3
Materials and Methods

3.1 *C. elegans* NEURONAL NETWORK

Caenorhabditis elegans is a transparent, soil dwelling nematode which is found commonly in many parts of the world. This free-living worm has a life span of around 17 days and can be either male or hermaphrodite by nature. The hermaphrodite animal consists 118 groups of 302 neurons [Hall and Russell, 1991; White et al., 1986]. The general structure of the nervous system of *C. elegans* is made up of two units. The centrally located pharynx contains a nerve ring of twenty cells representing the first unit. The other unit comprises of the rest of the neuropil with a ventral and dorsal nerve cord that holds nine nervous system ganglia.

We constructed the *C. elegans* Neuronal Network (CeNN), a graph theoretical model of its nervous system, using the data obtained from WormAtlas (www.wormatlas.org) [Beth L Chen et al., 2006]. This wiring diagram consists of 277 non-pharyngeal neurons, covering 6393 chemical synapses and 890 electrical junctions [Choe et al., 2004]. In this network multiple synapses between neurons were merged and neuromuscular junctions were excluded to create a binary (unweighted) directional graph containing 2105 synaptic connections. After including 20 pharyngeal neurons, the connectome holds 2345 binary connections among 297 neurons. In CeNN, neurons represent nodes and synaptic connections represent links (Figure 3.1). We used the CeNN data (297 nodes and 2345 edges) including the pharyngeal neurons for phenotypic and genotypic characterization of control mechanisms (Chapter 4). For rest of the studies involving *C. elegans* (Chapter 5, Chapter 6 and Chapter 7), we used data (277 nodes and 2105 edges) excluding the neurons in pharynx (Annexure A).

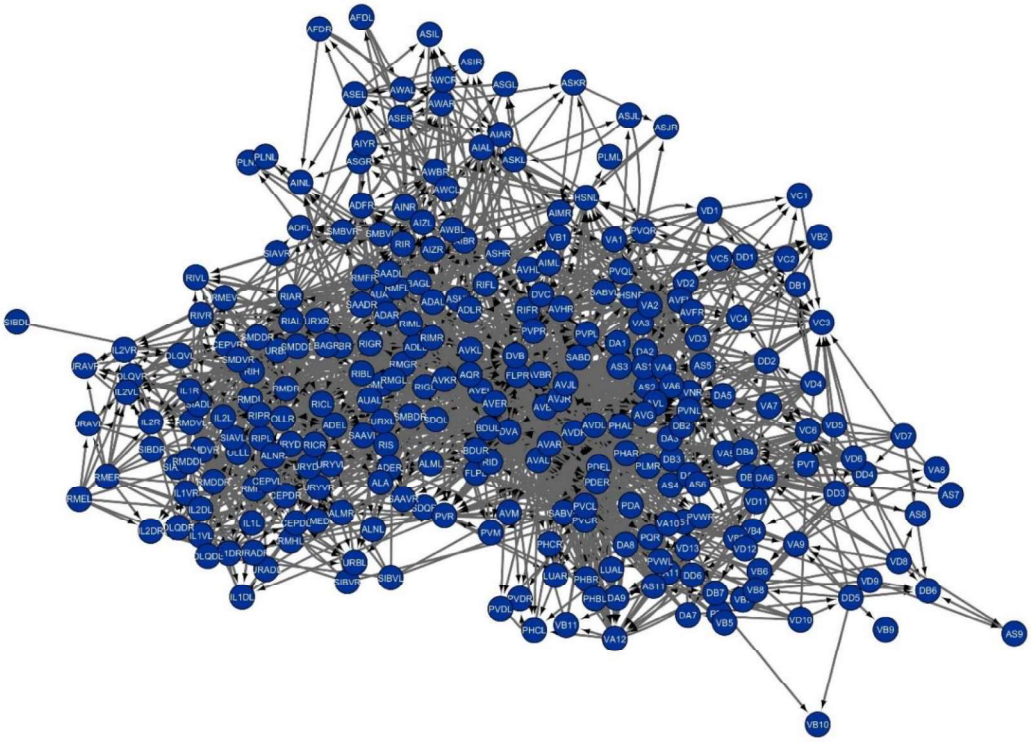

Figure 3.1 : *C. elegans* neuronal network: A network representation of *C. elegans* nervous system.

3.2 RANDOM CONTROLS OF CeNN

We constructed two types of random controls of CeNN viz. Erdős–Rényi random control (ER) and degree distribution conserved control (DD) [Erdös, P., 1984; Maslov and Sneppen, 2002]. Erdős–Rényi random control for a graph G was created by connecting randomly chosen pair of vertices n_i and n_j in the network with probability p. This strategy maintains the number of nodes, number edges as well as average degree $< k >$ of CeNN, while randomizing the wiring pattern. Due to probabilistic nature of the model, the ER control is created as an ensemble of graphs over which statistics were computed.

Degree distribution conserved control is used to assess dependence of a topological feature on the connectivity/degree of nodes. In DD control, the objective is to preserve the in degree and out degree of each node as that of the CeNN. To construct a degree preserved control of graph $G(V, E)$ with V vertex and E edges, we used the strategy suggested by Maslov and Sneppen [Maslov and Sneppen, 2002]. According to this strategy, we randomly choose two edges (i, j) and (k, l) such that $i \neq j \neq k \neq l$. The source and target nodes of these edges are then swapped such that the newly formed edges do not pre-exist in the network, and such that the network is not fragmented (Figure 3.2). Repeating this step will create the desired randomized network by preserving the degree sequence. Thus, in this control in-degree and out-degree of each node was also preserved in addition to number of nodes and edges in the network.

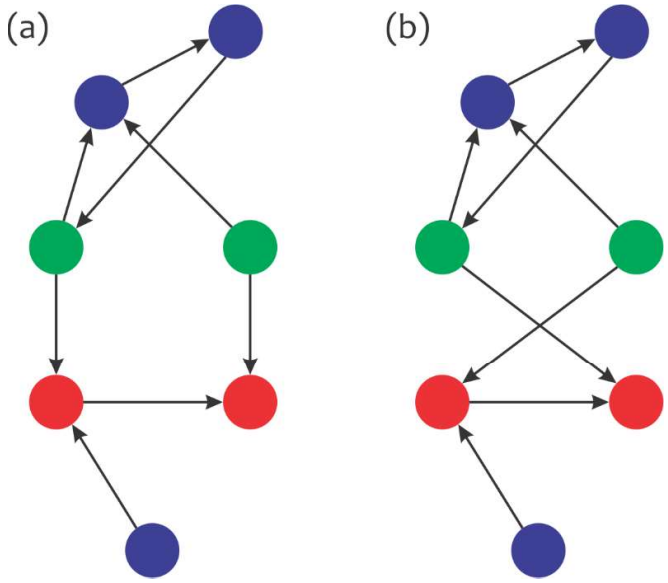

Figure 3.2 : Strategy to construct degree distribution preserved control. The source nodes are shown in green, and the target nodes are shown in red. Other nodes of the network are shown in blue.

3.3 TOPOLOGICAL PROPERTIES OF CeNN

In our investigations described in this thesis, we calculated following graph theoretical properties of the network embodying clustering, compactness, statistics of structural motifs and controllability of the network [Dorogovtsev, 2014; Y.-Y. Liu et al., 2011; R Milo et al., 2002].

3.3.1 Degree

Degree of a node k_i is defined as the number of nodes to which a given node i is connected. In a directed network degree can be divided into in-degree k_{in} and out-degree k_{out}.

3.3.2 Clustering coefficient

Clustering coefficient of a node C_i is defined as the ratio of number of triangles (triangle refers to a three node clique) made by a node with its neighbors to the maximum number of triangles that can be formed by it [Watts and Strogatz, 1998]. For a graph $G = (V, E)$ the clustering coefficient of a node i is defined as Eq.(3.1).

$$C_i = \frac{|\{e_{kj} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)} \tag{3.1}$$

Here, N_i refers to the neighborhood of node i, and k_i represents its connectivity (degree). The average clustering coefficient $\overline{(C)}$ was calculated by averaging clustering coefficients of all n nodes (Eq.(3.2)).

$$\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{3.2}$$

3.3.3 Characteristic path-length

Characteristic path-length (L) enumerates compactness, reflecting ease of information transfer, of the network. It is defined as the average of shortest path-lengths among all pairs of nodes in the network (Eq.(3.3)).

$$L = \frac{1}{n(n-1)} . \sum_{i \neq j} d(v_i, v_j) \tag{3.3}$$

3.3.4 Analysis of motifs

Network sub-structures that are significantly over-represented in networks compared to their random counterparts are known as motifs [R Milo et al., 2002]. Some of these motifs are known to be of functional significance [Azulay, Itskovits, and Zaslaver, 2016; Mangan and Alon, 2003; R Milo et al., 2002]. A directed binary graph can have 13 types of three node sub-structures. These three node sub-graphs could further be divided into angular and triangular motifs. Angular motifs are linear three node sub-structures, whereas triangular motifs comprise of three nodes sub-graphs with either unidirectional or bidirectional edges (Figure 3.3). We computed the number of substructures and over-representation of these motifs (using $Z\text{-}score = \frac{X-\mu}{\sigma}$) following the methodology of Milo *et al.* [R Milo et al., 2002].

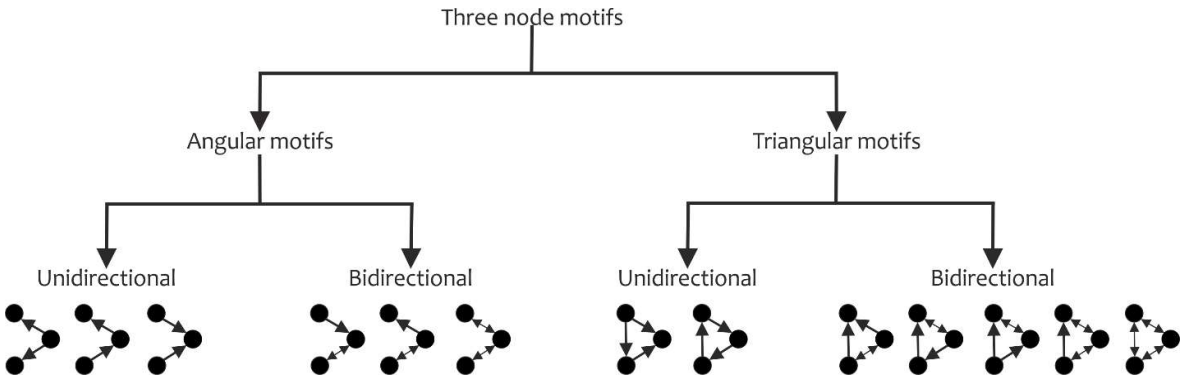

Figure 3.3 : Classification of three node sub-graphs.

Feed forward motifs

Network motifs are defined as patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized networks [R Milo et al., 2002].

They represent patterns of local connectivity among nodes that are present in numbers significantly higher than expected by chance. For our studies, we computed number of feed forward motifs, n_{FFM}, (among unidirectional triangular motifs) that are prevalent in many real world networks including CeNN [Alon, 2007a; R Milo et al., 2002]. We used the 'exhaustive recursive search' employed by Milo et al. for identification and enumeration of frequency of occurrence motifs [R Milo et al., 2002]. The $Zscore$ indicating significance of observed number of FFMs in CeNN was calculated by comparing it with its random controls (Eq.(3.4)).

$$Zscore = \frac{n_{FFM}(CeNN) - \overline{n_{FFM}(ER)}}{\sigma_{ER}} \tag{3.4}$$

Exhaustive recursive search for feed forward motifs

Step 1: Initiate a counter M to represent 'number of motifs' to zero.
Step 2: Starting from node i (= 1: N), find the outgoing edges in column a.
Step 3: From column a_j (= 1:length of column), find its outing edges in column b.
Step 4: If any outgoing edge of column b has outgoing edge = i, increase the counter M by 1.
Step 5: Repeat from step 2 until $i = N$.

3.4 STRUCTURAL CONTROLLABILITY

Most real systems are driven by nonlinear processes, but the controllability of nonlinear systems is in many aspects structurally similar to that of linear systems [Slotine and Li, 1991]. A system is said to be controllable if it can be driven from any initial state to any desired final state in finite time. This is possible if controllability matrix have full rank in accordance with Kalman's controllability rank condition [Kalman, 1963; D. Luenberger, 1979]. Finding full rank of the network for structural controllability is a computationally expensive task requiring a brute force search. A bypass method of maximum matching can perform the task efficiently by finding unmatched nodes that are known as driver nodes [Y.-Y. Liu et al., 2011] with an algorithmic complexity of $O(\sqrt{N} \times E)$, where N and E denote the number of nodes and edges respectively. Matched nodes are the ones which share the link in a maximum matching, else they are unmatched. These unmatched nodes are of importance because these have directed paths to matched nodes allowing full control of the network. The driver nodes are known to avoid hubs, which are essential for maintaining network integrity [Y.-Y. Liu et al., 2011]. Different algorithms with varying complexity have been proposed for finding maximum matching in a network [Dinic, 1970; Hopcroft and Karp, 1973; Micali and Vazirani, 1980; Mucha and Sankowski, 2004; Zhang, Lv, Yang, and Zhang, 2014]. Among them Hopcroft-Karp is versatile and has least complexity [Hopcroft and Karp, 1973]. For nervous system, we identified the critical neurons that fall into the category of unmatched nodes using maximum matching criterion.

3.4.1 Identification of driver neurons

Neurons that are critical for controlling the dynamical state of the network when provided with an external input are termed as 'Driver Neurons (D_n)'. We implemented the Hopcroft-Karp algorithm as well as augment matching algorithm for identification of maximum matching nodes to arrive at the exact set of driver neurons [Hopcroft and Karp, 1973; Pothen and Fan, 1990]. These algorithms compute the matching M in a graph $G = (V, E)$, having V vertices and E edges, such that vertex V_i has at most one incoming edge E_j. Matching is maximum if no other permutation of matching exists $M' \supset M$ [Duff et al., 2011]. Thus it provides a vector $p(j) = i$ if column j is matched to row i, or zero if column j is unmatched. The nodes/neurons corresponding to these unmatched columns are then referred to as the driver neurons. The complexity of Hopcroft-Karp algorithm is $O(\sqrt{N} \times E)$ and has been successfully implemented by Lui et al. [Y.-Y. Liu et al., 2011]. As opposed to Hopcroft-Karp algorithm in which the matching is ambiguous and dependent on the initiation point, the augment matching algorithm is unique. It allows identification of driver nodes without any uncertainty.

Analogous to driver nodes, we refer to critical neurons in CeNN as 'Driver Neurons'. Due to their role in control of network, driver neurons are of functional relevance to the neuronal network [Badhwar and Bagler, 2015]. We implemented the Hopcroft-Karp algorithm for computing minimum number of driver neurons for our studies described in Chapter 4. For studies performed as part of Chapter 5 and Chapter 6, the unique unmatched nodes were obtained by augment matching.

3.4.2 Number of driver neurons

In a network where every node can be in one of the multiple states, it has been shown that the state of the network can be controlled with the help of driver nodes [Badhwar and Bagler, 2015; Y.-Y. Liu et al., 2011; E. Tang and Bassett, 2017]. Aligned with this notion, driver neurons are those neurons which when controlled with an external input can provide full control over the state of the network [Y.-Y. Liu et al., 2011]. Due to their role in control of network 'number of driver neurons' (D_n) are of functional relevance to the neuronal network [Badhwar and Bagler, 2015]. We computed the minimum number of driver neurons using maximum matching criterion.

3.4.3 Hopcroft-Karp Algorithm for maximum matching

The strategy for obtaining maximum matching of a graph as described by Liu et al. is as follows [Y.-Y. Liu et al., 2011]. Figure 3.4 graphically depicts this strategy.

1. Pre-requisite: The graph should be sparse such that bipartitioning can be done of graph G such as U and V are two set of vertices from graph G with equal number of nodes.
2. Initiate an empty matching M column vector. Matching in a directed network is defined as a pair of edges not starting or ending in common vertex.
3. Use a breadth-first search (BFS) to find augmenting paths. It partitions the vertices of the graph into layers of matching and non-matching edges. For the search, start with the free nodes in U. This forms the first layer of the partitioning. The search finishes at the first layer where one or more free nodes in V are reached.
4. The free nodes in V are added to a set called F. This means that any node added to F will be the ending node of an augmenting path $-$ and a shortest augmenting path at that since the BFS finds shortest paths.
5. Once an augmenting path is found, a depth-first search (DFS) is used to add augmenting paths to current matching M. At any given layer, the DFS will follow edges that lead to an unused node from the previous layer. Paths in the DFS tree must be alternating paths (switching between matched and unmatched edges). Once the algorithm finds an augmenting path that uses a node from F, the DFS moves on to the next starting vertex.
6. The algorithm terminates when the algorithm can find no more augmenting paths in the breadth-first search step.
7. The time complexity of this algorithm is $O\sqrt{N}E$.

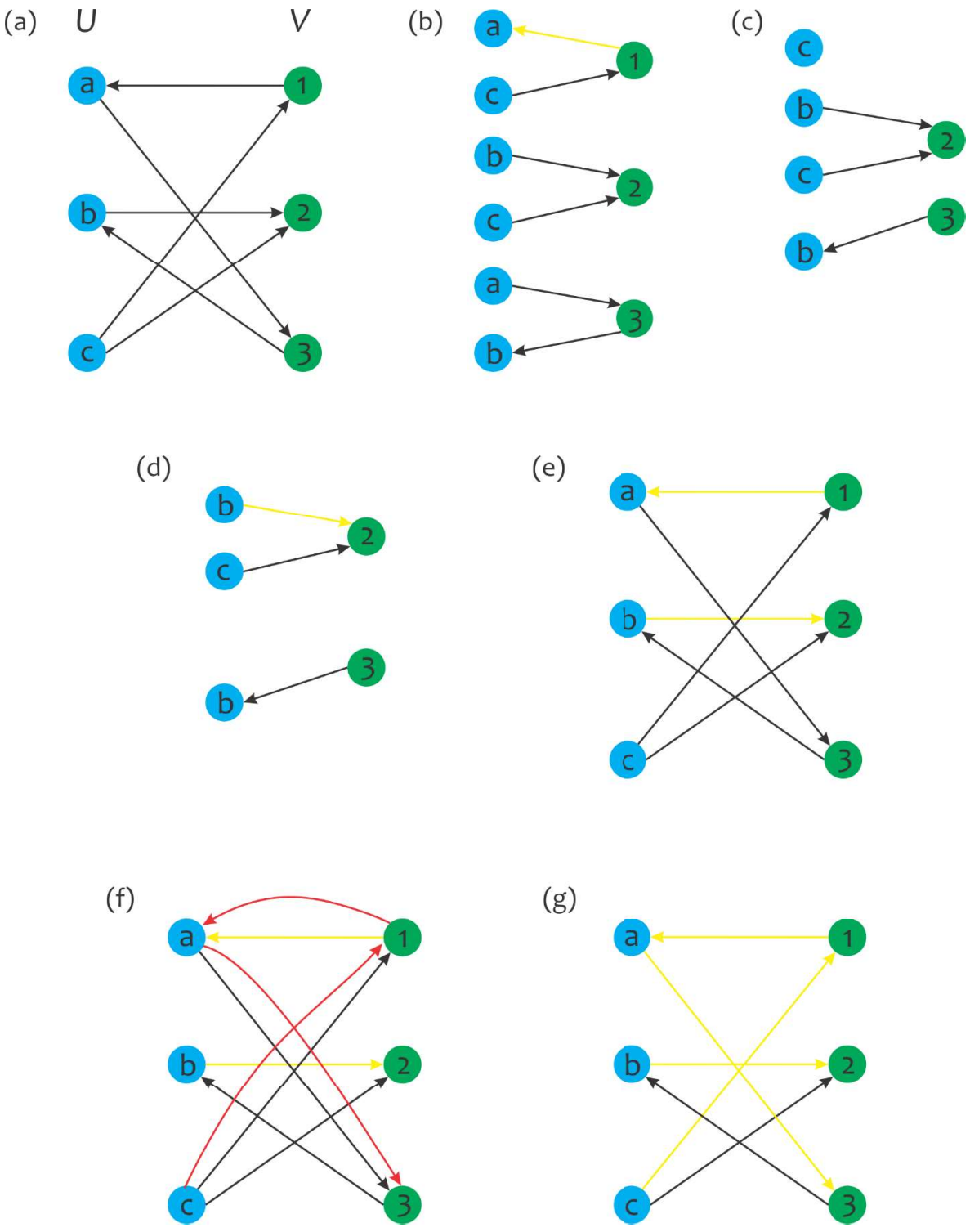

Figure 3.4 : Hopcroft-Karp algorithm for maximum matching (adapted from ["Hopcroft–Karp algorithm," 2017]). (a) A directed graph represented as bipartite with *U* and *V* subsets. (b) Perform BFS starting at all the vertices in V without a match. Pick any unmatched leaf and go all the way back to a root using DFS. Match the leaf to the root. Repeat the process to find the next matching is observed. (c) Delete all the instances of 1 and a found in the trees. (d) Match b to 2 and delete b from the tree spanning from 3. (e) In this iteration, 1 is matched to a, 2 is matched to b, and 3, along with c, are left without a match. (f) Perform DFS once again from the unmatched leaf all the way to the root. (g) Thus produced the maximal matching between *U* and *V* in the graph. The yellow edge represent matching edge red hops represent depth first search steps.

Drawbacks of Hopcroft-Karp algorithm:

1. It cannot tackle the cycle of odd length in a graph because odd number of nodes cannot be divided equally to bipartite graph which is one of the prerequisites of the algorithm.
2. The matching is ambiguous i.e. it depends on the initiation point of each iteration.

To overcome this shortcoming, we used a two-step algorithm developed by Pothen et al. which is a modified strategy from that described by Duff et al. [Duff et al., 2011; Pothen and Fan, 1990].

3.4.4 Maximum matching algorithm

We have utilized this algorithm across this thesis for identification of driver neurons. The algorithmic strategy suggested by Pothen (1990) is as follows [Pothen and Fan, 1990]. The strategy is depicted graphically in Figure 3.5.

Pre-requisite: Adjacency matrix ($N \times N$) of the graph with C columns and r rows

1. Initialize
    a. Set the matching M and the set of unmatched columns U to be empty.
2. Cheap matching
    a. For each column vertex $c \in C$, match c to the first unmatched row vertex $r \in adj(c)$, if there is such a row and if c cannot be matched then add c to U.
3. Augment matching
    a. Create a new unmatched column U_{new} initially as null
    b. For each column vertex $c \in U$ search for an augmenting path from c using either BFS or DFS.
    c. Visit only row vertices that have not been visited previously in this pass.
    d. Mark all visited row vertices.
    e. If an augmenting path is not found, include c in U_{new}.

The time complexity of this algorithm is $O(NE)$.

3.5 STUDY SPECIFIC MATRIEALS AND METHODS

Till now we have explained the materials and methods which have been utilized in this study. Specific methods pertaining to a chapter will be discussed in the respective chapter. Following is the list of specific materials discussed in various chapters.

- The types of neurons based on their function, location and span along with gene ontological enrichment studies are discussed in Chapter 4.
- To simulate synaptic plasticity, we developed a distance constrained synaptic plasticity model which will be discussed in Chapter 5.
- We developed motif tuning algorithm for studies involving motif evolution and control, which are discussed in detail under Chapter 6.
- The details of about structural balance concept and its measurement will be discussed in Chapter 7.

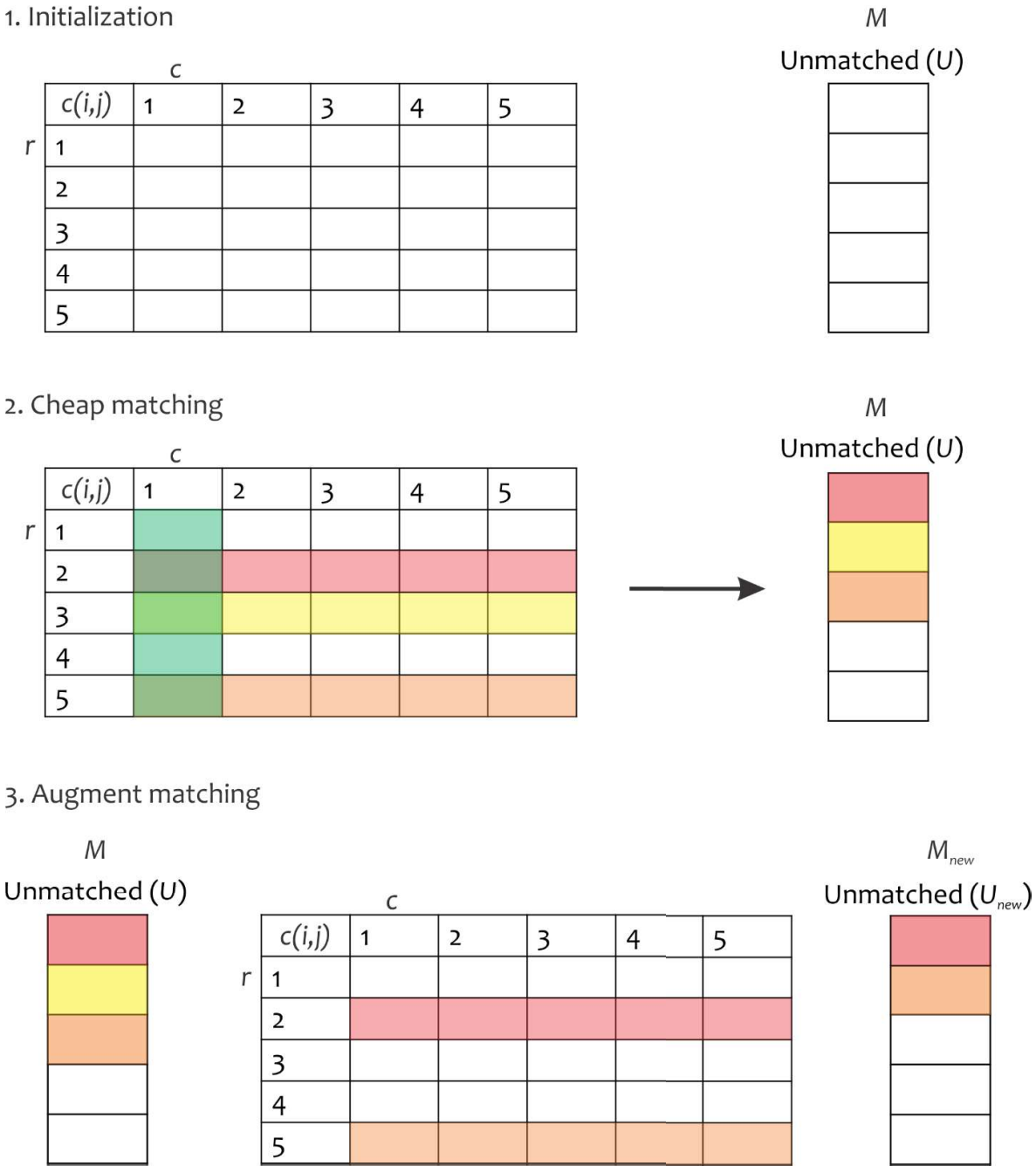

Figure 3.5 : Schematic representation of Augment matching algorithm [Pothen and Fan, 1990].

3.6 BRAIN FUNCTIONAL NETWORKS

Brain functional networks can be build using various techniques that record the functional activity of the brain. As an organ that depends on electrophysiology of its cells, brain generates electrical and magnetic waves which can be picked by the electrodes placed on the scalp. Techniques such as electro encephalogram (EEG) and megnato encephalogram (MEG) exploit this feature of brain activity. Both of these signals captured through these techniques are weak and can only correspond to cortical structures of the brain. Hence, their utility is limited due to inefficient representation of brain's functional activity. Magnetic resonance imaging (MRI) adds another dimension of brain activity and depends on the fact that active neurons utilize oxygen which in turn increases the flow of blood. This consequently changes the blood oxygen level

dependent (BOLD) signal which can be picked up by an MRI machine. The detection of function based on BOLD signal is referred to as functional magnetic resonance imaging (fMRI).

fMRI produces 4D data representing change in intensity of BOLD signal from different regions of the brain (voxels). Thus it presents with an opportunity to look at brain function from a graph theoretical perspective by modeling fMRI data as a network of correlations. We utilized resting state fMRI imaging of pre-diagnosed patient and compared it with healthy individuals to find graph theoretical features of brain functional networks that characterize a neuropathology. A generic framework capturing the deviations in BOLD signals for construction of networks is depicted in Figure 3.6. Detailed protocol of data collection and pre-processing steps used for our studies are explained in Chapter 8.

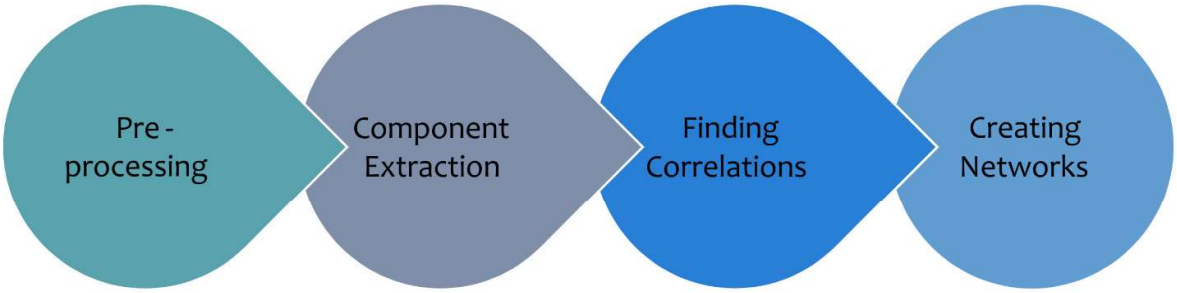

Figure 3.6 : Generic framework for creating a functional network from 4D fMRI data.

…