10-5-2012

Analyzing Self-similar and Fractal Properties of the C. Elegans Neural Network

Tyler M. Reese
University of Connecticut - Storrs

Antoni Brzoska
University of Connecticut - Storrs

Daniel J. Kelleher
University of Connecticut - Storrs

Analyzing Self-Similar and Fractal Properties of the *C. elegans* Neural Network

Tyler M. Reese¹, Antoni Brzoska¹, Dylan T. Yott², Daniel J. Kelleher¹*

1 Department of Mathematics, University of Connecticut, Storrs, Connecticut, United States of America, **2** Department of Mathematics, Boston University, Boston, Massachusetts, United States of America

Abstract

The brain is one of the most studied and highly complex systems in the biological world. While much research has concentrated on studying the brain directly, our focus is the structure of the brain itself: at its core an interconnected network of nodes (neurons). A better understanding of the structural connectivity of the brain should elucidate some of its functional properties. In this paper we analyze the connectome of the nematode *Caenorhabditis elegans*. Consisting of only 302 neurons, it is one of the better-understood neural networks. Using a Laplacian Matrix of the 279-neuron "giant component" of the network, we use an eigenvalue counting function to look for fractal-like self similarity. This matrix representation is also used to plot visualizations of the neural network in eigenfunction coordinates. Small-world properties of the system are examined, including average path length and clustering coefficient. We test for localization of eigenfunctions, using graph energy and spacial variance on these functions. To better understand results, all calculations are also performed on random networks, branching trees, and known fractals, as well as fractals which have been "rewired" to have small-world properties. We propose algorithms for generating Laplacian matrices of each of these graphs.

Introduction

Fractal theory has become an increasingly prevalent topic of both debate and research in recent years. Beginning with Mandelbrot's discussion of Britain's immeasurable coastline [1], fractal analysis has found applications in both the mathematics and scientific communities. In the geometric sense, fractals are objects that contain self-symmetry: they exhibit the same pattern on increasingly smaller scales.

More recently, fractal theory has found applications in the biological realm. Kinetics of ion channels have been modeled with fractal structures [2,3]. Fractal dimension has been used to analyze human EEG signals [4] as well as the complex morphology of living cells [5,6]. The applications of fractal theory in neuroscience have been a particularly prevalent topic of research [7–9]. Glial cells have been analyzed in-depth using fractal dimensions and modeling [10–12]. Dendritic branching has been shown to exhibit self-similarity [13,14], and three-dimensional fractal structures have been used to approximate the white matter surface of the human brain, based on MRI images [15]. Nevertheless, some have warned against the possible misuses of fractal theory in neuroscience [16,17]. In particular, calculations on fractal dimensions of biological systems have been called into question, where some studies have attempted to use this measurement as an overly-generalized tool which lacks definite relation to actual biological mechanisms.

In this paper we use a graph-theoretical approach to probe the structure of the *Caenorhabditis elegans* neural network for self-similarity. Similar work was done by Sporns in examining the

presence of fractal patterns in neuron connectivity: in [18], fractal networks were generated and structural measures were calculated, including small-world properties, complexity, and motif composition. Advances in graph theory have proven useful in analyzing complex neural networks [19–21] as well as underlying motifs in the brain [22,23]. In this paper we apply mathematical techniques to a physical map of the *C. elegans* connectome. With a well-connected component of only 279 neurons, it is an excellent candidate for graph theoretical research on a complete-brain model. While [24] presents a geometric structure of this system, our research builds upon that of [25] in which Varshney et al. propose a finalized schematic of the *C. elegans* neural network.

The *C. elegans* brain is composed of three types of neuronal cells: sensory neurons, motor neurons, and interneurons. Two types of connection exist between these neurons: chemical synapses and gap junctions. The gap junction network, which sends electrical signals via ion transport, is an undirected system. Conversely, chemical synapses possess clear directionality [25]. In this paper we study the overarching connectivity between neurons. In order to analyze the fundamental network-structure of the *C. elegans* neurons, we consider only the skeleton of the brain's organization. Although some neurons share multiple points of contact and chemical synapses send directional signals, we only observe that two neurons are connected. As a result, we study an undirected and unweighted network combining the chemical and gap junctions, representing only the framework of connections (See Methods). In the process, this analysis loses many of the biological details which correspond to functionality and neuron hierarchy.

While such information is vital in understanding the mechanisms acting within the brain, the goal of this study is to mathematically analyze the system's structural connectivity, whereby this simplified network is sufficient.

In order to index each of these connections we use the graph Laplacian matrix, $L = [l_{i,j}]$. For a graph, G, we define d_v as the degree of vertex v: the number of total connections. If vertex u is connected to vertex w then $l_{u,w} = -1$ and $l_{w,u} = -1$, where $l_{i,j}$ corresponds to the entry in the ith row and j^{th} column. Furthermore, $l_{v,v} = d_v$, and all other entries of matrix L are 0. The original goal of this study was to examine the structure of the *C. elegans* neural network for self-similarity, and results were compared to identical calculations on other graphs. One should note that although fractal theory has repeatedly been applied in neuroscience, in studying the structure of a network the results are not as simple as saying "fractal" or "not-fractal." Instead we search specifically for *self-similar* structures in the network's organization.

Results and Discussion

For the *C. elegans* model, we derived a Laplacian matrix from the adjacency matrices used in [25]. Algorithms were developed to produce similar matrices representing random graphs, branching trees, and rewired fractal geometries. These graphs were generated with similar properties as the *C. elegans* neural network, including number of vertices and probability of connection. Details on matrix generation can be found in Methods.

The Eigenvalue Counting Function

The eigenvalue counting function is a cumulative distribution function on the spectrum of a matrix, in this case the Laplacian (see Methods). Plotting this function gives an expedient way to analyze the spectrum of the graph Laplacian [26]. It is known that this function exhibits spectral "gaps" when applied to fractal geometries, corresponding to sections of slope-zero in the plot [27]. The asymptotics of this function have also been shown to be linked with heat dissipation in networks [28]. Figure 1 shows plots of the eigenvalue counting function on the Laplacian matrices.

There is a clear presence of step-like portions of those graphs corresponding to known fractals. These sections of slope-zero correspond to spectral gaps, consistent with expected results. The eigenvalue counting function plot of the *C. elegans* connectome (Fig. 1 (a)) does not show definitive spectral gaps, indicating that the nematode brain is *not* strictly fractal in structure. This, however, does not eliminate the possibility of some degree of self-similarity. Figure 1 (c) contains the graph corresponding to a random-branching tree. The large vertical jump at $x = 1$, with a change on the y-axis of approximately 200, indicates that the eigenvalue 1 occurs with extremely high multiplicity. This is caused by the nature of the tree's organization. There is a large number of endpoints: vertices at which no further branching occurs, connected only to the "parent" vertex. As the highly interconnected neural network is not tree-like, dissimilarity in observed eigenvalue counting patterns is consistent with expected results. Although the eigenvalue counting function of the *C. elegans* neural network does resemble those of the random network (Fig. 1 (e)) and the rewired Sierpinski Gasket (Fig. 1 (f)), this cannot conclusively point to similar structural organization, whereas drastic dissimilarity would point to fundamental differences.

Weyl Ratios

The Weyl ratio of a graph is defined as

$$W(x) = N(x)/x^{\alpha}$$

where $N(x)$ is the eigenvalue counting function. α is determined by the logarithmic asymptotics of the eigenvalue counting function. One way to determine this α is via a linear regression on $N(x)$ when plotted on logarithmic axes. Log-log periodicity of Weyl ratios is present in fractal geometries, and has been observed in graph-approximations of fractals. Similar periodicity in Weyl ratio patterns of other networks can point to self-similarity in these graphs. For more on Weyl ratio analysis on fractals, see [29].

As expected, the Weyl ratios of known self-similar fractals show a high degree of organization. That of the Sierpinski gasket in particular (Fig. 2 (b)) shows unmistakable periodicity. It should be noted that the Weyl ratio graph of the branching-tree (Fig. 2 (c)) is again different from those of other networks, arising from a lack of what we call "looping." In highly interconnected networks, many cyclic paths exist, allowing a signal to arrive back at a starting vertex by traveling through a series of other vertices. Trees, on the other hand, lack this feature: only one path exists between any two points, helping to create a unique Weyl ratio pattern.

While several cases of slight periodicity could be argued for, this evidence is not definitive enough to indicate self-similarity in the *C. elegans* neural network (Fig. 2 (a)). However, there exists similarity between the Weyl ratio patterns generated by the spectrum of the *C. elegans* neural network, the random network (Fig. 2 (e)), and the rewiring of the Sierpinski Gasket (Fig. 2 (f)). The significance of examining a "rewired" fractal structure will become clear later in this paper. It is important to note that this likeness in Weyl ratio patterns can suggest some structural similarity.

The Eigen-Projection Method

We replicated the network visualization performed in [25] and extended this technique to our other networks. This was done in Euclidean space via the eigen-projection method explained in [30], similar to those processes described in [31,32]. This spectral approach to visualizing graphs utilizes the eigenfunctions of degree-normalized Laplacian matrices (see Methods). The eigen-projection method, also known as "plotting in eigenfunction coordinates," plots the vertices of a graph using the eigenfunctions of its Laplacian matrix as a coordinate basis. See Methods for a more rigorous description.

After embedding each vertex in either 2- or 3-dimensional space, neuronal or network connections were represented with line segments between the appropriate points. In the case of the *C. elegans* diagram, the same color-coding as [25] was utilized: where red represents sensory neurons, green are motor neurons, and blue indicates interneurons. Lastly, points were labeled with the corresponding neuron abbreviations. This was done using a slight variation of the VISUALIZE program used by Chklovski and Varshney, available at [33].

The eigen-projection visualizations (Figure 3) allow us to make further qualitative distinctions between the *C. elegans* brain and other networks. In support of previous observations, it is again clear that the nematode connectome is not strictly fractal in structure. On the contrary, the eigenfunction graphs of the Sierpinski Gasket once again display characteristics expected of self-similar fractals: a high degree of ordering and self-symmetry. While the eigenvalue counting function and Weyl ratios showed little distinction between the *C. elegans* brain and a random graph, eigen-projections provide differentiation between the two. The
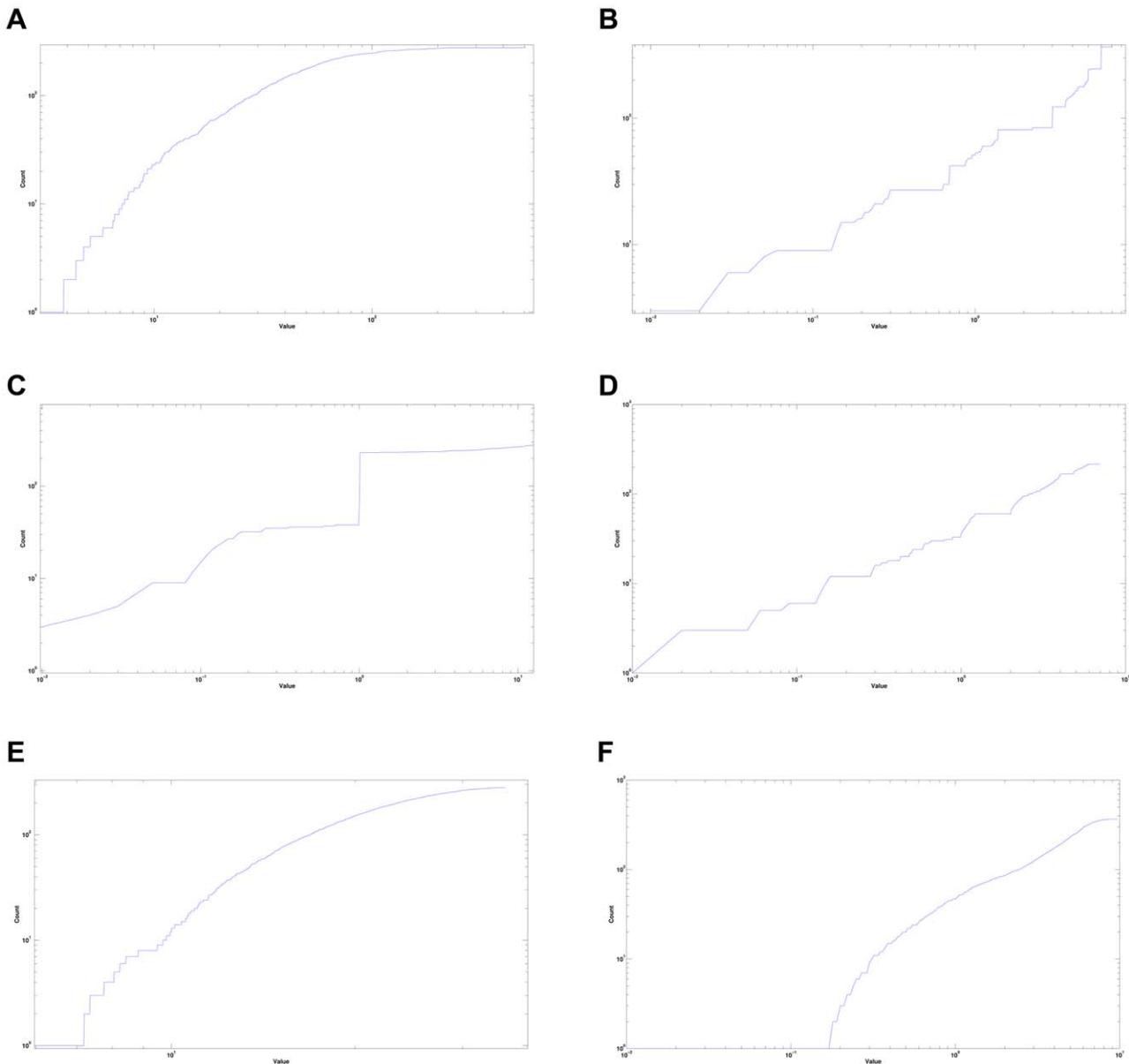
Figure 1. Weyl Ratios (a) *C. elegans* neural network (b) Sierpinski Gasket, Level 5 (c) Random Tree $n = 279, m = 10$ (d) Hexacarpet Level 3 (e) Random Network $n = 279, p = 0.07$ (f) Sierpinski Gasket Rewiring $p = 0.15$.
doi:10.1371/journal.pone.0040483.g001

random graph appears, as expected, more or less a scatter of points. The *C. elegans* brain, however, shows a definite structure with organized connectivity, suggesting that the *C. elegans* neural network is *not* a randomly connected system of neurons. On the other hand, the *C. elegans* neural network maintains its resemblance to a rewired Sierpinski gasket when plotted in eigenfunction coordinates. While there is no effective way to quantify this heuristic similarity in a relevant manner, it sustains its interest experimentally and continues to suggest the presence of some structural parallels.

The eigen-projections display some of the functional organization of the *C. elegans* neural network. It is clear that the neurons are arranged roughly by neuron type. There is a distinctive cluster of motor neurons (green), a larger sub-component of sensory neurons (red), and interneurons interspersed throughout the network (blue). This indicates that although the brain may not posses the strict

self-similarity of a fractal structure, it is indeed highly organized as one would anticipate, developed for entirely functional purposes.

Small-World Network Properties

We consider two functions defined on graphs: average clustering coefficient and average path length. The clustering coefficient of a vertex v, c_v, is the probability that any two vertices neighboring v are also connected to each other. The path length between two vertices u and v is the shortest path along the graph's edges connecting u and v (Note that this path usually travels through a number of other vertices). Using Djisktra's algorithm, it is possible to rigorously determine the shortest path between a given vertex and each other vertex on the graph. By repeating the algorithm for each node on the graph, it is possible to determine the shortest path between each pair of vertices. The average path length, l, is
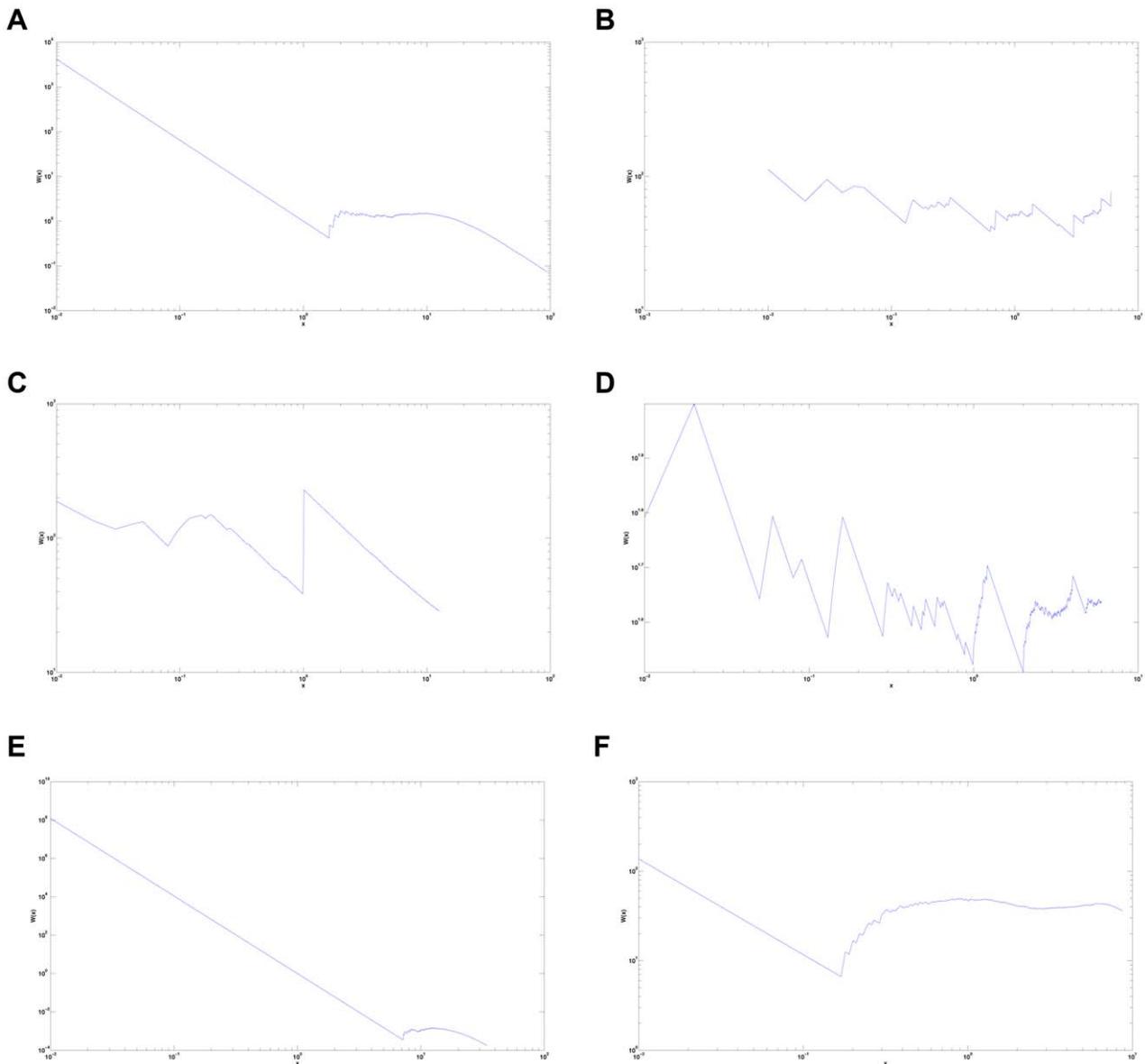
Figure 2. Weyl Ratios (a) *C. elegans* neural network (b) Sierpinski Gasket, Level 5 (c) Random Tree $n = 279, m = 10$ (d) Hexacarpet Level 3 (e) Random Network $n = 279, p = 0.07$ (f) Sierpinski Gasket Rewiring $p = 0.15$.
doi:10.1371/journal.pone.0040483.g002

calculated by finding the arithmetic mean of the shortest paths between each pair of vertices on the graph. Small-world networks are (generally) defined as networks which have a much higher c value than random networks, but maintain a value of l only slightly larger than that of a random network [34].

Small-world networks arise quite often in the natural sciences, as they allow for the efficient transfer of information while maintaining a certain level of complexity. There is a great deal of research which suggests that neural networks possess small-world properties [35,36]. In fact, [25] and [34] have previously demonstrated that the *C. elegans* neural network is small-world in nature. Our calculations of clustering coefficient and average path length confirm these findings. As Table 1 shows, the *C. elegans* neural network has an average path length only slightly larger than that of its associated random network (see Methods for how these 'associated random networks' were constructed). At the same time,

the clustering coefficient for *C. elegans* is six times larger than that of its associated random network, meaning the neural network of *C. elegans* satisfies the small-world properties as defined by [34].

This motivated our work with network-rewiring, related to that done by Watts and Strogatz. In [34] they showed that moving connections in an ordered network, with a certain probability p, led to some interesting changes in graph structure. Namely, when p is small, a slight increase in p causes a large drop in l but does not change c appreciably: the network takes on small-world characteristics. Intuitively this can be explained by the fact that these sparse random connections don't change a graph's strong localized structure, but it becomes easier to travel long distances via these new connections which can span large gaps. It is clear from Table 1 that the Sierpinski Gasket does not possess small-world characteristics. This supports the propositions of [37], which describes the existence of a dichotomy between fractal structures
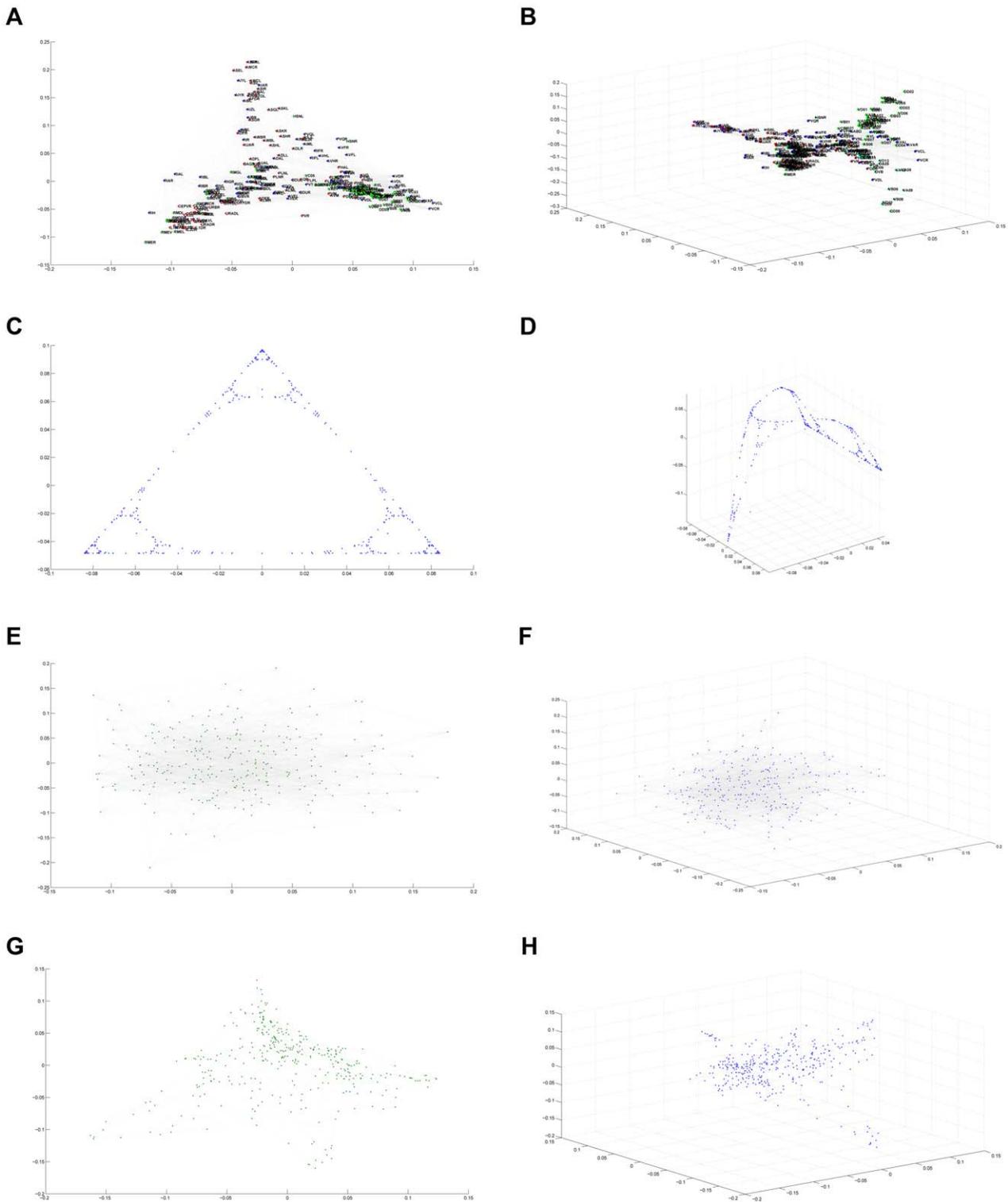
Figure 3. The Eigen Projection Method (a) *C. elegans* neural network, (φ_2, φ_3) (b) *C. elegans* neural network, $(\varphi_2, \varphi_3, \varphi_4)$ (c) Sierpinski Gasket, Level 5, (φ_2, φ_3) (d) Sierpinski Gasket, Level 5, $(\varphi_2, \varphi_3, \varphi_4)$ (e) Random Network $n = 279, p = 0.07$, (φ_2, φ_3) (f) Random Network $n = 279, p = 0.07$, $(\varphi_2, \varphi_3, \varphi_4)$ (g) Sierpinski Gasket Rewiring $p = 0.15$, (φ_2, φ_3) (h) Sierpinski Gasket Rewiring $p = 0.15$, $(\varphi_2, \varphi_3, \varphi_4)$.
doi:10.1371/journal.pone.0040483.g003

and small world networks. As a result, the Laplacian matrix of a small-world rewiring of the Sierpinski Gasket was included throughout this study. Sporns used a similar method in [18] by

generating fractal connections and ''rewiring'' such networks for small-world properties in studying complexity and self similarity in neuron connectivity.

Table 1. Clustering coefficient and path length.

Graph	Clustering Coefficient	Average Path Length
Sierpinski Gasket, Level 5	0.4495	17.3721
Random(Sierpinski Gasket)	0.0104	5.748
Sierpinski Gasket Rewire $p = 0.15$	0.2843	7.3833
Random(SG Rewire)	0.0104	5.748
C. elegans Neural Network	0.3371	2.5377
Random(*C. elegans* Neural Network)	0.0581	2.3458

Energies and Spacial Variances

Using an eigenfunction of a graph's Laplacian, φ, one can calculate a graph energy specific to φ. Knowing the resistance between any two vertices and a constant γ, these can be used to calculate the spacial variance of φ (See Methods). Variance is a measure of how localized a function is, and functions which possess a low spacial variance are said to be localized. In particular, localization occurs when an eigenfunction is approximately zero except for in a a small (localized) region: that is, the eigenfunction takes on most of its values inside a small number of connected regions on the graph. Localized eigenfunctions are known to be present in fractals but not in Euclidean or other smooth spaces [38]. Figure 4 shows distributions of the spacial variance of all eigenfunctions on the graphs used previously.

Of the four graphs considered here, the eigenfunctions of the random network possess the largest spacial variances. As this system was designed to lack general organization, non-localized eigenfunctions were both expected and observed. The eigenfunctions of the Sierpinski Gasket possess both highly concentrated and low-valued spacial variances. Such trends correspond to a high degree of localization, as anticipated in approximations of fractal geometries. Eigenfunctions of the rewired Sierpinski Gasket demonstrate a similar concentration pattern with slightly higher

spacial variance, indicating a slightly lower degree of eigenfunction localization.

Of particular interest are the spacial variances of eigenfunctions of the *C. elegans* neural network. Although these spacial variances do not show the same degree of concentration as those of the Sierpinski Gasket, the *values* of these variances are a power of 10 less than those of the fractal network. Whereas spacial variances on eigenfunctions of the Sierpinski Gasket are concentrated around 5×10^{-3} the majority of eigenfunctions of the neural network lie below 5×10^{-4}. These comparatively lower spacial variances indicate a high level of localization in the eigenfunctions. Such highly localized eigenfunctions can indicate the presence of self-similarity in the network, although further tests would be required to determine the absolute origin of this localization.

In [25] sparsity of eigenfunctions suggests the presence of subcircuits with a specific function. As localization suggests that the value of of an eigenfunction is concentrated on a few vertices, it is reasonable that localized eigenfunction may be used in place of sparse ones. In light of this, the existence of localized eigenfunctions is not entirely unexpected. However [25] looks for sparsity in eigenfunctions of the gap junction network only.
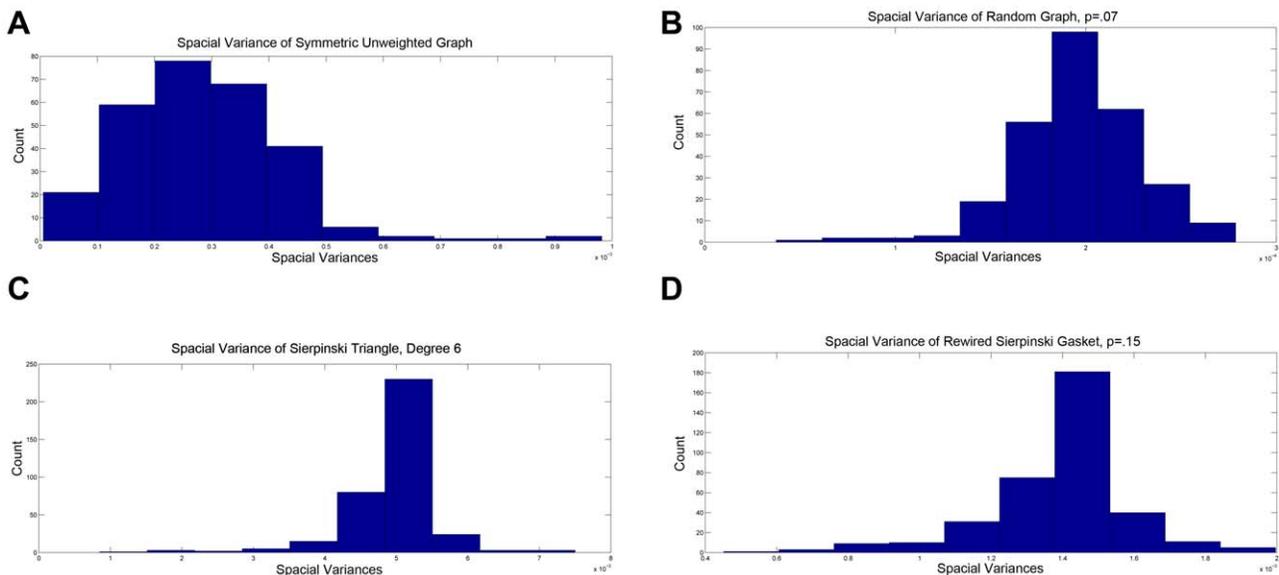

Figure 4. Spacial Variance (a) *C. elegans* neural network (b) Random Graph ($n = 279, p = 0.07$) (c) Sierpinski Gasket, Level 5 (d) Sierpinski Gasket Rewiring ($p = 0.15$).

Conclusions

In this paper we used a variety of graph theoretic and mathematical techniques to probe the structural framework of the *C. elegans* connectome. To better understand results, calculations were also performed on random networks, finite approximations of fractal geometries, and small-world "rewired" graphs. This study confirmed previous results, demonstrating that the neural network exhibits small-world characteristics. Furthermore, the network has highly localized eigenfunctions, which could suggest the presence of self-similar structural motifs. Further research would be required to determine the nature of this localization. Although the *C. elegans* neural network is not random, tree-like, nor fractal in structure, it is certainly highly ordered, aiding in functional efficiency of the system. Although *C. elegans* has proven to be a useful model organism, with a well-defined map of its neural network, this network consists of only 279 nodes. While this makes the system fairly efficient to study computationally, this small number of nodes makes network-analysis rather limited. Although ideally a much larger map would be used, due to the difficulty in determining the exact layout of each neuron in a network, few consistent complete-brain maps exist at this time. In the end, this paper presents a type of analytic "toolbox," offering many mathematical techniques which can be used to search for structural self-similarity within networks. In particular, the tools presented here can be used to study higher-order brains as more complex neural networks become well-understood.

Methods

In order to analyze only the framework of the *C. elegans* neural network, we constructed a Laplacian matrix derived form the adjacency matrices in [33]. The network of chemical synapses sends signals in one direction only, resulting in a non-symmetric adjacency matrix, C. To disregard this directionality, we added this matrix to its own transpose, C', creating a symmetric matrix indexing all chemical connections. We added this matrix to the adjacency matrix of the gap junction system, G (already symmetric as these connections are bidirectional).

$$B = [b_{i,j}] = (C + C') + G \text{ such that } i,j \leq 279$$

All non-zero entries of this combined matrix, B, were normalized to be 1, avoiding multiplicity of connection, resulting in matrix A.

$$A = [a_{i,j}] \text{ where } a_{i,j} = 1 \text{ if } b_{i,j} > 0, \text{otherwise } a_{i,j} = 0 \text{ when } b_{i,j} = 0$$

It is then simple to produce a Laplacian matrix, L, as shown below:

$$d_j = \sum_{i=1}^{i=279} a_{i,j} \text{ for each } j \leq 279$$

Note that d_j is the degree of each vertex j. The degree matrix, D, is now defined as:

$$D = [d_{i,j}] \text{ where } d_{i,j} = d_j \text{when } i = j \text{ otherwise } d_{i,j} = 0$$

Then the Laplaican matrix, L, is given by:

$$L = D - A$$

Generating Random Graphs and Trees

In order to generate a Laplacian matrix representation of the random graphs, we used the following algorithm:

First, fix the number of vertices, n, and the probability of connection, p, and construct an empty $n \times n$ matrix, $R = [r_{i,j}]$.

For each $r_{i,j}$ such that $i < j$ assign a random value $a_{i,j}$ such that $0 \leq a_{i,j} \leq 1$ for all $i,j \leq n$. If $a_{i,j} \leq p$ then $r_{i,j} = 1$, otherwise $r_{i,j} = 0$.

To produce an adjacency matrix of this graph, A, add R to its own transpose:

$$A = R + R'$$

Using this adjacency matrix, construct a Laplacian matrix using the method described previously.

The algorithm used for producing the Laplacian matrix of a random-branching tree is more involved. Again, fix the number of vertices, n, and also specify the maximum number of "children" from any given branch-point, m. Create an empty $n \times n$ matrix, $T = [t_{i,j}]$

Begin by generating a random integer a_1 such that $0 < a_1 \leq m$, and take $t_{1,1} = a_1$. This corresponds to the first vertex having $|a_1|$ branches. To represent these branches in the matrix, take $t_{1,j} = -1$ for $j = 2, \cdots, a_1 + 1$ and $t_{i,1} = -1$ for $i = 2, \cdots, a_1 + 1$.

Next move to all subsequent vertices. Because no "looping" exists in the structure of the tree, each node can only be connected to its parent vertex and its "children" vertices. We take $S = \{j : t_{i,j} = 0 \text{ for all } i \leq n\}$ Then k, where $k = \min(S)$ is the smallest-labeled node which does not have a parent vertex, i.e. the first column with all 0 entries corresponds to the first point not yet connected. (Note in the case of vertex 2, $k = a_1 + 2$). This vertex k is the first "offspring" from the next branch-point.

Now, as above, for each remaining vertex v we choose another random integer, a_v, such that $0 < a_v \leq \min(m, n - k + 1)$ and take $t_{v,v} = a_v + 1$. (Note that vertex v has $|a_v|$ children, however $a_v + 1$ is the degree of node v, taking into account its parent-connection). To represent the "offspring" branches of this vertex v, use the following:

$$t_{i,v} = -1 \text{ for } i = k, k+1, \cdots, k + (a_v - 1)$$

and

$$t_{v,j} = -1 \text{ for } j = k, k+1, \cdots, k + (a_v - 1)$$

Use $\min(m, n - k + 1)$ when choosing a_v to avoid adding more vertices than the n which was originally fixed.

The Eigenvalue Counting Function and Weyl Ratios

For a given graph Laplacian matrix, L, the eigenvalue counting function, $N(x)$ is a cumulative frequency function on the spectrum of the matrix where:

$$N(x) = \#\{\lambda_j \leq x\} \text{ where each } \lambda_j \text{ is an eigenvalue of } L$$

The growth of $N(x)$ is approximately x^α, thus the relevant portion of each graph, when using a logarithmic scale, appears linear. A linear regression was found for each relevant interval, and the slope, α, calculated. Using this α, we plotted the Weyl ratio, $W(x)$, such that:

$$W(x) = N(x)/x^\alpha.$$

Normalizing a Laplacian Matrix

We used two different forms of the graph-Laplacian matrix: the standard Laplacian and the degree-normalized Laplacian. In the case of eigen-projections, we utilize the degree-normalized matrix. We define the degree matrix, D, as before: a diagonal matrix whose non-diagonal elements are 0, and each entry $d_{j,j}$ is the degree of the jth vertex. Using both the standard Laplacian, L and its corresponding degree matrix, D, produce the degree-normalized Laplacian, Q:

$$Q = D^{-1/2} L D^{-1/2}$$

Using the degree-normalized Laplacian has many aesthetic advantages, as shown in Figure 5. The normalized matrix also has all eigenvalues λ_j such that $0 \leq \lambda_j \leq 2$.

Graphing in Eigenfunction Coordinates

We found all eigenvalues, λ_k, and their corresponding eigenfunctions, φ_k, for each matrix. Given two eigenfunctions φ_i and φ_j, (such that $i \neq j$) we then plotted the ordered pair $(\varphi_i(n), \varphi_j(n))$ for each n from 1 to 279, as described in [39]. The first eigenvalue of any Laplacian matrix is always 0, corresponding to a constant eigenfunction. Thus we only consider φ_i and φ_j with $i, j \geq 2$. Edges were then added between points to represent relevant connections, and the same color-coding as [25] was used. The same process was then repeated in three dimensions, plotting $(\varphi_i(n), \varphi_j(n), \varphi_k(n))$ for some $i, j, k \geq 2$, such that $i \neq j \neq k$.

Clustering Coefficient

The clustering coefficient measures the probability that two neighbors of a given vertex are also connected to one another. For a graph G and a given vertex v, let e_v denote the number of connections that exist between the neighbors of v. Take d_v as the number of neighbors of v (the degree of vertex v). Then the clustering coefficient of vertex v, c_v, is given by:

$$c_v = \frac{2e_v}{d_v(d_v - 1)}$$

Note that total number of possible connections among neighbors of v is $\frac{d_v(d_v - 1)}{2}$.

For a graph G with n vertices, the average clustering coefficient, c, is defined as:

$$c = \frac{1}{n} \sum_{v=1}^{n} c_v$$

Generating a Related Random Graph for Small-World Analysis

In order to analyze our networks for small-world properties, it was useful to compare these graphs to those of similar networks with randomly assigned edges. Small-world networks are nearly as well-connected as random graphs, but possess a well-localized structure. We developed the following algorithm for this process:

For a graph G with n vertices, let k be the number of edges on G. The average number of edges per vertex is k/n. Furthermore, the probability that any two random vertices are connected, p, is given by the number of existing connections divided by the total possible connections:

$$p = \frac{k}{\dfrac{n(n-1)}{2}} = \frac{2k}{n(n-1)}$$

Next we generate a random graph, $Rand(G)$, with n vertices and a p probability of connection between two vertices (See Methods). We then compute c and l for G and $Rand(G)$.

Graph Rewiring

First number each vertex in G from 1 to n, the total number of vertices. If there is a connection between vertices u and v in G, we generate a random number between 0 and 1. If this random number is less than a given probability p, then the connection will be rewired. Without loss of generality assume $u < v$. We then fix the connection to vertex u, and move the connection to another vertex, k, such that u and k are now connected whereas they were not previously.

Graph Energy

For a graph $G = (V, E)$ where V is the set of vertices and E is the set of edges, one can define an arbitrary scalar function $u : V \to \mathbb{R}$. The energy of u associated with G, $\mathcal{E}(u)$ is defined as:

$$\mathcal{E}(u) = \sum_{x,y \in E} (u(x) - u(y))^2.$$

We analyzed the energies of the Laplacian matrix eigenfunctions, thus $u = \varphi$.

Spacial Variance

In order to discuss spacial variance, we must first define the resistance between two vertices on a graph. Let $G = (V, E)$ be a graph and $x, y \in V$. Then the resistance between x and y, $d(x, y)$, is given by:

$$d(x,y) = \mathcal{E}(h(x,y))^{-1}$$

Where $h(x,y)$ is a harmonic function defined as follows:

Let $G = (V, E)$ be a graph and $x, y \in V$. Then the harmonic function corresponding to (x, y) is a scalar function $h(x,y) : V \to \mathbb{R}$ such that:

1. $h(x,y)(x) = 0$

2. $h(x,y)(y) = 1$

1. $\mathcal{E}(h)^{-1}$, where h is an arbitrary scalar function on V, maximized at $h(x,y)$.

Finding the harmonic function is equivalent to finding a vector h such that $Lh = z$, where z is a vector whose entries are all 0 except for those entries corresponding to x and y. This is analogous to what "harmonic" means in Euclidean space. This changes the maximization problem in condition 3 to solving a system of linear equations.

These Weyl ratios allow us to examine the spectrum of each matrix, looking for elements such as symmetry and periodicity [29].

Figure 5. Normalizing the Laplacian *C. elegans* neural network: (a) Un-normalized Laplacian, (φ_2, φ_3) (b) Normalized Laplacian (φ_2, φ_3).
doi:10.1371/journal.pone.0040483.g005

Using these we can now define the spacial variance of a graph. Again, let $G = (V, E)$ be a graph with n vertices and u be a scalar function on V. Let γ be a constant. Then the γth spacial variance of u over G is given by:

$$Var_\gamma(u) = \frac{1}{n} \sum_{x,y \in E} d(x,y)^\gamma (u(x) - u(y))^2$$

In this paper, the spacial variances of eigenfunctions of Laplacian matrices were evaluated at $\gamma = 1$.

Acknowledgments

Author Contributions

Conceived and designed the experiments: DJK. Analyzed the data: DJK TMR DTY AB. Wrote the paper: DJK TMR DTY AB. Wrote programs for analysis: TMR DTY AB.

References

1. Mandelbrot BB (1967) How long is the coast of Britain? Statistical self-similarity and fractal dimension. Science 156: 636–638.
2. Liebovitch LS, Fischbarg J, Koniarek JP, Todorova I, Wang M (1987) Fractal model of ion-channel kinetics. Biochim Biophys Acta 896: 173–180.
3. Lowen SB, Liebovitch LS, White JA (1999) Fractal ion-channel behavior generates fractal firing patterns in neuronal models. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics 59: 5970–5980.
4. Paramanathan P, Uthayakumar R (2008) Application of fractal theory in analysis of human elec-troencephalographic signals. Comput Biol Med 38: 372–378.
5. Bernard F, Bossu JL, Gaillard S (2001) Identification of living oligodendrocyte developmental stages by fractal analysis of cell morphology. J Neurosci Res 65: 439–445.
6. Smith TG, Lange GD, Marks WB (1996) Fractal methods and results in cellular morphology–dimensions, lacunarity and multifractals. J Neurosci Methods 69: 123–136.
7. Fernandez E, Jelinek HF (2001) Use of fractal theory in neuroscience: methods, advantages, and potential problems. Instituto de Bioingenier ia 24: 309–321.
8. Kiselev VG, Hahn KR, Auer DP (2003) Is the brain cortex a fractal? Neuroimage 20: 1765–1774.
9. Werner G (2010) Fractals in the nervous system: conceptual implications for theoretical neuro-science. Front Physiol 1: 15.
10. Smith TG, Behar TN (1994) Comparative fractal analysis of cultured glia derived from optic nerve and brain demonstrate different rates of morphological differentiation. Brain Res 634: 181–190.
11. Smith TG, Behar TN, Lange GD, Marks WB, Sheriff WH (1991) A fractal analysis of cultured rat optic nerve glial growth and differentiation. Neuroscience 41: 159–166.
12. Reichenbach A, Siegel A, Senitz D, Smith TG (1992) A comparative fractal analysis of various mammalian astroglial cell types. Neuroimage 1: 69–77.
13. Caserta F, Stanley HE, Eldred WD, Daccord G, Hausman RE, et al. (1990) Physical mechanisms underlying neurite outgrowth: A quantitative analysis of neuronal shape. Phys Rev Lett 64: 95–98.
14. Bieberich E (2002) Recurrent fractal neural networks: a strategy for the exchange of local and global information processing in the brain. BioSystems 66: 145–164.
15. Free SL, Sisodiya SM, Cook MJ, Fish DR, Shorvon SD (1996) Three-dimensional fractal analysis of the white matter surface from magnetic resonance images of the human brain. Cereb Cortex 6: 830–836.
16. Jelinek HF, Fernández E (1998) Neurons and fractals: how reliable and useful are calculations of fractal dimensions? J Neurosci Methods 81: 9–18.
17. Murray J (1995) Use and Abuse of Fractal Theory in Neuroscience. The Journal of Comparative Neurology 361: 369–371.
18. Sporns O (2006) Small-world connectivity, motif composition, and complexity of fractal neuronal connections. BioSystems 85: 55–64.
19. Bullmore E, Sporns O (2009) Complex brain networks: graph theoretical analysis of structural and functional systems. Nat Rev Neurosci 10: 186–198.
20. Stam CJ, Reijneveld JC (2007) Graph theoretical analysis of complex networks in the brain. Non-linear Biomed Phys 1: 3.
21. Fallani FV, Costa LF, Rodriguez FA, Astolfi L, Vecchiato G, et al. (2010) A graph-theoretical approach in brain functional networks. Possible implications in EEG studies. Nonlinear Biomed Phys 4 Suppl 1: S8.
22. Sporns O, Kotter R (2004) Motifs in brain networks. PLoS Biol 2: 1910–1918.
23. Itzkovitz S, Alon U (2005) Subgraphs and network motifs in geometric networks. Phys Rev E Stat Nonlin Soft Matter Phys 71: 026117.
24. Morita S, Oshio Ki, Osana Y, Funabashi Y, Oka K, et al. (2001) Geometrical structure of the neuronal network of Caenorhabditis elegans. Physica A 298: 553–561.
25. Varshney LR, Chen BL, Paniagua E, Hall DH, Chklovskii DB (2011) Structural Properties of the Caenorhabditis elegans Neuronal Network. PLoS Comput Bio 7 7: e1001066.
26. Das KC (2004) The Laplacian spectrum of a graph. Comput Math Appl 48: 715–724.
27. Zhou D (2008) Spectral analysis of Laplacians on certain fractals. ProQuest LLC, Ann Arbor, MI, 109 pp. Thesis (Ph.D.)–University of Waterloo (Canada).
28. Begue M, DeValve L, Miller D, Steinhurst B (2009) Spectrum and Heat Kernel Asymptotics on General Laakso Spaces. Fractals (to appear).
29. Berry T, Heilman SM, Strichartz RS (2009) Outer approximation of the spectrum of a fractal Laplacian. Experiment Math 18: 449–480.
30. Koren Y (2005) Drawing graphs by eigenvectors: theory and practice. Comput Math Appl 49: 1867–1888.
31. Mohar B (1991) The Laplacian spectrum of graphs. In: Graph theory, combinatorics, and applications. Vol. 2 (Kalamazoo, MI, 1988), New York: Wiley, Wiley-Intersci. Publ. pp. 871–898.
32. Pisanki T, Shawe-Taylor J (2000) Characterizing Graph Drawing with Eigenvectors. J Chem Inf Comput Sci 40: 567–571.
33. Varshney L, Chen B, Paniagua E, Hall D, Chklovskii D (2011) Structural properties of the Caenorhabditis elegans neuronal network. Available: http://mitedu/lrv/www/elegans/.
34. Watts DJ, Stogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393: 440–442.
35. Bassett DS, Meyer-Lindenberg A, Achard S, Duke T, Bullmore E (2006) Adaptive reconfiguration of fractal small-world human brain functional networks. Proc Natl Acad Sci USA 103: 19518–19523.
36. Sporns O, Honey CJ (2006) Small worlds inside big brains. Proc Natl Acad Sci 103: 19219–19220.
37. Csányi G, Szendröi B (2004) Fractal-small-world dichotomy in real-world networks. Phys Rev E Stat Nonlin Soft Matter Phys 70: 016122.
38. Okoudjou KA, Saloff-Coste L, Teplyaev A (2008) Weak uncertainty principle for fractals, graphs and metric measure spaces. Trans Amer Math Soc 360: 3857–3873.
39. Begue M, Kelleher DJ, Nelson A, Panzo H, Pellico R, et al. (2011) Random walks on barycentric subdivisions and the Strichartz hexacarpet. ArXiv e-prints.