the mean of dataset x and $\bar{y}$ is the mean of dataset y.

5.5 Hardware and Software Environment

All graphical was done in MATLAB using its built-in graph and network analysis toolboxes. MATLAB version R2020b was used. All MATLAB scripts were run on a 2.4 GHz Quad-Core Intel i7 system.

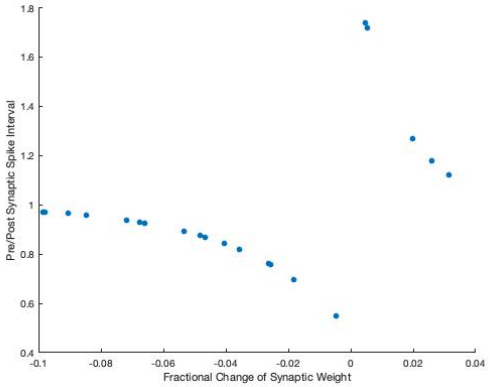
Chapter 6

RESULTS AND DISCUSSION

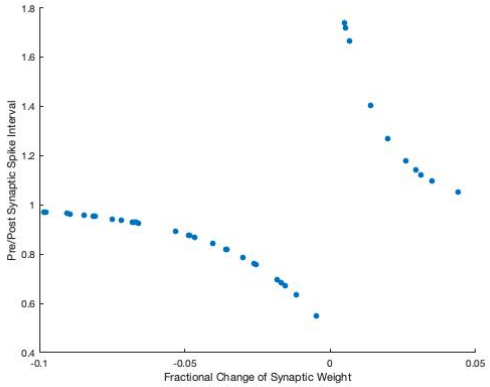
In this chapter we present the results for the following experiments: 1) STDP verification; 2) the visualization of the full network and centrality measures; 3) distributions of centrality measures, and 4) correlations between centrality measures. The brain graphs were generated using simulation results from the BrainGrid simulator [78]. Detailed analysis has been done for degree, eigenvector, PageRank, and betweenness centrality measures. Hubs, closeness and authority centrality measures have just been used to study the correlation between different measures.

6.1 STDP Model Verification

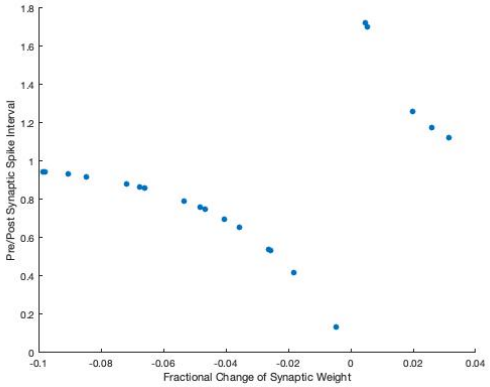
STDP model verification included a CPU-based experiment using the single-threaded version of the code. We monitored the fractional change in weight and the time difference between pre- and post-synaptic spikes, and plotted graphs to study their correlation. The experiments followed the workflows mentioned in section 4.4. We implemented multiple STDP models and we ran experiments to verify each of them. The details of all implementations are given in section 4.2. The first experiment was based on the basic classical additive STDP implementation by Song et al. [80]. The second experiment was based on the suppression additive model by Froemke and Dan [35]. The third experiment was similar to the first one but the values of parameters like A_+, A_- were taken from Kempter et al. [61]. We also implemented the multiplicative model for STDP. As we can see in figure 6.1, all experiments except for multiplicative STDP match the results from the source literature. This verifies the implementations for additive model from Song et al. [80], Froemke and Dan [35], and Kempter et al. [61]. The multiplicative STDP that we implemented used size dependent plasticity rules on a single cell simulation receiving random inputs which yielded very different weight distributions than the additive models [86, 54]. This model's explanation in the literature was unclear and we could not develop a good algorithm from it. For future work, we can try implementing a different multiplicative STDP model which has a more detailed algorithm. We used the STDP model from Song et al. [80] for subsequent simulations and analysis.



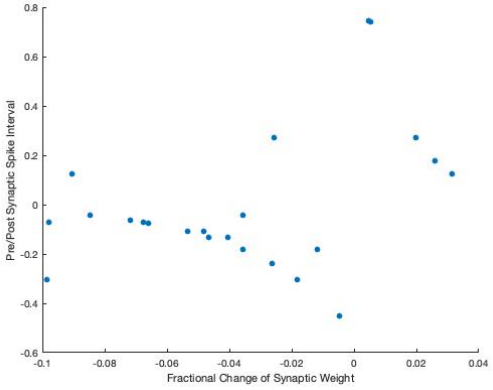
(a) Additive STDP model recreated from Song et al. [80]



(b) Suppression additive model recreated from Froemke and Dan [35]



(c) Additive STDP model recreated from [61]



(d) Multiplicative STDP model recreated from [86]

Figure 6.1: STDP verification test results. The calculated fractional changes of weights were plotted against the spike pair time intervals.

6.2 Graph Analysis of BrainGrid Growth Simulation

Initial analysis was done on the network obtained from the growth simulation. The graph generated was an undirected weighted graph. The growth simulation was equivalent to 28 days development *in vitro*. The brain graph was generated using the weights at the end of the simulation. As we can see in figure 6.2, most of the dots which represent a connection, are close to the diagonal, within a small area showing connections with other nodes with indices close to any given node's index. The layout of the neurons in our network and the small radii of connectivity causes repetition in the

Table 6.1: Graph parameters for growth simulation (last epoch). Note that this is an undirected graph, so the number of synaptic connections (and average degree) is twice the number of edges (and average edges).

Number of Edges	176541
Connection Density	0.00353082
Average Degree	35.3082

pattern of connections, and there are no long-range connections in the network. Table 6.1 shows the computed values of basic metrics for the brain graph generated using the data for the last epoch of the growth simulation. As we can see in that table, the network has an average degree of 35. As the average degree is greater than 1, the graph is strongly connected. This is especially important as all nodes should be connected to make information passing possible throughout the network.

Brain networks have high clustering coefficient, low average path length and high small-worldness index value [22]. Since the results from fMRIs of human brains correspond to specific brain regions and different neuron sizes, we were unable to compare their parameter values to our network and translate the implications of variations in network structure. In order to check if our network has significantly higher small-worldness index, we instead used a random graph model as baseline for comparing our network's connectivity parameters. We generated an Erdős–Rényi random graph with the same number of edges. For this comparison, we calculated the average path by ignoring the weights of the edges, considering the weight as 1 if a connection exists between two nodes and 0 if there is no connection (in other words, treating the graph as unweighted). As we can see in table 6.2, the clustering coefficient, average path length and small worldness of our network is higher than the random graph. The average path length of our network is $L = 7.575$ which means that on average we would need more than 7 hops to travel from one node to another. The small world parameter, $sw = C/L$, should be greater than one for the network to be a small world network [57]. For our network, sw is not close to one, but still is approximately 18 times larger than the value for a random graph. Our network is more closely connected and has higher small-worldness index, which are characteristic of brain networks. Hence, our network is more like a brain network than a random network.

Table 6.2: Comparison of network parameters for random graph and growth graph.

Graph	Clustering Coefficient (C)	Average Path Length (L)	Small Worldness (C/L)
Growth	0.5834	7.575	0.077
Random	0.0035	0.8178	0.0043

6.2.1 Network Growth

Connection growth was analyzed from the first to last epoch of the growth simulation to study the growth pattern. Connections don't start forming until the 220^{th} epoch, i.e 10 days *in vitro* (DIV). As we have a fixed repetitive layout for neurons, the connection pattern repeats every 100 neurons. Thus, the graph starts growing in a uniform pattern, as described previously. This is consistent with our simulated network starting with no connections and growing them during development. The different types of neurons are distributed regularly throughout the network, the details of this arrangement are discussed in section 5.1.

6.2.2 Centrality Values

Centrality has been used in network theory to quantify the role of a node in the overall structure of the network. Different definitions of centrality relate to different network properties and are applicable in specific contexts. Brain network centrality has been used to detect changes in functional brain networks due to various diseases like diabetic nephropathy and retinopathy and Alzheimer's disease [91, 18]. Of the existing centrality measures, we calculated the degree, eigenvector, PageRank, and betweenness centrality values for the network. The reasons for choosing these are described in section 5.2. Centrality values were calculated at the end of the growth simulation to study the network properties for the fully grown network. We visualized the values based on their location in the network layout. Our goals were to analyze how the values were distributed, identify nodes with high centrality values, or *hubs*, and study the possible reasons behind behind such observations.

As can be seen in Fig. 6.3, degree centrality does not seem to vary based on neuron location; there are neurons with very low and very high centrality throughout. This shows that degree centrality values are stratified and don't follow a particular pattern. Due to small radii of connectivity,

repetition of neuron layout every 100 neurons and uniform connectivity pattern throughout the network, neurons across the network have equal probability of forming a large number of connections. The neurons at the border of the network have sightly lower values due to edge effect (reduced number of neighbors). Except for neurons located at the border of the network, the degree is not dependent on neuron location. The maximum value of degree centrality was 52, which means that the most central nodes according to this measure were connected to 52 other neurons. Considering that all connections were short range, these central nodes formed extremely dense connections. We can see in figure 6.4 that the highest number of neurons lie in the histogram buckets for degrees of 40 to 50, indicating that most neurons are well connected, but that there is also a significant subset of neurons with low connectivity, with degree less than 10.

We can see in figure 6.5 that the points with highest eigenvector centrality correspond to neurons that are centrally located in the network. A node's eigenvector centrality is calculated recursively and depends on all nodes connected to it directly or indirectly, but the effect of those values decreases as the distance from the node increases. Since the nodes in the middle will have the minimum mean distance from other nodes, it is likely that these have high eigenvector centrality values. Figure 6.6, which shows the distribution of eigenvector centrality for our network, is similar to the distribution of the human brain graph generated by Zuo et al. [96] and reproduced in Figure 3.7. This shows that the distribution of eigenvector centrality for the growth network generated using BrainGrid is consistent with the distribution in the human brain.

PageRank centrality values are shown in figure 3.4. In the figure, we can see that the points are scattered, and the plot resembles that of degree centrality in figure 6.3. The distribution of PageRank values, shown in figure 6.8), is also similar to the distribution of degree centrality. The resemblance exists because PageRank centrality in an undirected graph is statistically similar to degree centrality [32]. High centrality values occur throughout the graph rather than just at the center due to the repetitive connectivity patterns in our network.

The nodes which are at the center of the location matrix have the highest betweenness values, as shown in Figure 6.9, where points in the center of the plot are highest. This is due to the fact that these neurons are most likely to lie on the shortest path between two nodes due to their location. Neurons located on the border of the network have almost zero centrality values as they are least likely to lie on a shortest path. We can see in figure 6.10, betweenness centrality has an exponential distribution, with most of the neurons having zero betweenness. This suggests that information

passing in the network cold be controlled by a few central neurons.

Betweenness centrality measures the importance of nodes in information passing in the network; as this is a primary function of neural networks, we present a more detailed analysis of this centrality measure in the following section.

6.2.3 Betweenness Centrality

Figure 6.11 shows the betweenness values at different points in time during network growth. Around epoch 110 (5 DIV), we can see that the distribution is irregular and the centrality values are very high as there are few paths in the network. Betweenness centrality is the ratio of the number of shortest paths passing through a node to the total number of shortest paths; having less number of paths results implies that the denominator value is low and thus, results in high centrality values. We can see an interesting pattern around epoch 170 (8 DIV) the neurons located at the diagonal achieve significantly higher betweenness values than the other neurons. At epoch 230 (10 DIV), the distribution becomes smoother, and is mildly peaked in the middle and somewhat lower at border of the network. Around epoch 290 (14 DIV), betweenness values become strongly peaked at the middle of the network layout. After epoch 360 (17 DIV), the distribution becomes less peaked in the middle and is uniformly low at the border. We can also observe that few neurons have high betweenness values and we see the emergence of hubs.

Kawasaki and Stiber [60] analyzed the variance of the radii of connectivity as a function of DIV. Since, neuron connections depend on the radii of connectivity and betweenness values depend on network connections, it would be interesting to see how the radii of connectivity affects betweenness centrality values. We can investigate the correlation between these features in future work.

Figure 6.12 shows the distribution of neurons based on betweenness centrality values and the distance of a node from the nearest border; we can see in the graph the bins representing zero distance from the border have zero or extremely low betweenness values. This shows that the majority of neurons at the border of the network have zero betweenness plot confirms the effects of edge effect. The location disadvantage makes it difficult for these neurons to lie in the shortest path connecting two node. The nodes which are centrally located have location advantage, and a high possibility of being in the shortest path between two nodes. Thus, the highest betweenness values are achieved by these neurons.

Table 6.3: Correlation between betweenness centrality at 28 simulated DIV and distance from the nearest endogenously active neuron, inhibitory neuron, and border.

Distance calculated from	Correlation Coefficient
Endogenously Active Neuron	0.138
Inhibitory Neuron	-0.049
Nearest border	-0.044

We tried to study the effect of the distance from endogenously active and inhibitory neurons. In order to accomplish this, we calculated the Pearson correlation. Table 6.3 shows that all of these factors have extremely low correlation coefficient, and thus, we can conclude that these factors do not affect betweenness values. Betweenness centrality is uniformly low at the border of the network, but away from the border, the distance from the border has no correlation with the centrality values. In order to confirm this, we visualized graphs where neurons were binned based on both betweenness and distance. The results from this aligned with the inferences drawn from the edge effect and the clustering coefficient i.e. most neurons with distance zero from the border had almost 0 values and the rest of the distribution was irregular.

6.2.4 Correlation between different centrality measures

We calculated the correlation matrix for a broad array of centrality measures to evaluate the strength and direction of the linear relationship between them. We used Pearson coefficient for this. We can see in Table 6.4 that the coefficient values for the centrality measures of our dataset were moderately correlated and within the permissible range based on the results by Zuo et al. [96]. Since, none of the results were beyond the expected range based on the results by Zuo et al. [96], we did not expand our analysis and use a different correlation coefficient. PageRank and degree centrality have the highest correlation at 0.98, which is in line with the fact that these two parameters have similar values for undirected graphs and this indicates that in a uniform network high PageRank values are obtained by neurons which are highly connected to other neurons. The lowest correlation is between degree and closeness at 0.22.

Table 6.4: Correlation Matrix for different centrality measures for growth simulation

Centrality		**1**	**2**	**3**	**4**	**5**
1	Degree	1	0.48	0.22	0.53	0.98
2	Betweenness	0.48	1	0.51	0.60	0.44
3	Closeness	0.22	0.51	1	0.73	0.14
4	Eigenvector	0.53	0.60	0.73	1	0.46
5	PageRank	0.98	0.44	0.14	0.46	1
Average		0.64	0.60	0.52	0.66	0.60
Standard Deviation		0.33	0.22	0.35	0.21	0.37

6.3 Graph Analysis of STDP Simulations

After the growth simulation, we serialized the resultant weights to a file so that they could be deserialized into subsequent simulations of network pruning using STDP. The brain graphs generated were weighted directed graphs. We can see that the number of points (connections) in figure 6.13 is significantly less than in figure 6.2, reflecting the decreased number of connections. It is difficult to visualize the change in individual synaptic weight values for all nodes. In figure 6.14, we visualize connections for four nodes in the middle of the network for the growth network (a), and after STDP simulations of duration 5s (b) and 300s (c). The 4 nodes selected had (x, y) coordinates (48, 49), (48, 50), (49, 49), and (49, 50). We can see that for a 5 second run, all connections remain intact but the symmetry of the undirected growth graph breaks: after STDP, the weights of the connections from node 1 to 2 increases and 2 to 1 decreases. After and STDP simulation of 300 seconds, the connections between nodes 4 to 3 are severed. From these observations, we can infer that even a very small duration of STDP changes weights and longer duration STDP leads to pruning of connections. In our simulation, we remove a connection if the weight decreases to zero. The change in weight caused by each pair of (presynaptic, postsynaptic) spikes is small and thus it takes time for connections to be removed. This proves our hypothesis of symmetry breaking.

6.3.1 Centrality values

Centrality values for the STDP model were calculated for the same reasons as for the growth model and to study the change after pruning. We calculated indegree, outdegree, PageRank, and betweenness centralities for the network after 5 and 300 seconds of STDP simulation. As our graph is now a directed graph, we calculated indegree and outdegree centrality instead of degree centrality. We have not used eigenvector centrality for STDP analysis as it cannot be calculated for directed graphs without changing the original formula or symmetrizing the network.

We can see in figure 6.15 that the plot for indegree centrality at 5s (left) is similar to the plot for degree centrality in the growth simulation (fig 6.3), but after 300s (right) we see major changes in the distribution and the plot is no longer stratified. As shown in figure 6.16, the histogram bin for the highest indegree centrality have very low size after a 300s STDP run, whereas the bin sizes representing low centrality values increase. This shows that after pruning the network using STDP and removing connections with low synaptic weights i.e. weak connections, few hubs or nodes with high centrality value remain.

The changes in outdegree centrality are similar. In figure 6.17, we observe that the after 5s of STDP (left), there isn't much difference in the plot compared to the growth simulation. After 300s (right), change is apparent, although not as prominent as for indegree centrality. In the distribution plot in figure 6.18, we can see that the size of the bins for highest outdegree centrality decreases after 300s compared to the growth simulation.

Figure 6.19 shows the PageRank centrality values for the STDP simulation; we can see that compared to the growth simulation, shown in figure 6.7, PageRank changes drastically. In an undirected graph, PageRank centrality is dependent on degree centrality [32]. After STDP, our graph is no longer undirected; thus, the PageRank centrality graph loses its resemblance to degree centrality and the graph is no longer stratified. As we can see in figure 6.20, the majority of neurons have close to zero centrality value as the bin size for low centrality is the highest, and the number of neurons with high centrality values decreases. This suggests the emergence of a small number important nodes or hubs.

The graph for betweenness centrality value after STDP, shown in figure 6.21, is similar to the graph after the growth simulation, shown in figure 6.9. We see that the values increase gradually as we move towards the center of the lattice and a small number of neurons at the center of the

Table 6.5: Correlation between betweenness centrality after 300s STDP run and the distance from the nearest endogenously active neuron, inhibitory neuron, and border.

Distance calculated from	Correlation Coefficient
Endogenously Active Neuron	0.082
Inhibitory Neuron	-0.010
Nearest border	-0.023

Table 6.6: Parameters of Normal Distribution for Correlation Coefficient between centralities

Average correlation coefficient	0.5212
Standard Deviation of correlation coefficient	0.2728

network have high betweenness centrality values. Unlike other centrality measures, the distribution for betweenness centrality does not change after the STDP run. As we can see on figure 6.22, after the STDP run the histogram bin values change in magnitude compared to the growth simulation (fig 6.10, but the overall distribution shape remains the same.

6.3.2 Betweenness Centrality

Figure 6.23 shows the distribution of betweenness centrality values as a function of node distance the nearest network border for a 5s (left) and 300s (right) STDP simulation. We can see in the graphs that the bins representing zero distance from the border have zero or extremely low betweenness values. Betweenness centrality for STDP simulations have low correlations with distance of the neuron from the nearest endogenously active neuron, inhibitory neuron, or network border, as we can see in table 6.5. These results are similar to the growth model and indicate that STDP pruning isn't dependent on where a neuron is located with respect to an endogenously active or inhibitory neuron.

6.3.3 Correlation between different centrality measures

As we can see in 6.6, the average of the correlation coefficients was 0.52 and the standard deviation was 0.27. These values were used for data sanity check and they were within the expected range based on the results by Zuo et al. [96]. This indicated that the centrality measures for our simulation are moderately correlated and thus, we can rule out redundancy which would have been a problem if our measures were highly correlated.

Table 6.7 presents detailed correlation information. Indegree centrality had the strongest average correlation at 0.59 and incloseness centrality had the weakest average correlation at 0.23. This infers that the number of connections of a node has more influence on the centrality measures than the distance from other nodes. Correlations among specific centrality measures lay between -0.055 to 0.84. Hubs and authority centrality had the highest correlation at 0.84. This was expected as these two metrics are codependent. Degree centrality had the second highest correlation to hubs centrality at 0.80, indicating that neurons with high degree are more likely to become hubs in the network.

Degree centrality and PageRank were strongly correlated in the network after the growth simulation. After the STDP run, the coefficient value between PageRank and indegree was 0.77, and PageRank and outdegree was 0.56. This indicates that the correlation is still strong but not as dependent as before. PageRank and incloseness centrality had the smallest correlation coefficient value at -0.055. This indicates that a node's importance in terms of PageRank is unrelated to the path traversed to get to the node. Even if a node has a large average path length, if it is connected to important nodes, it will have a high PageRank centrality. Indegree and incloseness had the second weakest relation at 0.17. This shows that even the number of incoming edges have little influence on the path length of a node.

Table 6.7: Correlation Matrix for different centrality measures for STDP simulation

Centrality		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	Outdegree	1	0.69	0.54	0.51	0.17	0.80	0.56	0.56
2	Indegree	0.69	1	0.42	0.51	0.19	0.69	0.84	0.77
3	Betwenness	0.54	0.42	1	0.41	0.40	0.38	0.28	0.33
4	Outcloseness	0.51	0.51	0.41	1	0.51	0.39	0.37	0.31
5	Incloseness	0.17	0.19	0.40	0.51	1	0.19	0.21	-0.055
6	Hubs	0.80	0.69	0.38	0.39	0.19	1	0.84	0.39
7	Authorities	0.56	0.84	0.28	0.37	0.21	0.84	1	0.45
8	PageRank	0.56	0.77	0.33	0.31	-0.055	0.39	0.45	1
Average		0.55	0.59	0.39	0.43	0.23	0.53	0.51	0.39
Standard Deviation		0.23	0.24	0.21	0.20	0.29	0.26	0.27	0.29

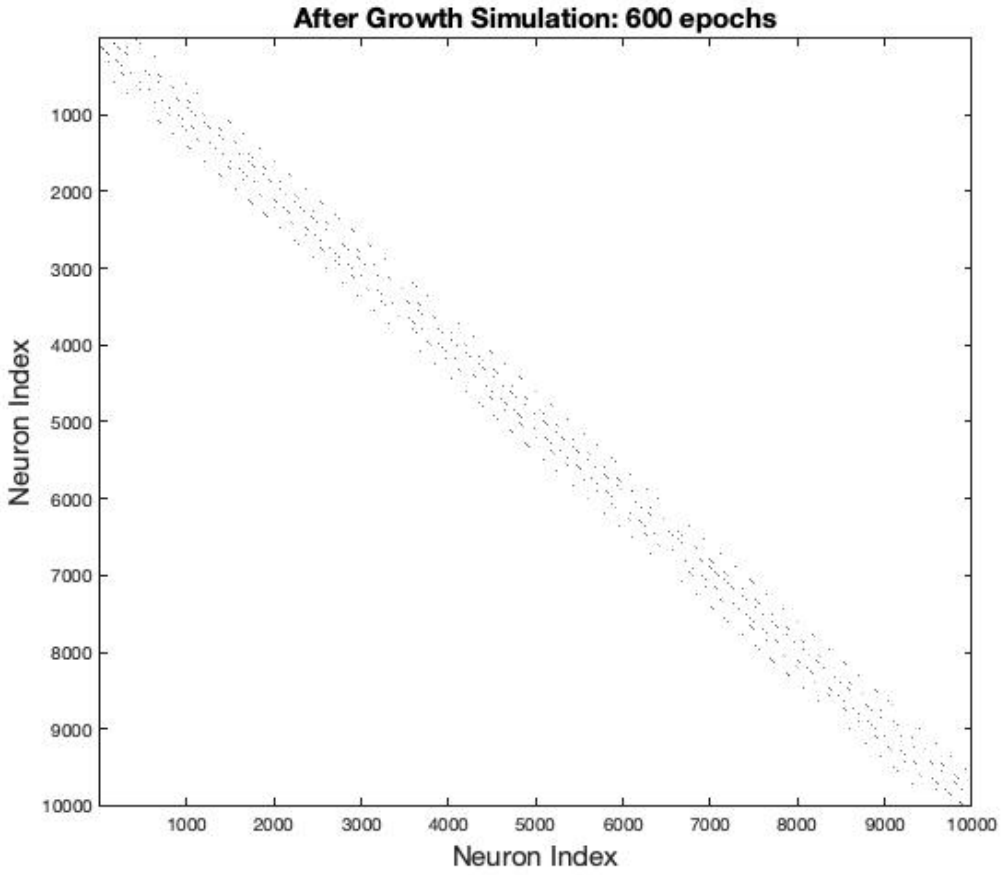


Figure 6.2: Edge distribution in the network at the end of the simulation. X and Y axes are neuron indices for all 10,000 simulated neurons. Points plotted indicate connections between neurons. While point color indicates edge weight (synaptic strength), limited resolution in this graph precludes presentation of the full detail of the 10,000 by 10,000 connection matrix.

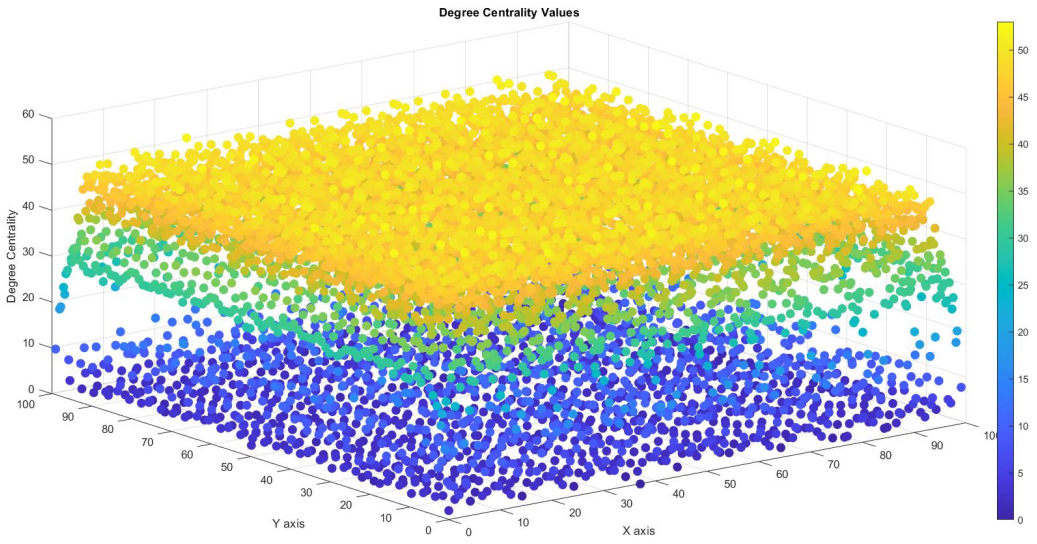


Figure 6.3: Degree centrality for all neurons at the end of the growth simulation. X and Y axes correspond to neuron (x, y) location. Points plotted indicate degree centrality value for each neuron. Centrality values are shown as both as the Z coordinate and as point color. Degree centrality values don't follow a particular pattern and neurons throughout the network have varied values. Due to small radii of connectivity and repetition of neuron layout after every 100 neurons, neurons around the network have equal probability of forming a large number of connections.

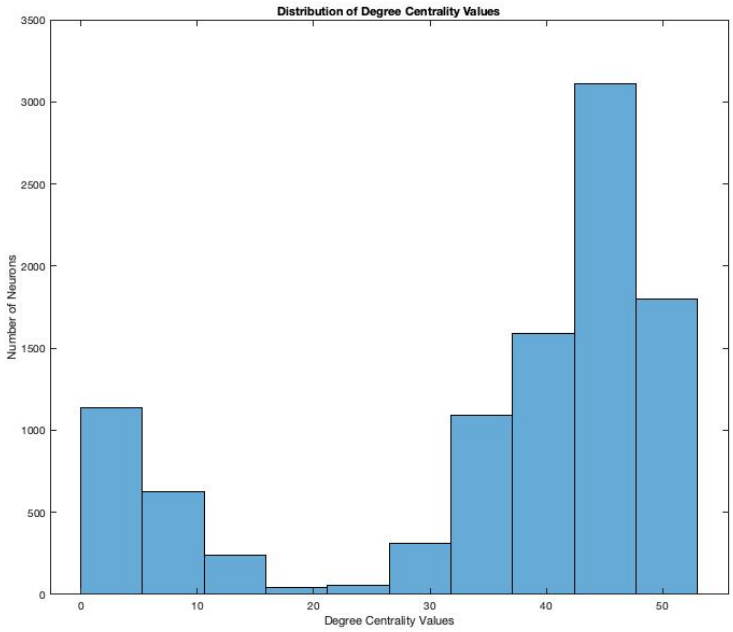


Figure 6.4: Distribution of degree centrality for all neurons at the end of growth simulation. X axis represents the indegree centrality value for the histogram bins and Y axis shows the number of neurons in the bin.

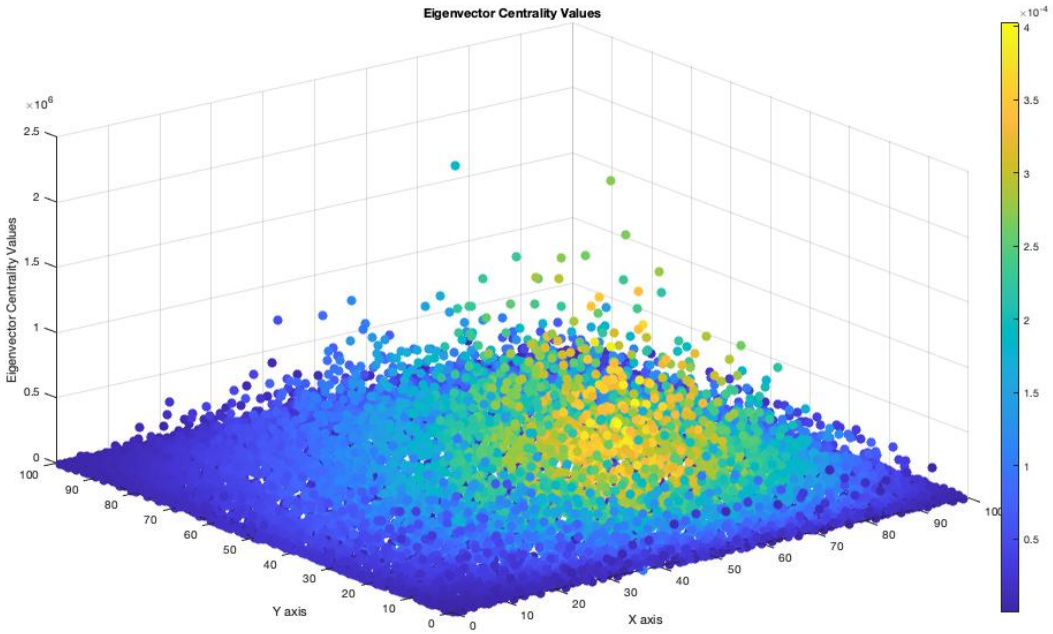


Figure 6.5: Eigenvector centrality for all neurons at the end of growth simulation. Axes, etc. same as figure 6.3. Highest centrality values correspond to centrally located neurons. Eigenvector centrality depends on a node's neighbors eigenvector centrality, and is calculated recursively. Due to this recursive calculation, nodes located at the center of the matrix with least average distance from the other nodes have high eigenvector centrality values.

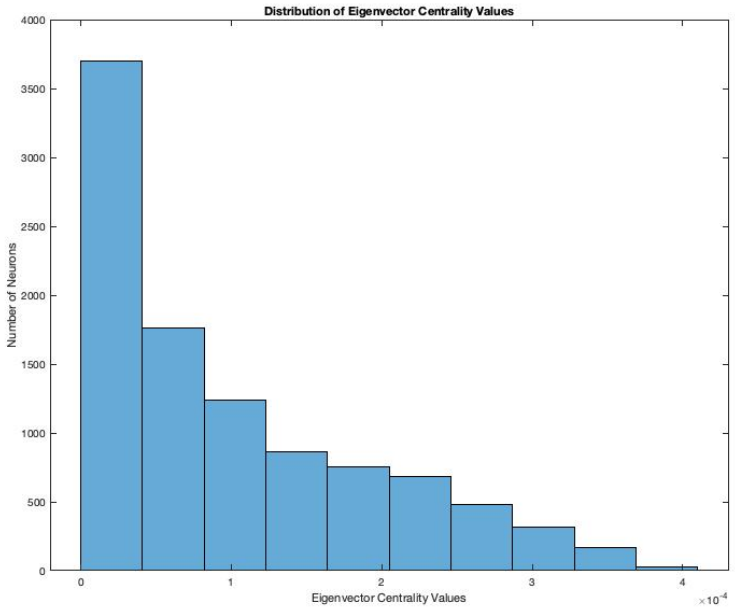


Figure 6.6: Distribution of eigenvector centrality for all neurons at the end of growth simulation. Axes, etc. same as figure 6.4.

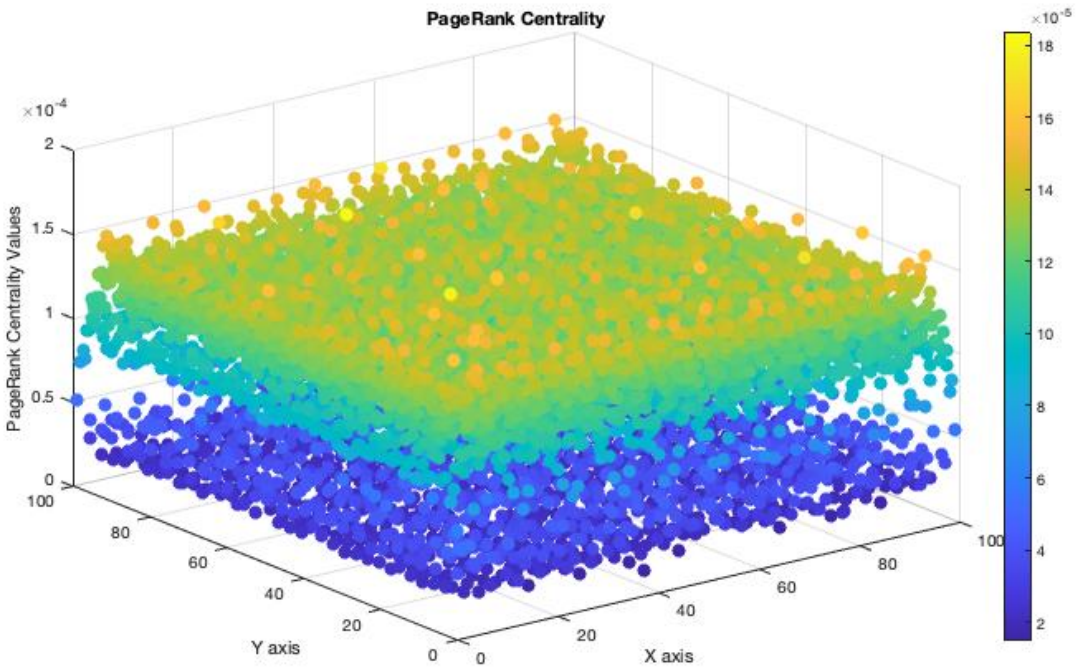


Figure 6.7: PageRank centrality for all neurons at the end of growth simulation. Axes, etc. same as figure 6.3. PageRank centrality value distribution is similar to degree centrality, as PageRank centrality in an undirected graph is statistically similar to degree centrality [32].

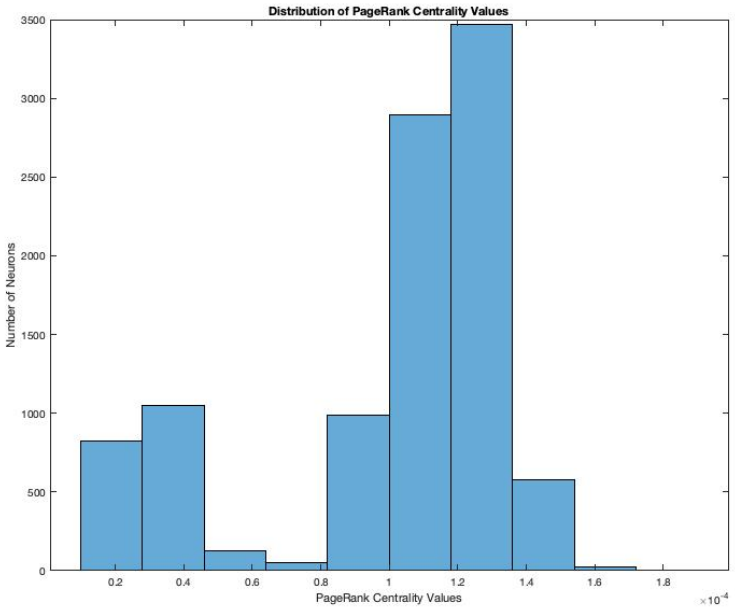


Figure 6.8: Distribution of PageRank centrality for all neurons at the end of growth simulation. Axes, etc. same as figure 6.4.

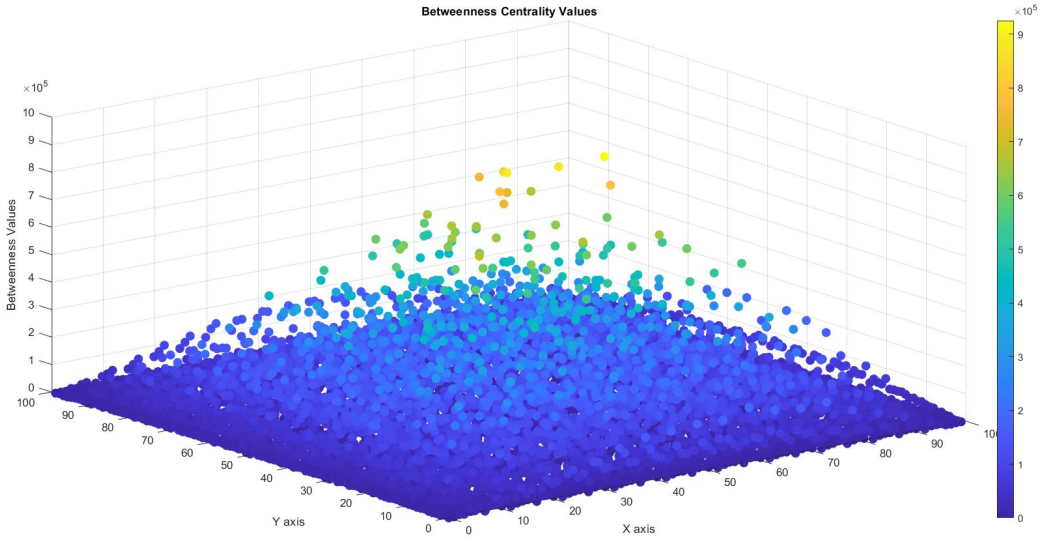


Figure 6.9: Betweenness centrality for all neurons at the end of growth simulation. Axes, etc. same as figure 6.3. Nodes at the center of the neuron matrix have the highest betweenness values, as they are most likely to lie on the shortest path between two nodes due to their location.

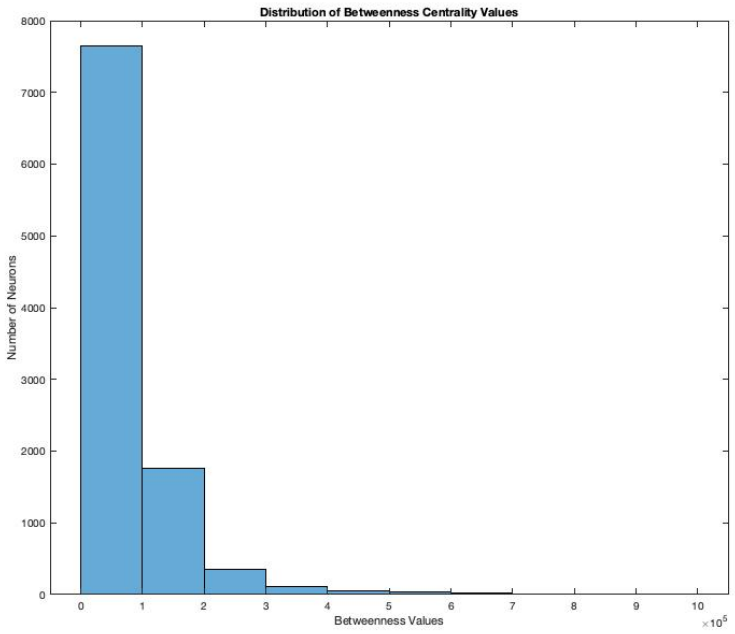
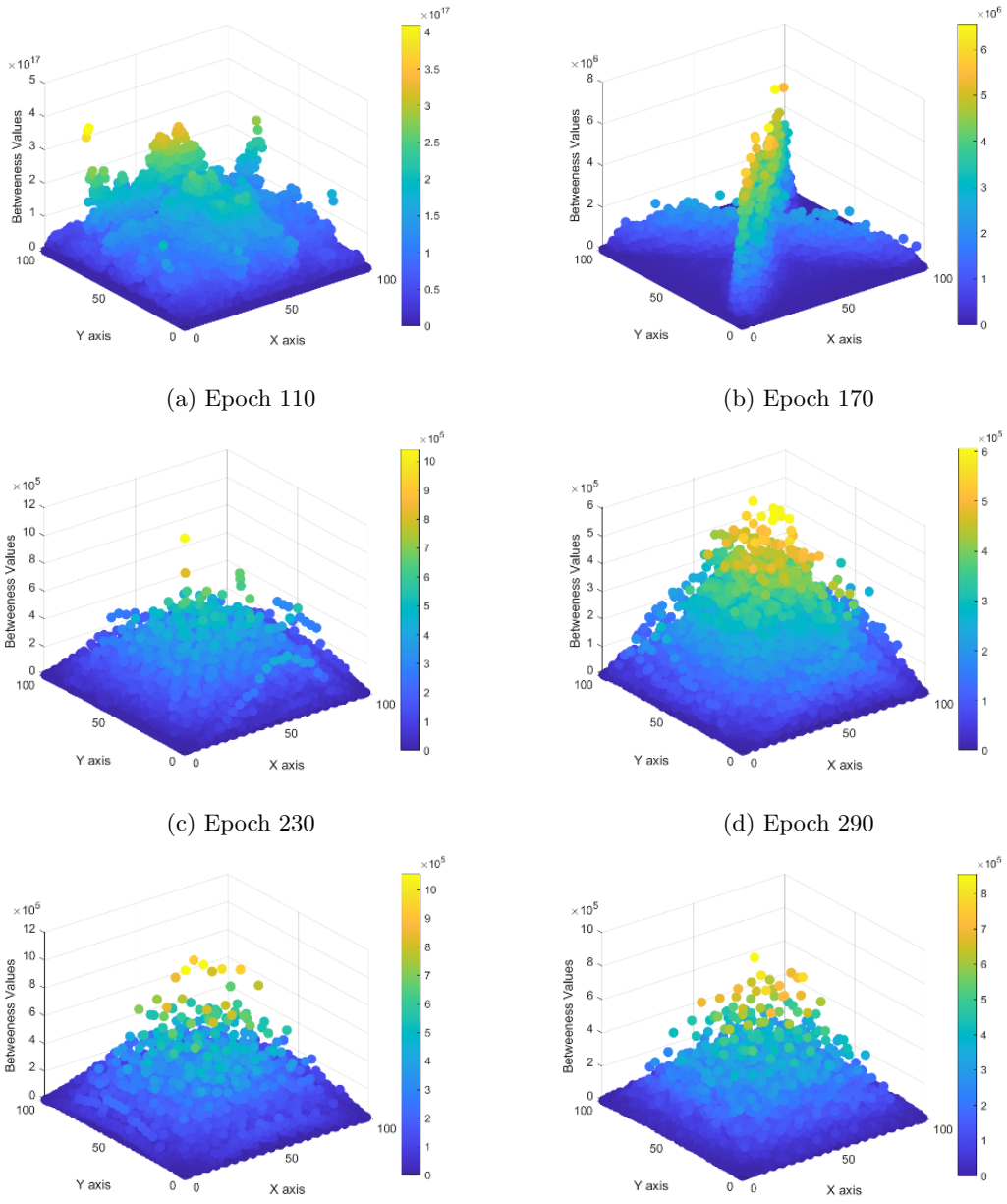


Figure 6.10: Distribution of betweenness centrality for all neurons at the end of growth simulation. Axes, etc. same as figure 6.4.



(a) Epoch 110

(b) Epoch 170

(c) Epoch 230

(d) Epoch 290

(e) Epoch 350

(f) Epoch 410

Figure 6.11: Betweenness values at different points in time during network growth. For each individual plot, X and Y axes are neuron (x, y) location. Points plotted indicate betweenness centrality values for each neuron. Centrality values are shown as both as Z coordinate and point color. Images demonstrate the changes in the distribution during development: (a) 5 DIV, (b) 8 DIV, (c) 11 DIV, (d) 14 DIV, (e) 16 DIV, and (f) 19 DIV.

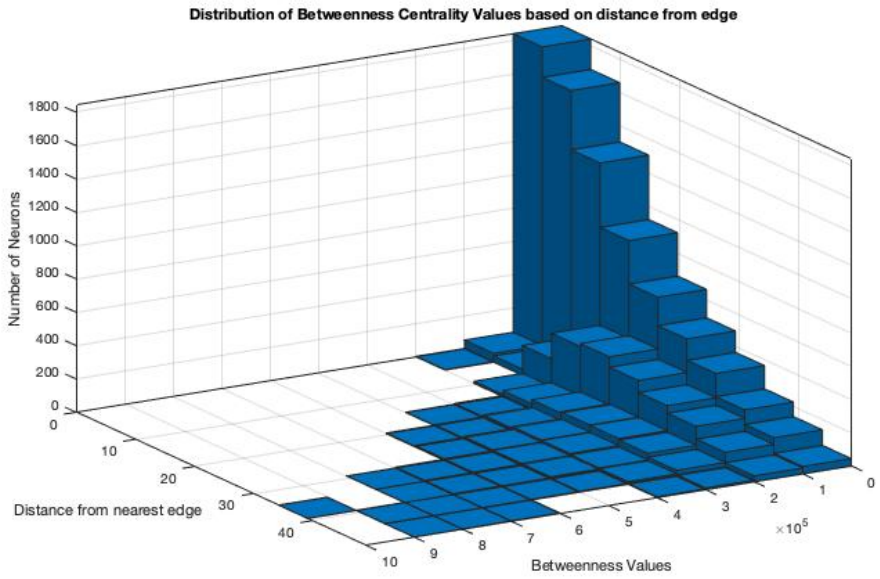


Figure 6.12: Histogram of betweenness centrality as a function of distance of a neuron from the nearest network border after growth simulation. X axis shows betweenness value, Y axis shows distance of the neuron from nearest border of the network, and Z axis shows neuron count.

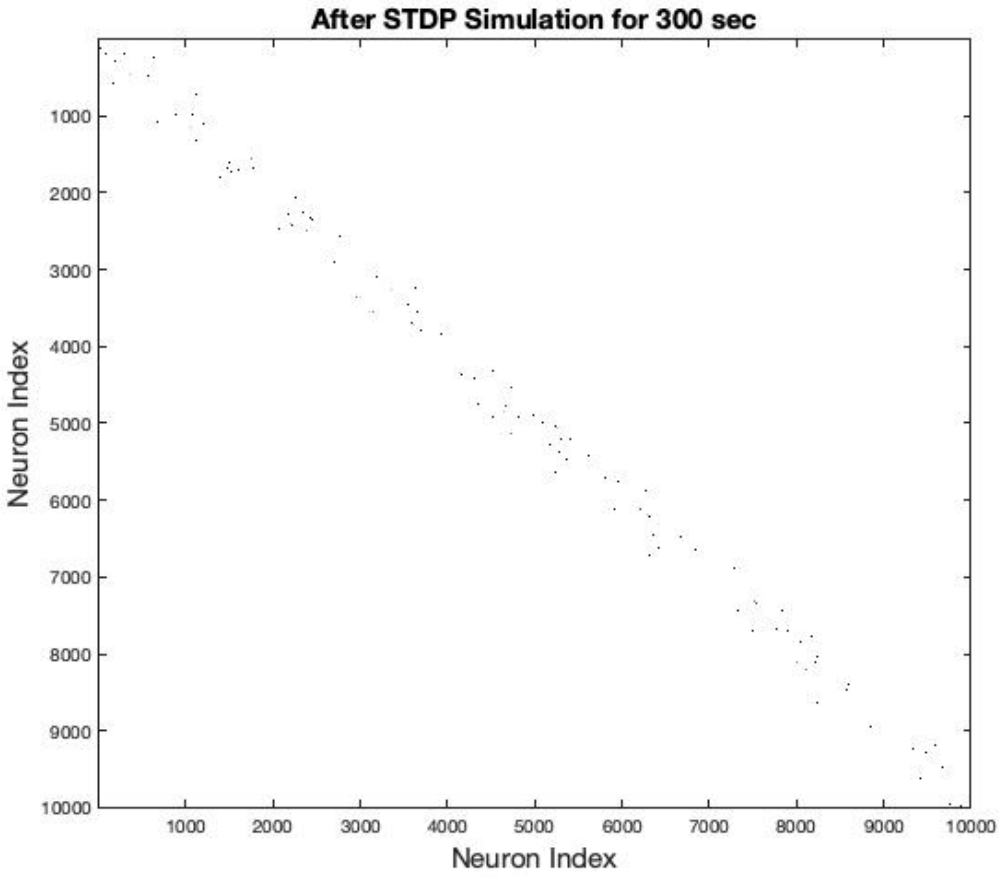
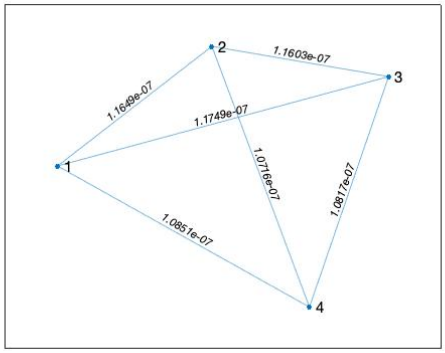
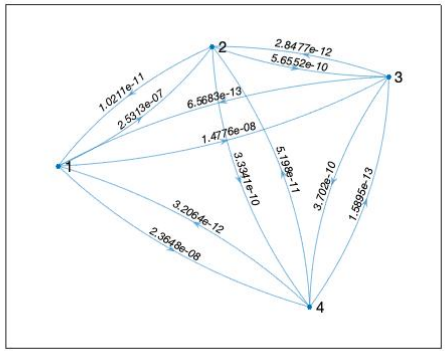


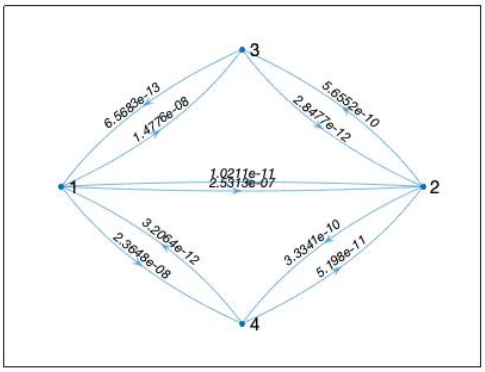
Figure 6.13: Edge distribution in the network after 300s STDP simulation.



(a) After growth simulation



(b) After STDP for 5 sec



(c) After STDP for 300 sec

Figure 6.14: Change in connections between four central nodes. Points denote nodes and lines connecting them represent edges. Graph is undirected in (a). After 5s (b) or 300s (c) STDP simulation, weights in one direction strengthen while weights in the opposite direction decrease.

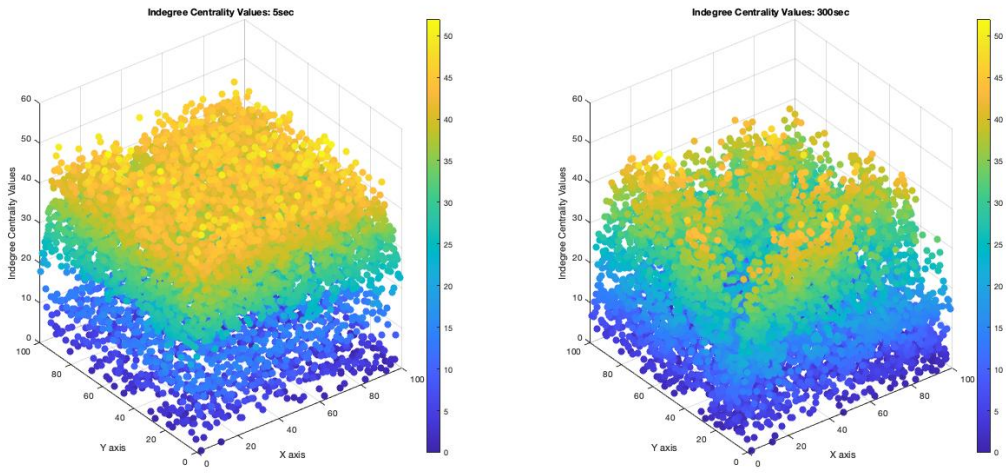


Figure 6.15: Indegree centrality for all neurons after 5s (left) and 300s (right) STDP. X and Y axes correspond to neuron (x, y) location. Points plotted indicate indegree centrality value for each neuron, as both as the Z coordinate and color. Degree centrality values don't follow a clear pattern and neurons throughout the network have varied values. Distribution changes as the network is pruned and the number of neurons with high indegree centrality decreases.

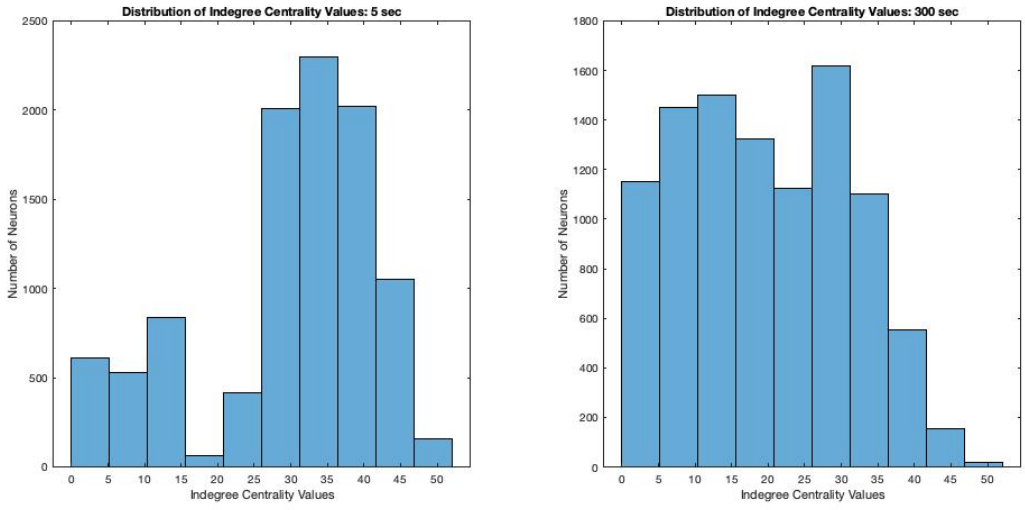


Figure 6.16: Indegree centrality histograms for all neurons after 5s (left) and 300s (right) STDP.

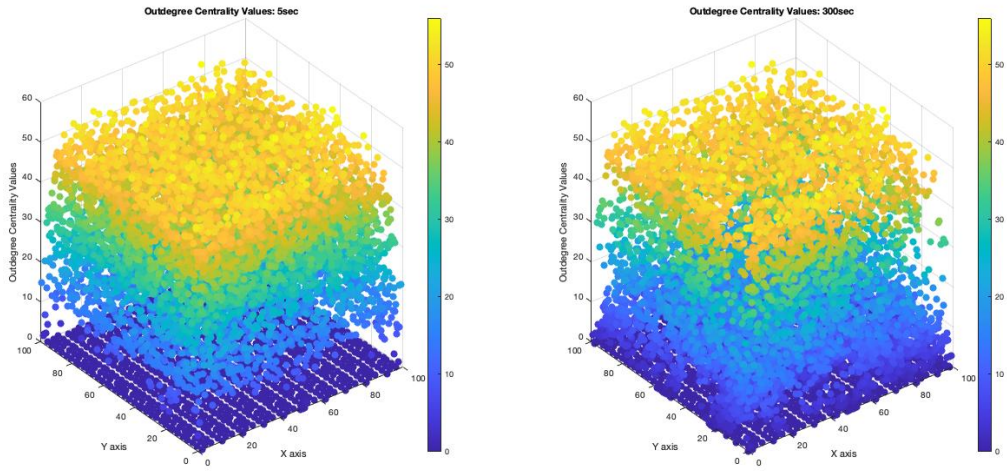


Figure 6.17: Outdegree centrality for all neurons after 5s (left) and 300s (right) STDP. Axes, etc. same as figure 6.15. The range of outdegree centrality doesn't change drastically but the number of neurons with high outdegree centrality decreases.

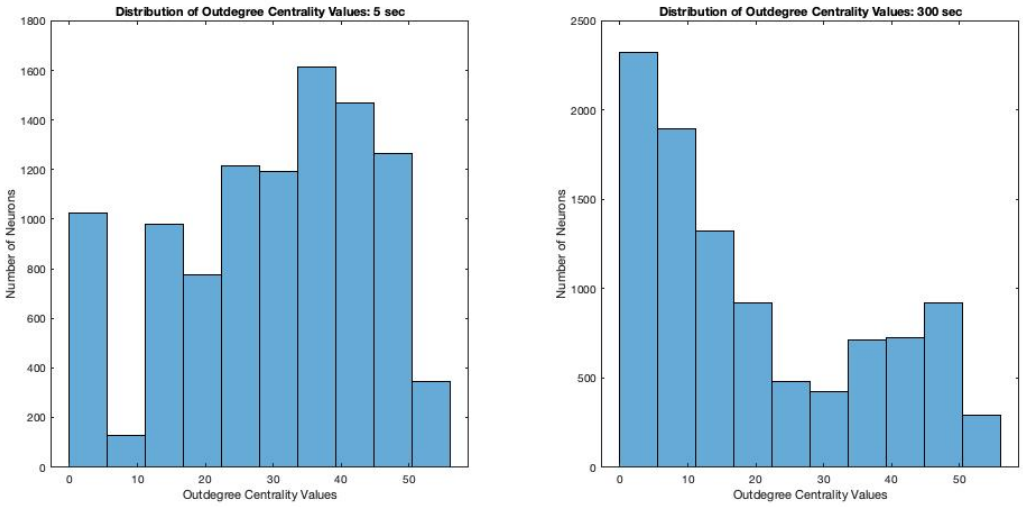


Figure 6.18: Outdegree centrality histograms for all neurons after 5s (left) and 300s (right) STDP.

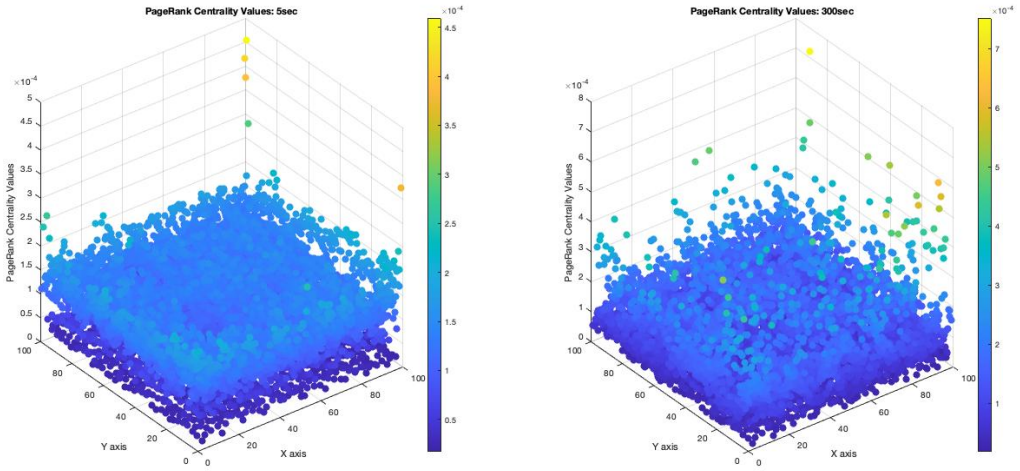


Figure 6.19: PageRank centrality for all neurons after 5s (left) and 300s (right) STDP. Axes, etc. same as figure 6.15. The number of neurons with high PageRank centrality decreases, and the distribution is not similar to degree centrality anymore because the graph is directed.

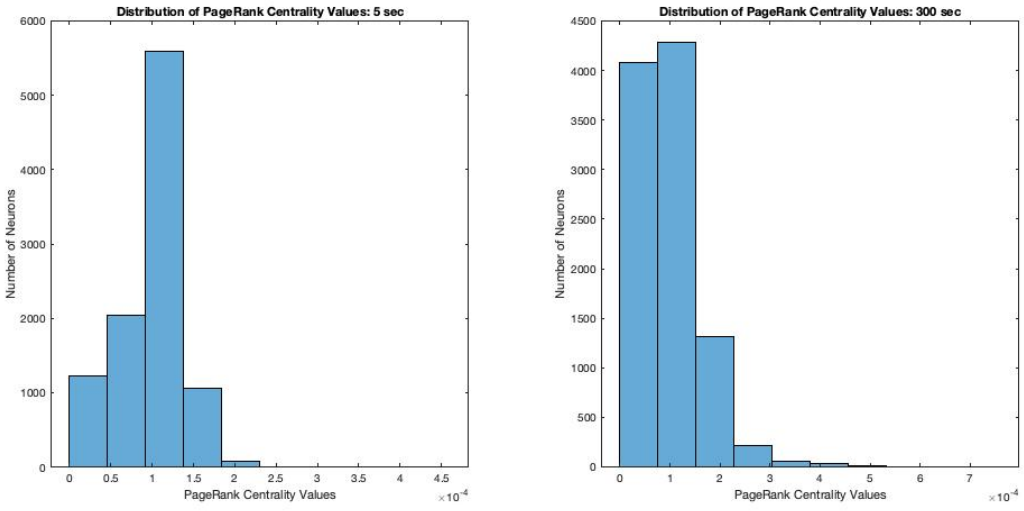


Figure 6.20: PageRank centrality histograms for all neurons after 5s (left) and 300s (right) STDP.

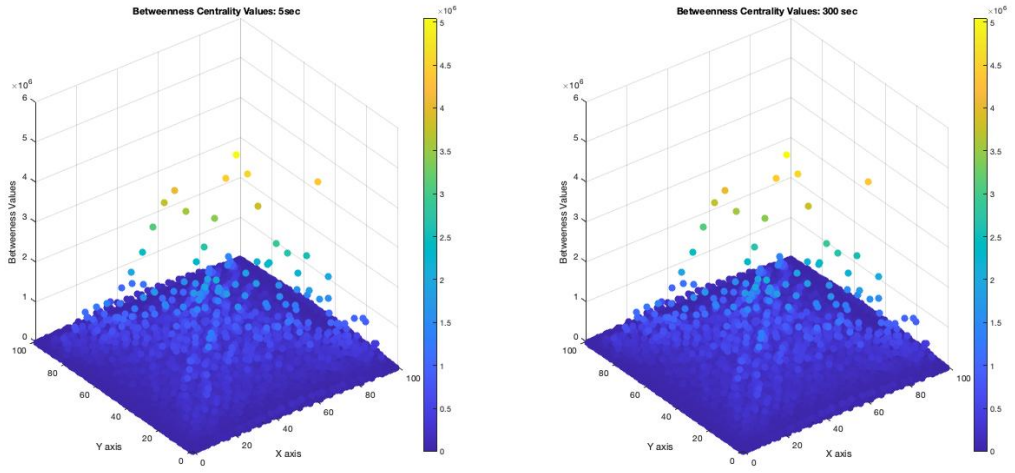


Figure 6.21: Betweenness centrality for all neurons after 5s (left) and 300s (right) STDP. Axes, etc. same as figure 6.15. The distribution of betweenness centrality remains similar to the growth model.

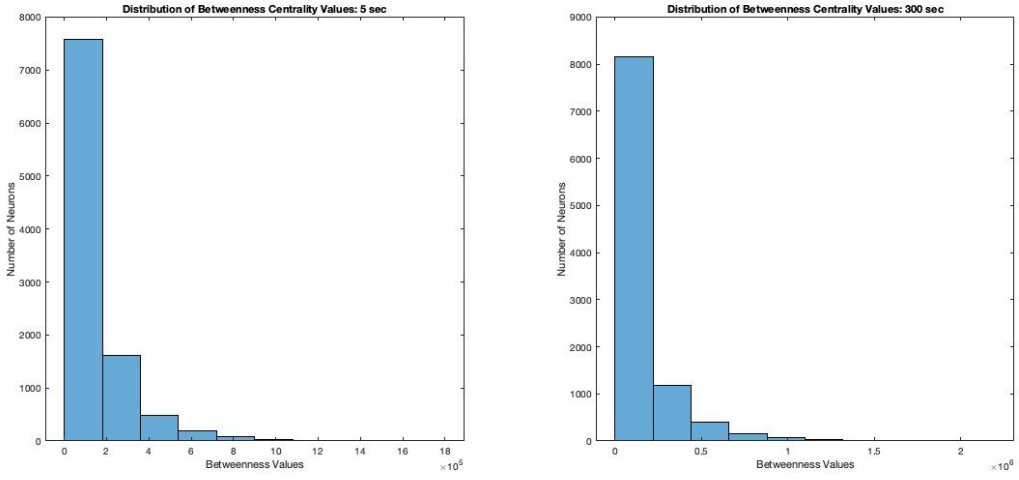


Figure 6.22: Betweenness centrality histograms for all neurons after 5s (left) and 300s (right) STDP.

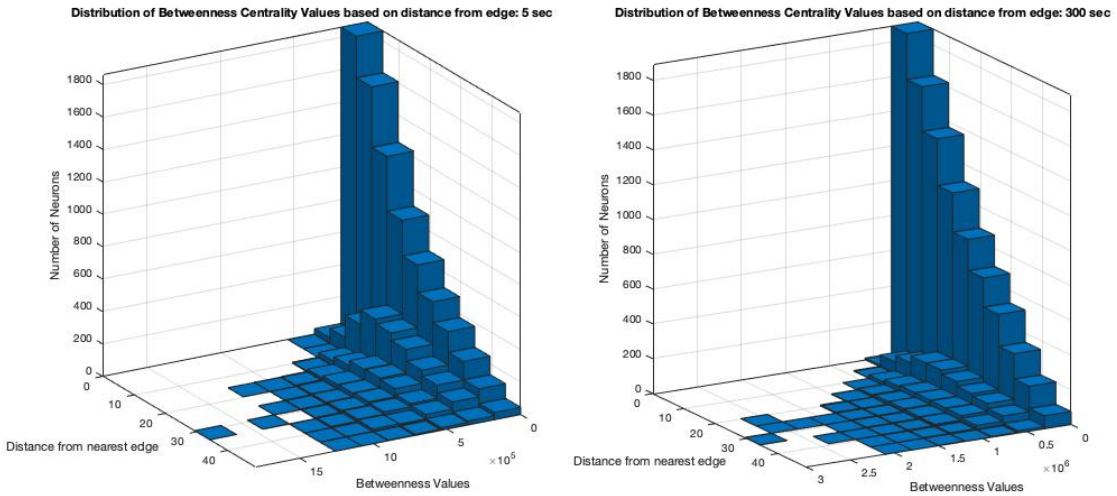


Figure 6.23: Betweenness centrality histogram as a function of distance from nearest network border for 5s (left) and 300s (right) STDP simulations.

Chapter 7

CONCLUSION AND FUTURE WORK

This thesis implemented different forms of STDP, simulated the refinement phase in neuronal development by running an STDP simulation after running a growth simulation, investigated the functional and structural properties of the brain graph generated before and after pruning the cortical culture growth using STDP, and identified hubs in the network using centrality measures. We validated the STDP results by matching results to the source literature of the STDP models. We were able to prune the network by removing weak connections using STDP, and this is proven by presenting the plots for network connections for both growth and STDP simulation. In graphical analysis, the graph formed using the cortical growth model [60] is an undirected weighted graph with uniform symmetric connections due to repetition in the network layout, and the graph formed after the STDP run is a directed weighted graph with asymmetric connections. The network for both growth and refinement is a strongly connected network which has significantly higher small-worldness index than an Erdős–Rényi random graph, indicating that our graph is more similar to an actual brain network than a random graph. In centrality analysis, visualization results show that while a large number of central neurons exist in the network after the growth simulation, the number decreases after STDP pruning. This indicates the development of brain network hubs in the early phases of refinement; these hubs could potentially play a key role in the network spiking activity.

This is the first analysis of STDP pruning on a simulated cortical growth network as far as the author is aware. As our network is able to simulate both growth and refinement in vitro, it will enable scientists to do study the progression of the network in extensive detail during the complete development process. Recognizing hubs enable us to further study their role in the functional connectivity of the network. Hubs have been found to be critical for cognitive functions and are more vulnerable to degenerative diseases like Alzeimer's disease [47]. Being able to identify these nodes and then studying the network behavior after selectively removing them will allow researchers to study the effect of damage to these neurons due to degeneration and brain injury.

7.1 Future Work

Future work concerns deeper analysis of the network and analyzing the factors affecting centrality values. Future work can include the following ideas.

First, we can investigate the spiking activity of the network during the STDP simulation to analyze the changes in network behavior and find out the role of hubs in network activity. We can check if the neurons with high centraliry values are burst originators or major burst leaders (MBLs) [48]. Identifying MBLs can help to better predict the origin of bursts and understand the spontaneous activity.

Second, we can expand the work by studying other network properties like network efficiency, weighted network cost etc., and doing a deeper analysis into the features responsible for high centrality values. We can investigate the reasons for stratification of degree centrality values in the growth simulation and the characteristics responsible for the occurrence of specific centrality ranges.

Third, the current implementation of STDP is limited to excitatory neurons. This can be extended to inhibitory neurons to expand the capability of our model to accommodate all types of cells.

Lastly, we can also try to identify structural motifs, which are recurrent patterns or subgraphs that occur in large number in our network [71, 77], and analyze their importance in information flow.

BIBLIOGRAPHY

[1] Fornito A, Zalesky A, and Breakspear M. *Graph analysis of the human connectome: promise, progress, and pitfalls*, October 2013. URL `https://pubmed.ncbi.nlm.nih.gov/23643999/`. ISSN: 1095-9572 Publisher: Neuroimage Volume: 80.

[2] Sophie Achard and Ed Bullmore. *Efficiency and cost of economical brain functional networks.* PLoS Comput Biol, 3(2):e17, February 2007. ISSN 1553-7358. doi: 10.1371/journal.pcbi.0030017.

[3] Sophie Achard, Raymond Salvador, Brandon Whitcher, John Suckling, and Ed Bullmore. *A Resilient, Low-Frequency, Small-World Human Brain Functional Network with Highly Connected Association Cortical Hubs.* J. Neurosci., 26(1):63–72, January 2006. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.3874-05.2006. URL `https://www.jneurosci.org/content/26/1/63`. Publisher: Society for Neuroscience Section: Behavioral/Systems/Cognitive.

[4] William Aiello, Fan Chung, and Linyuan Lu. *A random graph model for power law graphs.* Experiment. Math., 10(1):53–66, 2001. URL `https://projecteuclid.org:443/euclid.em/999188420`.

[5] Réka Albert and Albert-László Barabási. *Statistical mechanics of complex networks.* Reviews of Modern Physics, 74(1):47–97, 2002. ISSN 0034-6861. doi: 10.1103/revmodphys.74.47.

[6] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson. *Molecular Biology of the Cell.* Garland, 4th edition, 2002.

[7] Jeffrey Alstott, Michael Breakspear, Patric Hagmann, Leila Cammoun, and Olaf Sporns. *Modeling the Impact of Lesions in the Human Brain.* PLOS Computational Biology, 5(6):e1000408, 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000408. URL `https://dx.doi.org/10.1371/journal.pcbi.1000408`.

[8] Vajarala Ashikh, Gopikrishna Deshpande, Rangaprakash D., and D. Dutt. *Clustering of dynamic functional connectivity features obtained from functional Magnetic Resonance Imaging data*. pages 308–312, August 2015. doi: 10.1109/ICACCI.2015.7275626.

[9] Danielle Bassett and Edward Bullmore. *Small-World Brain Networks*. The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry, 12:512–23, January 2007. doi: 10.1177/1073858406293182.

[10] Danielle S. Bassett and Edward T. Bullmore. *Human Brain Networks in Health and Disease*. Current opinion in neurology, 22(4):340, August 2009. doi: 10.1097/WCO.0b013e32832d93dd. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2902726/`. Publisher: NIH Public Access.

[11] Danielle S. Bassett, Ankit N. Khambhati, and Scott T. Grafton. *Emerging Frontiers of Neuroengineering: A Network Science of Brain Connectivity*. Annual Review of Biomedical Engineering, 19(1):327–352, 2017. ISSN 1523-9829. doi: 10.1146/annurev-bioeng-071516-044511. URL `https://dx.doi.org/10.1146/annurev-bioeng-071516-044511`.

[12] Roger E. Beaty, Mathias Benedek, Scott Barry Kaufman, and Paul J. Silvia. *Default and Executive Network Coupling Supports Creative Idea Production*. Scientific Reports, 5(1):10964, June 2015. ISSN 2045-2322. doi: 10.1038/srep10964. URL `https://www.nature.com/articles/srep10964`. Number: 1 Publisher: Nature Publishing Group.

[13] John M. Beggs and Dietmar Plenz. *Neuronal Avalanches in Neocortical Circuits*. Journal of Neuroscience, 23(35):11167–11177, 2003. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.23-35-11167.2003. URL `https://www.jneurosci.org/content/23/35/11167`. Publisher: Society for Neuroscience _eprint: https://www.jneurosci.org/content/23/35/11167.full.pdf.

[14] Curtis C. Bell, Victor Z. Han, Yoshiko Sugawara, and Kirsty Grant. *Synaptic plasticity in a cerebellum-like structure depends on temporal order*. Nature, 387(6630):278–281, 1997. ISSN 0028-0836. doi: 10.1038/387278a0.

[15] Michael Beyeler, Kristofor Carlson, Ting-Shuo Chou, Nikil Dutt, and Jeff Krichmar. *CARLsim*

3: A User-Friendly and Highly Optimized Library for the Creation of Neurobiologically Detailed Spiking Neural Networks. 07 2015. doi: 10.1109/IJCNN.2015.7280424.

[16] G Bi and M Poo. *Synaptic Modification by Correlated Activity: Hebb's Postulate Revisited*. Ann Rev Neurosci, 24:139–66, January 2001. URL http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.neuro.24.1.139.

[17] Guo-Qiang Bi and Mu-Ming Poo. *Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type*. The Journal of Neuroscience, 18(24):10464–10472, 1998. ISSN 0270-6474. doi: 10.1523/jneurosci.18-24-10464.1998.

[18] Maja A.A. Binnewijzend, Sofie M. Adriaanse, Wiesje M. Van Der Flier, Charlotte E. Teunissen, Jan C. De Munck, Cornelis J. Stam, Philip Scheltens, Bart N.M. Van Berckel, Frederik Barkhof, Alle Meije Wink, and et al. *Brain network alterations in Alzheimer's disease measured by Eigenvector centrality in fMRI are related to cognition and CSF biomarkers*. Human Brain Mapping, 35(5):2383–2393, 2014. ISSN 1065-9471. doi: 10.1002/hbm.22335.

[19] John M. Bolland. *Sorting out centrality: An analysis of the performance of four centrality models in real and simulated networks*. Social Networks, 10(3):233–253, 1988. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(88)90014-7. URL https://www.sciencedirect.com/science/article/pii/0378873388900147.

[20] Sergey Brin and Lawrence Page. *The Anatomy of a Large-scale Hypertextual Web Search Engine*. Comput. Netw. ISDN Syst., 30(1-7):107–117, April 1998. ISSN 0169-7552. doi: 10.1016/S0169-7552(98)00110-X. URL http://dx.doi.org/10.1016/S0169-7552(98)00110-X.

[21] Ed Bullmore and Olaf Sporns. *Complex brain networks: graph theoretical analysis of structural and functional systems*. Nature Reviews Neuroscience, 10(3):186–198, 2009. ISSN 1471-003X. doi: 10.1038/nrn2575.

[22] Edward Bullmore and Danielle Bassett. *Brain Graphs: Graphical Models of the Human Brain Connectome*. Annual review of clinical psychology, 7:113–40, April 2010. doi: 10.1146/annurev-clinpsy-040510-143934.

[23] Anthony Burkitt, Hamish Meffin, and David Grayden. *Spike-Timing-Dependent Plasticity: The Relationship to Rate-Based Learning for Models with Weight Dynamics Determined by a Stable Fixed Point.* Neural computation, 16:885–940, 06 2004. doi: 10.1162/089976604773135041.

[24] B. L. Chen, D. H. Hall, and D. B. Chklovskii. *Wiring optimization can relate neuronal structure and function.* Proceedings of the National Academy of Sciences, 103(12):4723–4728, 2006. ISSN 0027-8424. doi: 10.1073/pnas.0506806103. URL https://dx.doi.org/10.1073/pnas.0506806103.

[25] Luis M. Colon-Perez, Michelle Couret, William Triplett, Catherine C. Price, and Thomas H. Mareci. *Small Worldness in Dense and Weighted Connectomes.* Frontiers in Physics, 4, 2016. ISSN 2296-424X. doi: 10.3389/fphy.2016.00014. URL https://dx.doi.org/10.3389/fphy.2016.00014.

[26] Rui Ponte Costa, Robert C Froemke, P Jesper Sjöström, and Mark Cw Van Rossum. *Unified pre- and postsynaptic long-term plasticity enables reliable and flexible learning.* eLife, 4, 2015. ISSN 2050-084X. doi: 10.7554/elife.09457.

[27] Daniel. *The Spike-Timing Dependence of Plasticity.* Neuron, 75(4): 556–571, 2012. ISSN 0896-6273. doi: 10.1016/j.neuron.2012.08.001. URL https://dx.doi.org/10.1016/j.neuron.2012.08.001.

[28] Easley David and Kleinberg Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, USA, 2010. ISBN 0521195330.

[29] Willem De Haan, Katherine Mott, Elisabeth C. W. Van Straaten, Philip Scheltens, and Cornelis J. Stam. *Activity Dependent Degeneration Explains Hub Vulnerability in Alzheimer's Disease.* PLOS Computational Biology, 8(8):e1002582, 2012. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002582. URL https://dx.doi.org/10.1371/journal.pcbi.1002582.

[30] P. Erdös and A. Rényi. *On Random Graphs I.* Publicationes Mathematicae Debrecen, 6:290, 1959.

[31] Paul Erdos and Alfred Renyi. *On the evolution of random graphs.* Publ. Math. Inst. Hungary. Acad. Sci., 5:17–61, 1960.

[32] Farzad V. Farahani, Waldemar Karwowski, and Nichole R. Lighthall. *Application of Graph Theory for Identifying Connectivity Patterns in Human Brain Networks: A Systematic Review.* Frontiers in Neuroscience, 13, 2019. ISSN 1662-453X. doi: 10.3389/fnins.2019.00585. URL https://dx.doi.org/10.3389/fnins.2019.00585.

[33] Alex Fornito. *Graph Theoretic Analysis of Human Brain Networks*, page 283–314. Neuromethods, 2016. doi: 10.1007/978-1-4939-5611-1-10.

[34] Linton C. Freeman. *A Set of Measures of Centrality Based on Betweenness.* Sociometry, 40(1):35–41, March 1977. ISSN 00380431. doi: 10.2307/3033543. URL http://dx.doi.org/10.2307/3033543.

[35] Robert C. Froemke and Yang Dan. *Spike-timing-dependent synaptic modification induced by natural spike trains.* Nature, 416(6879):433–438, 2002. ISSN 0028-0836. doi: 10.1038/416433a.

[36] Linda Geerligs, Remco J. Renken, Emi Saliasi, Natasha M. Maurits, and Monicque M. Lorist. *A Brain-Wide Study of Age-Related Changes in Functional Connectivity.* Cereb Cortex, 25(7): 1987–1999, July 2015. ISSN 1460-2199. doi: 10.1093/cercor/bhu012.

[37] E. N. Gilbert. *Random Graphs.* Ann. Math. Statist., 30(4):1141–1144, 12 1959. doi: 10.1214/aoms/1177706098. URL https://doi.org/10.1214/aoms/1177706098.

[38] M. Girvan and M. E. J. Newman. *Community structure in social and biological networks.* PNAS, 99(12):7821–7826, June 2002. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.122653799. URL https://www.pnas.org/content/99/12/7821. Publisher: National Academy of Sciences Section: Physical Sciences.

[39] Jennifer Golbeck. *Analyzing the Social Web.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2013. ISBN 0124055311.

[40] G. Gong, P. Rosa-Neto, F. Carbonell, Z. J. Chen, Y. He, and A. C. Evans. *Age- and Gender-Related Differences in the Cortical Anatomical Network.* The Journal of Neuroscience, 29(50): 15684–15693, 2009. ISSN 0270-6474. doi: 10.1523/jneurosci.2308-09.2009.

[41] W. Shane Grant and Randolph Voorhies. *cereal Docs - Quick Start*, 2017. URL https://uscilab.github.io/cereal/quickstart.html.

[42] Önder Gürcan. *Exploration of Biological Neural Wiring Using Self-Organizing Agents*. PhD thesis, 09 2013.

[43] R. Gütig, R. Aharonov, S. Rotter, and Haim Sompolinsky. *Learning Input Correlations through Nonlinear Temporally Asymmetric Hebbian Plasticity*. The Journal of Neuroscience, 23(9): 3697–3714, 2003. ISSN 0270-6474. doi: 10.1523/jneurosci.23-09-03697.2003.

[44] R. Gütig, R. Aharonov, S. Rotter, and Haim Sompolinsky. *Learning Input Correlations through Nonlinear Temporally Asymmetric Hebbian Plasticity*. J. Neurosci., 23(9):3697–3714, May 2003. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.23-09-03697.2003. URL `https://www.jneurosci.org/content/23/9/3697`. Publisher: Society for Neuroscience Section: ARTICLE.

[45] Julie S. Haas, Thomas Nowotny, and H. D. I. Abarbanel. *Spike-timing-dependent plasticity of inhibitory synapses in the entorhinal cortex*. J Neurophysiol, 96(6):3305–3313, December 2006. ISSN 0022-3077. doi: 10.1152/jn.00551.2006.

[46] Julie S. Haas, Thomas Nowotny, and H.D.I. Abarbanel. *Spike-Timing-Dependent Plasticity of Inhibitory Synapses in the Entorhinal Cortex*. Journal of Neurophysiology, 96(6):3305–3313, 2006. ISSN 0022-3077. doi: 10.1152/jn.00551.2006.

[47] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. *Mapping the Structural Core of Human Cerebral Cortex*. PLOS Biology, 6(7):e159, 2008. ISSN 1545-7885. doi: 10.1371/journal.pbio.0060159. URL `https://dx.doi.org/10.1371/journal.pbio.0060159`.

[48] Michael I. Ham, Luis M. Bettencourt, Floyd D. Mcdaniel, and Guenter W. Gross. *Spontaneous coordinated activity in cultured networks: Analysis of multiple ignition sites, primary circuits, and burst phase delay distributions*. Journal of Computational Neuroscience, 24(3):346–357, 2008. ISSN 0929-5313. doi: 10.1007/s10827-007-0059-1. URL `https://dx.doi.org/10.1007/s10827-007-0059-1`.

[49] Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, June 1949. ISBN 0-8058-4300-0.

[50] Yi-Hsin Emily Hsu. *Extending a Neural Simulator to Combine Growth and Spike-Timing-Dependent Plasticity.* Master's thesis, University of Washngton, Bothell, 2020.

[51] Peter R. Huttenlocher, C. D. Courten, L. Garey, and H. V. D. Loos. *Synaptogenesis in human visual cortex — evidence for synapse elimination during normal development.* Neuroscience Letters, 33:247–252, 1982.

[52] PR Huttenlocher. *Synaptic density in human frontal cortex - developmental changes and effects of aging.* Brain research, 163(2):195—205, March 1979. ISSN 0006-8993. doi: 10.1016/0006-8993(79)90349-4. URL `https://doi.org/10.1016/0006-8993(79)90349-4`.

[53] Javier Iglesias, Jan Eriksson, François Grize, Marco Tomassini, and Alessandro Villa. *Dynamics of pruning in simulated large-scale spiking neural networks.* Bio Systems, 79:11–20, 03 2005. doi: 10.1016/j.biosystems.2004.09.016.

[54] Eugene Izhikevich and Niraj Desai. *Relating stdp to bcm.* Neural computation, 15:1511–23, 08 2003. doi: 10.1162/089976603321891783.

[55] Eugene M. Izhikevich. *Polychronization: computation with spikes.* Neural Comput, 18(2): 245–282, February 2006. ISSN 0899-7667. doi: 10.1162/089976606775093882.

[56] Eugene M. Izhikevich, Joseph A. Gally, and Gerald M. Edelman. *Spike-timing Dynamics of Neuronal Groups.* Cerebral Cortex, 14(8):933–944, August 2004. doi: 10.1093/cercor/bhh053. URL `http://dx.doi.org/10.1093/cercor/bhh053`.

[57] Karen E. Joyce, Paul J. Laurienti, Jonathan H. Burdette, and Satoru Hayasaka. *A New Measure of Centrality for Brain Networks.* PLOS ONE, 5(8):e12200, 2010. ISSN 1932-6203. doi: 10.1371/journal.pone.0012200. URL `https://dx.doi.org/10.1371/journal.pone.0012200`.

[58] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, editors. *Principles of Neural Science.* Elsevier, New York, third edition, 1991.

[59] Fumitaka Kawasaki. *Accelerating large-scale simulations of cortical neuronal network development.* Master's thesis, University of Washngton, Bothell, 2012.

[60] Fumitaka Kawasaki and Michael Stiber. *A simple model of cortical culture growth: burst property dependence on network composition and activity*. Biol Cybern, 108(4):423–443, August 2014. ISSN 1432-0770. doi: 10.1007/s00422-014-0611-9.

[61] Richard Kempter, Wulfram Gerstner, and J. Leo van Hemmen. *Hebbian learning and spiking neurons*. Phys. Rev. E, 59(4):4498–4514, April 1999. doi: 10.1103/PhysRevE.59.4498. URL https://link.aps.org/doi/10.1103/PhysRevE.59.4498. Publisher: American Physical Society.

[62] Jewel YunHsuan Lee, Michael Stiber, and Dong Si. *Machine Learning of Spatiotemporal Bursting Behavior in Developing Neural Networks*. Annu Int Conf IEEE Eng Med Biol Soc, 2018: 348–351, July 2018. ISSN 2694-0604. doi: 10.1109/EMBC.2018.8512358.

[63] R. Duncan Luce and Albert D. Perry. *A method of matrix analysis of group structure*. Psychometrika, 14(2):95–116, June 1949. ISSN 1860-0980. doi: 10.1007/BF02289146. URL https://doi.org/10.1007/BF02289146.

[64] M.-E. Lynall, D. S. Bassett, R. Kerwin, P. J. Mckenna, M. Kitzbichler, U. Muller, and E. Bullmore. *Functional Connectivity and Brain Networks in Schizophrenia*. The Journal of Neuroscience, 30(28):9477–9487, 2010. ISSN 0270-6474. doi: 10.1523/jneurosci.0333-10.2010.

[65] E. Maeda, H. P. Robinson, and A. Kawana. *The mechanisms of generation and propagation of synchronized bursting in developing networks of cortical neurons*. J. Neurosci., 15(10): 6834–6845, October 1995. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.15-10-06834.1995. URL https://www.jneurosci.org/content/15/10/6834. Publisher: Society for Neuroscience Section: Articles.

[66] P Mandira. *What is the function of the synaptic terminal (part of the neuron)? | Socratic*, 2017. URL https://socratic.org.

[67] H. Markram, Y. Wang, and M. Tsodyks. *Differential signaling via the same axon of neocortical pyramidal neurons*. Proceedings of the National Academy of Sciences, 95(9):5323–5328, 1998. ISSN 0027-8424. doi: 10.1073/pnas.95.9.5323.

[68] MATLABCentrality. *Measure node importance - MATLAB centrality*, 2017. URL https://www.mathworks.com/help/matlab/ref/graph.centrality.html.

[69] Adam McLaughlin and David Bader. *Scalable and High Performance Betweenness Centrality on the GPU.* International Conference for High Performance Computing, Networking, Storage and Analysis, SC, 2015:572–583, 01 2015. doi: 10.1109/SC.2014.52.

[70] Sifis Micheloyannis, Michael Vourkas, Vassiliki Tsirka, Eleni Karakonstantaki, Kassia Kanatsouli, and Cornelis J. Stam. *The influence of ageing on complex brain networks: a graph theoretical analysis.* Hum Brain Mapp, 30(1):200–208, January 2009. ISSN 1097-0193. doi: 10.1002/hbm.20492.

[71] R. Milo. *Superfamilies of Evolved and Designed Networks.* Science, 303(5663):1538–1542, 2004. ISSN 0036-8075. doi: 10.1126/science.1089167.

[72] M. E. J. Newman. *Networks: an introduction.* Oxford University Press, Oxford; New York, 2010. ISBN 9780199206650 0199206651.

[73] Stuart Oldham and Alex Fornito. *The development of brain network hubs.* Developmental Cognitive Neuroscience, 36:100607, 2019. ISSN 1878-9293. doi: 10.1016/j.dcn.2018.12.005. URL `https://dx.doi.org/10.1016/j.dcn.2018.12.005`.

[74] Micah Richert, Jayram Moorkanikara Nageswaran, Nikil Dutt, and Jeffrey L. Krichmar. *An Efficient Simulation Environment for Modeling Large-Scale Cortical Processing.* Front. Neuroinform., 5, 2011. ISSN 1662-5196. doi: 10.3389/fninf.2011.00019. URL `https://www.frontiersin.org/articles/10.3389/fninf.2011.00019/full`. Publisher: Frontiers.

[75] J. Rubenstein and P. Rakic. *Neural Circuit Development and Function in the Healthy and Diseased Brain: Comprehensive Developmental Neuroscience.* Comprehensive developmental neuroscience. Elsevier Science, 2013. ISBN 978-0-12-397346-7. URL `https://books.google.com/books?id=0yfrY7PIsg4C`.

[76] R. Salvador, J. Suckling, M. Coleman, J. Pickard, D. Menon, and E. Bullmore. *Neurophysiological architecture of functional magnetic resonance images of human brain.* Cerebral cortex, 15 9:1332–42, 2005.

[77] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. *Network motifs in the transcriptional regulation network of Escherichia coli.* Nature Genetics, 31(1):64–68, 2002. ISSN 1061-4036. doi: 10.1038/ng881. URL https://dx.doi.org/10.1038/ng881.

[78] Snigdha Singh and Michael Stiber. *BrainGrid Simulation Results: Growth and STDP*, March 2021. URL https://doi.org/10.5281/zenodo.4626403.

[79] Sen Song and L.F. Abbott. *Cortical Development and Remapping through Spike Timing-Dependent Plasticity.* Neuron, 32(2):339–350, 2001. ISSN 0896-6273. doi: 10.1016/s0896-6273(01)00451-2.

[80] Sen Song, Kenneth D. Miller, and L. F. Abbott. *Competitive Hebbian learning through spike-timing-dependent synaptic plasticity.* Nature Neuroscience, 3(9):919–926, September 2000. ISSN 1546-1726. doi: 10.1038/78829. URL https://www.nature.com/articles/nn0900919. Number: 9 Publisher: Nature Publishing Group.

[81] M. Stiber, F. Kawasaki, Delmar B. Davis, Hazeline U. Asuncion, J. Lee, and D. Boyer. *BrainGrid+Workbench: High-performance/high-quality neural simulation.* 2017 International Joint Conference on Neural Networks (IJCNN), pages 2469–2476, 2017.

[82] Yu Sun, Julian Lim, Zhongxiang Dai, KianFoong Wong, Fumihiko Taya, Yu Chen, Junhua Li, Nitish Thakor, and Anastasios Bezerianos. *The effects of a mid-task break on the brain connectome in healthy participants: A resting-state functional MRI study.* NeuroImage, 152: 19 – 30, 2017. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2017.02.084. URL http://www.sciencedirect.com/science/article/pii/S1053811917301969.

[83] Elin Teppa, Diego Javier Zea, and Cristina Marino Buslje. *Identification of Coevolving Amino Acids using Mutual Information.* 10 2012. ISBN 978-14775549-1-3.

[84] Tatyana S. Turova and Alessandro E.P. Villa. *On a phase diagram for random neural networks with embedded spike timing dependent plasticity.* BioSystems, 89(1-3):280–6, 2007. doi: 10.1016/j.biosystems.2006.05.019. URL https://www.hal.inserm.fr/inserm-00383808.

[85] Thomas Valente, Kathryn Coronges, Cynthia Lakon, and Elizabeth Costenbader. *How Correlated Are Network Centrality Measures?* Connections (Toronto, Ont.), 28:16–26, 01 2008.

[86] Mark van Rossum, Guoqiang Bi, and G Turrigiano. *Stable Hebbian Learning from Spike Timing-Dependent Plasticity.* The Journal of neuroscience : the official journal of the Society for Neuroscience, 20:8812–21, 01 2001. doi: 10.1523/JNEUROSCI.20-23-08812.2000.

[87] Daniel A. Wagenaar, Jerome Pine, and Steve M. Potter. *An extremely rich repertoire of bursting patterns during the development of cortical cultures.* BMC Neurosci, 7:11, February 2006. ISSN 1471-2202. doi: 10.1186/1471-2202-7-11.

[88] Jin-Hui Wang, Xi-Nian Zuo, Suril Gohel, Michael P. Milham, Bharat B. Biswal, and Yong He. *Graph Theoretical Analysis of Functional Brain Networks: Test-Retest Evaluation on Short- and Long-Term Resting-State Functional MRI Data.* PLOS ONE, 6(7):e21976, 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0021976. URL `https://dx.doi.org/10.1371/journal.pone.0021976`.

[89] Jinhui Wang. *Graph-based network analysis of resting-state functional MRI.* Frontiers in Systems Neuroscience, 2010. ISSN 1662-5137. doi: 10.3389/fnsys.2010.00016. URL `https://dx.doi.org/10.3389/fnsys.2010.00016`.

[90] Jinhui Wang, Liang Wang, Yufeng Zang, Hong Yang, Hehan Tang, Qiyong Gong, Zhang Chen, Chaozhe Zhu, and Yong He. *Parcellation-dependent small-world brain functional networks: A resting-state fMRI study.* Human Brain Mapping, 30(5):1511–1523, 2009. ISSN 1065-9471. doi: 10.1002/hbm.20623.

[91] Yu Wang, Lei Jiang, Xiao-Yu Wang, Weizhe Chen, Yi Shao, Qin-Kai Chen, and Jin-Lei Lv. *Evidence of altered brain network centrality in patients with diabetic nephropathy and retinopathy: an fMRI study using a voxel-wise degree centrality approach.* Therapeutic Advances in Endocrinology and Metabolism, 10:204201881986572, 2019. ISSN 2042-0188. doi: 10.1177/2042018819865723. URL `https://dx.doi.org/10.1177/2042018819865723`.

[92] Duncan J. Watts and Steven H. Strogatz. *Collective dynamics of 'small-world' networks.* Nature, 393(6684):440–442, June 1998. ISSN 0028-0836. doi: 10.1038/30918. URL `http://dx.doi.org/10.1038/30918`.

[93] Thomas Welton, Cris S. Constantinescu, Dorothee P. Auer, and Rob A. Dineen. *Graph Theoretic Analysis of Brain Connectomics in Multiple Sclerosis: Reliability and Relation-*

ship with Cognition. Brain Connectivity, 10(2):95–104, 2020. ISSN 2158-0014. doi: 10.1089/brain.2019.0717.

[94] S. Yu, D. Huang, W. Singer, and D. Nikolic. *A Small World of Neuronal Synchrony*. Cerebral Cortex, 18(12):2891–2901, 2008. ISSN 1047-3211. doi: 10.1093/cercor/bhn047. URL `https://dx.doi.org/10.1093/cercor/bhn047`.

[95] Andrew Zalesky, Alex Fornito, Ian H. Harding, Luca Cocchi, Murat Yücel, Christos Pantelis, and Edward T. Bullmore. *Whole-brain anatomical networks: does the choice of nodes matter?* Neuroimage, 50(3):970–983, April 2010. ISSN 1095-9572. doi: 10.1016/j.neuroimage.2009.12.027.

[96] Xi-Nian Zuo, Ross Ehmke, Maarten Mennes, Davide Imperati, F. Xavier Castellanos, Olaf Sporns, and Michael P. Milham. *Network Centrality in the Human Functional Connectome*. Cerebral Cortex, 22(8):1862–1875, 2012. ISSN 1460-2199. doi: 10.1093/cercor/bhr269. URL `https://dx.doi.org/10.1093/cercor/bhr269`.