



Research

Cite this article: Chandrasekhar A, Navlakha S. 2019 Neural arbors are Pareto optimal. *Proc. R. Soc. B* **286**: 20182727. <http://dx.doi.org/10.1098/rspb.2018.2727>

Received: 03 December 2018
Accepted: 09 April 2019

Subject Category:

Neuroscience and cognition

Subject Areas:

computational biology, neuroscience

Keywords:

biological networks, neural arbors, Pareto optimality, graph theory, plant architectures

Author for correspondence:

Saket Navlakha
e-mail: navlakha@salk.edu

Electronic supplementary material is available online at <https://dx.doi.org/10.6084/m9.figshare.c.4472489>.

Neural arbors are Pareto optimal

Arjun Chandrasekhar^{1,2} and Saket Navlakha^{1,2}

¹Bioinformatics and Systems Biology Program, University of California, San Diego, UK

²Integrative Biology Laboratory, The Salk Institute for Biological Studies, La Jolla, CA 92037, USA

SN, 0000-0002-5505-9718

Neural arbors (dendrites and axons) can be viewed as graphs connecting the cell body of a neuron to various pre- and post-synaptic partners. Several constraints have been proposed on the topology of these graphs, such as minimizing the amount of wire needed to construct the arbor (wiring cost), and minimizing the graph distances between the cell body and synaptic partners (conduction delay). These two objectives compete with each other—optimizing one results in poorer performance on the other. Here, we describe how well neural arbors resolve this network design trade-off using the theory of Pareto optimality. We develop an algorithm to generate arbors that near-optimally balance between these two objectives, and demonstrate that this algorithm improves over previous algorithms. We then use this algorithm to study how close neural arbors are to being Pareto optimal. Analysing 14 145 arbors across numerous brain regions, species and cell types, we find that neural arbors are much closer to being Pareto optimal than would be expected by chance and other reasonable baselines. We also investigate how the location of the arbor on the Pareto front, and the distance from the arbor to the Pareto front, can be used to classify between some arbor types (e.g. axons versus dendrites, or different cell types), highlighting a new potential connection between arbor structure and function. Finally, using this framework, we find that another biological branching structure—plant shoot architectures used to collect and distribute nutrients—are also Pareto optimal, suggesting shared principles of network design between two systems separated by millions of years of evolution.

1. Introduction

Both man-made and biological transport networks face trade-offs in their design. While resources can be spent to increase the speed or reliability of transport, this often comes at a cost. For example, in road networks, users wish to minimize the travel time to get from one point in a city to another, but this goal conflicts with the practical need to minimize costs in building infrastructure. In such cases, engineers typically seek to design network topologies that achieve the best ‘bang for the buck’. Here, we analyse how well neural arbors resolve a similar performance-versus-cost trade-off.

Neural arbors (dendrites and axons) can be viewed as transport networks rooted at the cell body that process and relay information from one neuron to another [1]. Prior work has proposed competing constraints on the topology of these networks [2]. For example, for both dendrites and axons, the principle of wiring economy states that the total length of an arbor should be minimized [3–8]; total wire is a measure of network cost and is especially important when space is limited or when wire is a commodity [9]. On the other hand, Cajal’s principle of conduction delay states that, for axons, the distance or time required to propagate an action potential from the soma (cell body/root) to downstream post-synaptic partners should be minimized for efficient signal propagation [10]. For dendrites, minimizing conduction delay similarly allows for efficient propagation of dendritic action potentials [11] or back-propagating action potentials [12], and it minimizes the effects of attenuation (weakening of the signal as it travels from a synapse to the soma) [13].

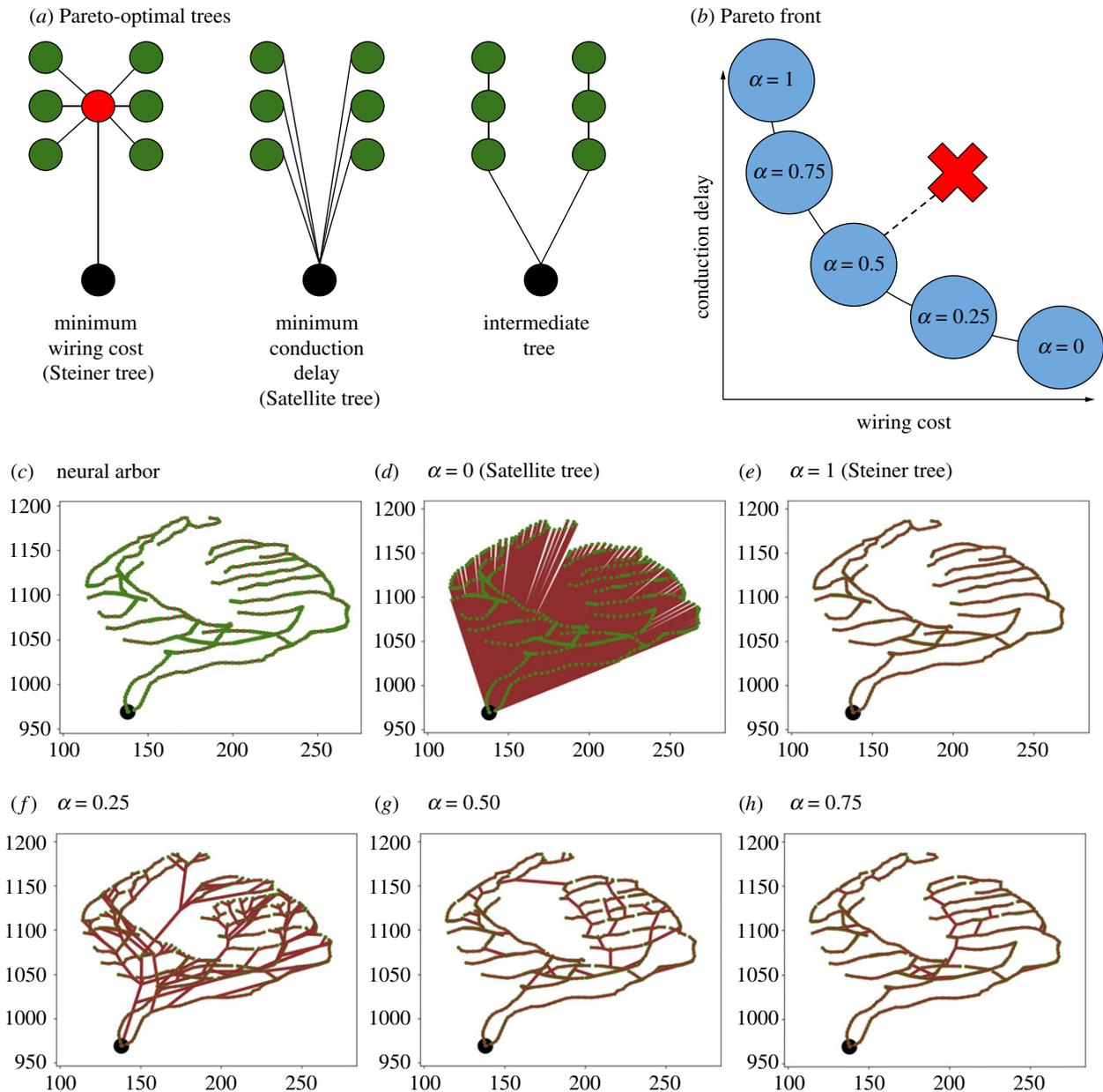


Figure 1. Illustration of Pareto optimal trees and the Pareto front. (a) For a given set of points (green), there are several possible trees that can be formed, each optimizing wiring cost and conduction delay differently. The black node indicates the root (cell soma) of the arbor, and the red points indicate branch points. The Steiner tree minimizes wiring cost. The Satellite tree minimizes conduction delay. Intermediate trees lie in between. (b) The Pareto front defines a set of trees for which improving upon one objective requires a loss in the other objective. The value of $\alpha \in [0, 1]$ indicates a prioritization weight that the arbor places on one objective versus the other. We evaluate how Pareto optimal a given tree (red 'X') is by computing the distance from that tree to the Pareto front. (c–h) Example Pareto optimal trees generated by our greedy algorithm using different values of α . The black dot is the soma, the green dots are synapses, brown dots are Steiner points and red lines are edges. The axes are coordinates, in micrometres.

The two topologies that minimize wiring economy and conduction delay, respectively, are often different (figure 1a). A Steiner tree [14] minimizes wiring economy, but this structure could have poor conduction delay for some nodes. On the other hand, the Satellite tree, an arbor that connects the cell body directly to each synaptic partner via a straight line would optimally minimize conduction delay, but this structure would be very inefficient for wiring economy.

Here, we use the theory of Pareto optimality to analyse how well neural arbors resolve this network design trade-off. Intuitively, a network is Pareto optimal if there does not exist an alternative topology that improves on one objective without hurting the other. Given enough evolutionary pressure and genetic diversity, we hypothesize that

mechanisms generating suboptimal topologies would be eliminated from the population [15]. The only topologies that would remain in the population are those that lie on the *Pareto front*, consisting of all topologies that are Pareto optimal.

We formalize this problem, develop a graph-theoretic algorithm to generate close to Pareto optimal topologies, and we evaluate how well neural arbors come to realizing Pareto optimality. In particular, we analysed the structure of 14 145 arbors from a variety of species, cell types, and brain regions and find that most arbors lie on or very close to the Pareto front, and that this is highly unlikely to occur by chance (as measured by three baseline graph models that do not seek to directly optimize either wiring cost or conduction delay). One advantage of this approach is that

each individual arbor can be associated with a single parameter ($\alpha \in [0, 1]$), indicating how the arbor weighs or prioritizes each objective. We find that the α values can be used to classify some functional categories of neurons based on their structure (e.g. axons versus dendrites, or different cell types or neurotransmitter types). Finally, to show the generality of this result to other biological branching structures, we re-analyse tracings of hundreds of plant shoot architectures and find that plants are also Pareto optimal. To our knowledge, this is the first quantified comparison between plant architectures and neural arbors based on their ability to trade-off two common network design criteria.

For additional related work in the study of neural and plant arbors [3–5,7,8,10,16–31] and theoretical computer science [32–39], see the electronic supplementary material.

2. Theory

(a) A framework for analysing neural arbors as transport graphs

As input, we are provided a set of points $\mathcal{P} = \{p_1, p_2, \dots, p_n; r\} \in \mathbb{R}^3$. The points p_i represent the three-dimensional locations of the n synapses that the arbor connects to (pre-synaptic for dendrites, and post-synaptic for the axon). The point r represents the location of the cell body. Our goal is to output a tree $T_{\mathcal{P}} = (V_{\mathcal{P}}, E_{\mathcal{P}})$ rooted at r that connects the points p_i , where $V_{\mathcal{P}}$ contains all points in \mathcal{P} and can include additional branch points not present in the input for more efficient connectivity. The edges $E_{\mathcal{P}} \subseteq V_{\mathcal{P}} \times V_{\mathcal{P}}$ are such that $T_{\mathcal{P}}$ is a tree, i.e. it is connected with no cycles. Neural arbors typically do not pre-define the precise locations of synaptic contacts ahead of time, but rather connect to potential partners as they grow [40,41]. Our model is not meant to characterize the process by which neural arbors grow and branch. Rather, given how a neural arbor branched, our model evaluates how optimal the arbor morphology was compared to all other possible morphologies that could have been used to connect the cell body to the synapses p_i .

We evaluate the wiring cost and conduction delay of the arbor as follows. For edge $(u, v) \in E_{\mathcal{P}}$, define the edge length, $l(u, v)$ to be the Euclidean distance between u and v . For two points $u, v \in V_{\mathcal{P}}$, define the graph distance $d(u, v)$ to be the length of the shortest path from u to v using the edges $E_{\mathcal{P}}$.

The *wiring cost* is the total length of all edges in the tree:

$$W(T_{\mathcal{P}}) = \sum_{(u,v) \in E_{\mathcal{P}}} l(u, v). \quad (2.1)$$

The *conduction delay* is the sum of the graph distances from the root to each synapse:

$$D(T_{\mathcal{P}}) = \sum_{p_i \in \mathcal{P} \setminus \{r\}} d(r, p_i). \quad (2.2)$$

The tree that minimizes wiring cost alone is the Steiner tree (figure 1a), which, unlike a minimum spanning tree, allows for branch points that may help reduce the total length of the arbor. There are efficient algorithms to compute a minimum spanning tree, but finding the optimal Steiner tree for a given set of points in three-dimensional space is NP-hard [42]. The tree that minimizes the conduction delay alone is the ‘Satellite’ tree, where each synapse is connected

via a straight line to the root of the tree (figure 1b). We compare trees built on the same set of input points, and thus we do not need to normalize either objective by the number of synapses, n .

(b) Defining Pareto optimal trees

It is generally impossible to find a tree that simultaneously minimizes both wiring cost and conduction delay because such a tree may not exist. One way to define the concept of an ‘optimal tree’ then is using the theory of Pareto optimality. Intuitively, a tree is Pareto optimal if no other tree on the same set of points has a lower value for both objectives, or a lower value for one objective without hurting the other objective. Formally, we say a tree T_1 *partially dominates* a tree T_2 (denoted $T_1 \preceq T_2$) if at least one of the two conditions hold:

- (1) $W(T_1) \leq W(T_2)$ and $D(T_1) < D(T_2)$; or
- (2) $W(T_1) < W(T_2)$ and $D(T_1) \leq D(T_2)$

This means that T_1 performs equal to or better than T_2 no matter how one prioritizes the two objectives. We say a tree T is *Pareto optimal* if it is not partially dominated by any other tree on the same set of points [43].

To generate Pareto optimal trees, we propose a simple linear interpolation between the two objectives, where the goal is to minimize the *Pareto cost*:

$$\min_{T_{\mathcal{P}}} \alpha W(T_{\mathcal{P}}) + (1 - \alpha) D(T_{\mathcal{P}}), \quad (2.3)$$

where $T_{\mathcal{P}}$ is a tree that connects the cell body (r) to the synapses (p_1, p_2, \dots, p_n) . Da & Cunha [44] prove that finding a tree that minimizes equation (2.3) is equivalent to finding a Pareto optimal tree. Here, α denotes how much weight is placed on one objective versus the other.

To generate the *Pareto front* (figure 1b), we need to find the set of trees that minimize equation (2.3) for each value of $\alpha \in [0, 1]$. If $\alpha = 0$, the optimal tree is the Satellite tree. If $\alpha = 1$, the optimal tree is the minimum Steiner tree. We developed an algorithm to generate optimal trees for intermediate values of α , as described below.

One advantage of this formulation is that each neural arbor can be associated with global arbor descriptor (its value α) that characterizes the trade-off between the two objectives that the arbor employs. We can then compare the distribution of these α values across arbors of different types to quantify morphological differences.

(c) An algorithm for generating Pareto optimal trees

To generate trees that lie on the Pareto front, we used a greedy algorithm that allows for branch points, which were not considered by prior work [7,8]. The algorithm takes as input \mathcal{P} and α and outputs $T_{\mathcal{P}}$ that attempts to minimize equation (2.3).

The algorithm starts with the root r in the tree, and all other points (nodes) outside the tree. In each step, the algorithm considers all possible edges between a node inside the tree and a node (synapse) outside the tree, and picks the edge that minimally increases the objective in equation (2.3). When the algorithm adds an edge, it adds k Steiner nodes equidistant along the edge, which may be used in subsequent steps as branch points. This process continues until

all synapses have been added to the tree. In experiments, we set $k = 10$.

Electronic supplementary material, figure S1 illustrates five steps of this process. Importantly, this algorithm is centralized and is designed to generate a near-optimal Pareto front that can be used to evaluate the optimality of neural arbors. This algorithm is not meant to mimic the distributed process by which neural arbors grow. We do not derive approximation bounds for this algorithm; however, in Results, we show that this algorithm performs well in practice.

Optimizing the algorithm. For arbors with thousands or tens of thousands of synapses this algorithm may not be very efficient because of the large number of possible edges to consider in each greedy step. Specifically, there are $\mathcal{O}((kn)^2)$ edges to consider, where n is the number of unconnected points and k is the number of Steiner points that we add along each edge. Here, we show that we can reduce the number of candidate edges to consider in each step to $\mathcal{O}(kn)$ without affecting the quality of the solution generated by the algorithm.

For each node $u \in V_{\mathcal{P}}$ already added to the tree, define:

$$\text{Close}(u, T_{\mathcal{P}}) = \underset{v}{\operatorname{argmin}} \{l(u, v) : v \in \mathcal{P} \setminus V_{\mathcal{P}}\},$$

i.e. the closest point to u that has not already been added to the tree. Then, define:

$$\text{Cost}(u, v, T_{\mathcal{P}}, \alpha) = \alpha(W(T_{\mathcal{P}}) + l(u, v)) + (1 - \alpha)(D(T_{\mathcal{P}}) + l(u, v) + d(u, r)),$$

i.e. the value of the objective, as defined by equation (2.3), that would result from adding edge (u, v) to $T_{\mathcal{P}}$.

Lemma 2.1 Fix $u \in V_{\mathcal{P}}$; then $\text{Cost}(u, v, T_{\mathcal{P}}, \alpha)$ is minimized by picking $v = \text{Close}(u, T_{\mathcal{P}})$ over all possible choice of v (nodes not already added to the tree).

Proof. Suppose $u \in V_{\mathcal{P}}$, $v \notin V_{\mathcal{P}}$ and we add edge (u, v) to $T_{\mathcal{P}}$. The increase in wiring cost is $\alpha \times l(u, v)$. By definition, this is minimized if $v = \text{Close}(u, T_{\mathcal{P}})$. When we connect v to u , the shortest path from v to r involves going from v to u and then taking the shortest path from u to r . Thus, the increase in conduction delay is $(1 - \alpha) \times d(v, r) = (1 - \alpha)(l(u, v) + d(u, r))$. Because $d(u, r)$ is unchanged by connecting v to u , the increase in conduction delay is minimized by minimizing $(1 - \alpha) \times l(u, v)$, which is minimized if $v = \text{Close}(u, T_{\mathcal{P}})$. Thus, both objective functions are minimized by connecting u to $\text{Close}(u, T_{\mathcal{P}})$. \square

Thus, at every step, we do not need to consider all possible edges between a node inside the tree and a node outside the tree. The optimal greedy step only involves edges of the form $(u, \text{Close}(u, T_{\mathcal{P}}))$, with $u \in V_{\mathcal{P}}$. The full algorithm is shown in algorithm 1 (electronic supplementary material).

To generate the Pareto front we apply the greedy algorithm to values of $\alpha \in \{0.00, 0.01, \dots, 0.99, 1.00\}$. Examples of arbors generated by the algorithm for different values of α are shown in figure 1c–h.

It remains unclear how close this algorithm gets to finding truly Pareto optimal trees. Minimizing equation (2.3) is NP-hard; however, some special cases are informative to look at in detail. For $\alpha = 0$, the algorithm will provably generate the optimal tree (the Satellite tree, which minimizes

conduction delay). For $\alpha = 1$, the algorithm will generate a tree with a wiring cost no higher than the wiring cost of the minimum spanning tree, which is itself provably within a factor of 2 of the wiring cost of the optimal Steiner tree [45]. In practice, this gap is often much tighter (see e.g. Conn *et al.* [46], where a similar greedy algorithm was shown to find trees within 5–6% of the Steiner tree for small sets of input points). To further evaluate the optimality of this algorithm, we compare it against a brute-force algorithm [47] for point sets that are small enough to generate the optimal spanning tree, and two other greedy heuristics [35,37] on larger point sets.

3. Results

First, we demonstrate that our greedy algorithm outperforms other algorithms at generating near Pareto optimal topologies. Second, we show that neural arbors lie much closer to the Pareto front than expected by three baselines, and that this result is robust to different assumptions about the locations of the synapses along the arbor. Third, we show that some functional categories of arbors can be differentiated based on how they are structured. Fourth, we use this same network design framework to demonstrate that plant arbors are also Pareto optimal.

(a) The greedy algorithm effectively approximates the Pareto front

As mentioned above, finding Pareto optimal arbors (i.e. minimizing the equation (2.3)) is NP-hard, forcing us to consider heuristics for generating the Pareto front for large arbors. We first determined how close to Pareto optimal arbors generated by our greedy algorithm were compared to the following three competitor algorithms:

- (1) *Brute-force algorithm* [47]. This algorithm simply enumerates all possible spanning trees and selects the one that minimizes equation (2.3) for a given value of α . There are n^{n-2} possible spanning trees, where n is the number of input points. Thus, we only consider the brute-force algorithm for small point sets (up to eight points).
- (2) *Khuller's light approximate shortest-path tree (LAST) algorithm* [35]. See the electronic supplementary material for details of this algorithm.
- (3) *Karger's algorithm* [37]. This algorithm is equivalent to our greedy algorithm without the use of Steiner points. Like our algorithm, Karger's algorithm has been used in geometric settings [37], and has been used for studying neural arbors [7,8].

None of these three algorithms add branch (Steiner) points. The comparison between our greedy algorithm and these three algorithms thus also highlights the utility of branch points in designing Pareto optimal topologies.

For the test framework, we selected a set of $n + 1$ points (\mathcal{P}) randomly from the space $[-10, 10]^3$, and one of these points was selected uniformly at random and designated to be the root (r). For each point set, we varied $\alpha \in (0, 1)$ in step sizes of 0.01 and ran each algorithm above to generate a different Pareto front. We then compared the three Pareto

fronts generated by the algorithms by comparing how many trees from each front were partially dominated (electronic supplementary material, Methods) by trees from one of the other fronts.

Small point sets. We first examined 411 point sets, each with between five and eight points. For each point set, we generated a Pareto front (99 trees, using different values of $\alpha \in [0, 1]$) using the brute-force algorithm, Khuller's algorithm, Karger's algorithm and our greedy algorithm. We computed a Pareto front using each algorithm; we then compared the quality of these Pareto fronts to determine which algorithm generated the Pareto front that was closest to optimal (electronic supplementary material, Methods).

We find that only 7% of the trees generated by our greedy algorithm are dominated by some tree generated by one of the other two algorithms. This is in contrast to 52% for the brute-force algorithm, 56% for Karger's algorithm and 77% for Khuller's algorithm. The brute-force algorithm cannot be dominated by either Karger's or Khuller's algorithms because neither of these algorithms allow branch points. Thus, the brute-force algorithm is actually dominated by the greedy algorithm 52% of the time. The improved performance of the greedy algorithm over brute-force highlights the value of using branch points in finding Pareto optimal arbors.

Larger point sets. Next, we examined 1632 point sets, each with 9–500 points. This size is too large for the brute-force algorithm to run in a reasonable amount of time.

We find that only 13% of the trees generated by our greedy algorithm are dominated by some tree generated by one of the other two algorithms. This is in contrast to 82% for Karger's algorithm, and 99.8% for Khuller's algorithm. Thus, the greedy algorithm often generates a more optimal Pareto front compared to Khuller's and Karger's algorithms, which were previously used for this problem [7,8], and its use of Steiner points allows for more biological realism.

(b) Neural arbors are Pareto optimal

Next, we used the greedy algorithm to test how close neural arbors were to being Pareto optimal. We obtained 14 145 3D arbor reconstructions from Neuromorpho (electronic supplementary material, Methods). For each arbor, the location of the root (r) was designated in the data, and the locations of the synapses (the p_i) were assumed to lie uniformly along the traced arbor (electronic supplementary material, Methods). For each arbor point set, we applied the greedy algorithm for values of $\alpha \in [0, 1]$ to generate the Pareto front. We then evaluated the wiring cost and conduction delay for the actual neural arbor, and the distance of the arbor to the Pareto front (figure S2; electronic supplementary material, Methods). We also determined the wiring cost and conduction delay for arbors generated by three baseline algorithms—Random, Centroid and Barabási–Albert (electronic supplementary material, Methods)—for the same point set to assess how likely it is for the arbor to fall near the Pareto front by chance. These three baselines generate a tree $T_{\mathcal{P}}$ for a set of points \mathcal{P} . The Centroid tree creates a Steiner point at the centroid of the points in \mathcal{P} and connects each point in \mathcal{P} to this centroid. Unlike the Centroid tree, Random and Barabási–Albert do not attempt to produce a tree with any geometric properties. We did not consider other geometric models to assess whether the optimizations made by neural arbors could also be realized by non-geometric

models by chance. For example, if in a model a node was simply connected to its nearest neighbour, this would implicitly optimize for wiring length.

Strikingly, we found that almost all neural arbors fell very close to the Pareto front. In figure 2*a–d*, we show example Pareto plots for four individual arbors. The Pareto front is shown in blue circles, with one circle per value of $\alpha \in \{0, 0.01, \dots, 0.99, 1.00\}$. We also mark the location of the neural arbor with a red 'X', and the locations of the trees generated by each baseline method using the same input point set.

Neural arbors fell much closer to the Pareto front than the three baselines. In figure 2*e*, we plot the distances from the neural arbor to the Pareto front over all 14 145 point sets. In this plot, the y -axis is the log of the distance from the arbor to the Pareto front (lower means closer to optimal), and the x -axis denotes the arbor number (ranging from 1 to 14 145), ordered by distance. Of the 14 145 point sets, 92.21% of the neural arbors lay closer to the Pareto front than the Centroid tree; 99.99% lay closer than the Barabási–Albert tree; and 100% lay closer than the Random tree. Using a binomial test (electronic supplementary material, Methods), we find that the neural arbor is significantly more optimal than the three baselines ($p < 10^{-322}$ in all cases). These results are summarized in electronic supplementary material, table S2.

The results above suggest that nearly all neural arbors lie closer to the Pareto front than other baselines, but they do not describe how much closer. Over all point sets, the average distance ratio (distance to the Pareto front for the baseline divided by distance to the Pareto front for the neural arbor) was 1.38 ± 0.20 for the Centroid tree, 5.96 ± 2.36 for the Barabási–Albert tree and 30.22 ± 20.06 for the Random tree. In other words, on average, the neural arbors were at least 38% lower for wiring length, or at least 38% for conduction delay, compared to the Centroid tree; at least 496% lower than the average Barabási–Albert tree; and at least 2922% lower than the average random spanning. Using a T-test (electronic supplementary material, Methods), we find that the neural arbors lie significantly closer to the Pareto front than all baselines ($p < 10^{-324}$), and thus achieve better trade-offs between wiring cost and conduction delay than expected by chance.

These results were derived using neural arbors where the synapse locations were assumed to lie uniformly along the arbor. In the electronic supplementary material, Results, we analyse arbors with different spacing between synapses (electronic supplementary material, figure S3), and with known synapse locations (electronic supplementary material, figure S4), and find similar results.

(c) Classifying neural arbors by their structure

Are functional differences between arbors reflected in their structure? Here, we study two questions. First, are arbors from different biological categories equally close to being Pareto optimal? Second, do arbors from different biological categories have significantly different values of α (i.e. do they weigh wiring cost and conduction delay differently)? Uncovering such relationships could be used to hypothesize about unknown function given structure.

We first associated each arbor with a biological category based on the type of arbor (apical dendrite, basal dendrite or axon), the cell type of the neuron (e.g. Purkinje versus granule), or the neurotransmitter type used by the cell

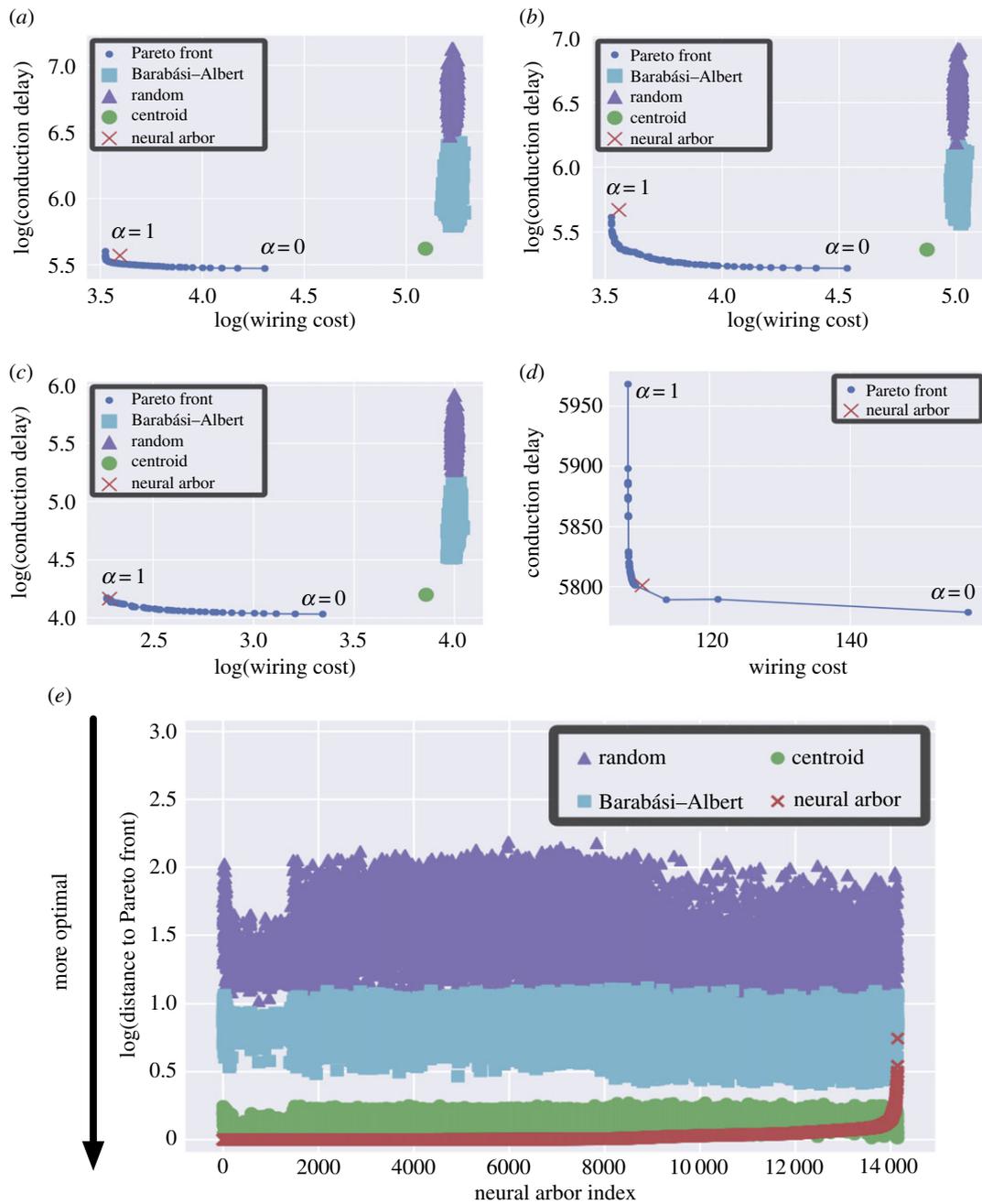


Figure 2. Neural arbors are Pareto optimal. (a–d) Example Pareto front analysis for four arbors: (a) sensory receptor cell from the mouse neocortex, (b) sensory receptor cell from the mouse dorsal root ganglion, (c) amacrine cell from the human retina and (d) sensory receptor neuron from the mouse peripheral nervous system. In each panel, the blue curve shows the Pareto front generated using the greedy algorithm. The red 'X' shows the location of the neural arbor, the green circle shows the location of the Centroid tree, the purple triangles show the locations of the uniform random spanning trees (Random), and the aqua squares show the locations of the Barabási–Albert trees. The x-axis is the log wiring length, and the y-axis is the log conduction delay. Neural arbors lie much closer to the Pareto front compared to the other three baselines. In (d), the arbor is best explained by a value $\alpha = 0.03$ (right-end of the Pareto front). To make its Pareto front clear, we omit the baseline models and we do not use a log scale on either axis. (e) Summary of all 14 145 arbors. For each arbor (x-axis), we plot the distance to the Pareto front for the neural arbor, the Centroid tree, the average Random tree, and the average Barabási–Albert tree. The x-axis is sorted according to the neural arbor's distance to the Pareto front. Neural arbors tend to fall significantly closer to the Pareto front than the other baselines.

(e.g. glutamatergic versus GABAergic). See electronic supplementary material, table S1 for full list of categories. We then associated each arbor with two values: its distance to the Pareto front (electronic supplementary material, Methods, equation (2)), and a value of $\alpha \in [0, 1]$ that comes closest to matching its topology (electronic supplementary material, Methods, equation (3)). Below, we ask if there are systematic differences in these two values for arbors in one category versus another.

Arbor type. The distance to the Pareto front was smaller for apical dendrites (1.01 ± 0.10) and basal dendrites

(1.01 ± 0.02) compared to axons (1.13 ± 0.14) (figure 3a). Error (\pm) values represent standard deviations. Using Welch's T-test on the null hypothesis that the average value of this distance is the same, this difference was significant ($p < 10^{-324}$). One possibility for why axons may lie further from the Pareto front than dendrites is that there is a potential third objective, such as space-filling, that axons also need to prioritize (Discussion).

Axons have higher values of α than dendrites: 0.86 ± 0.18 for axons versus 0.26 ± 0.30 for apical dendrites, and 0.52 ± 0.30 for basal dendrites (figure 3b). Using Welch's T-test on

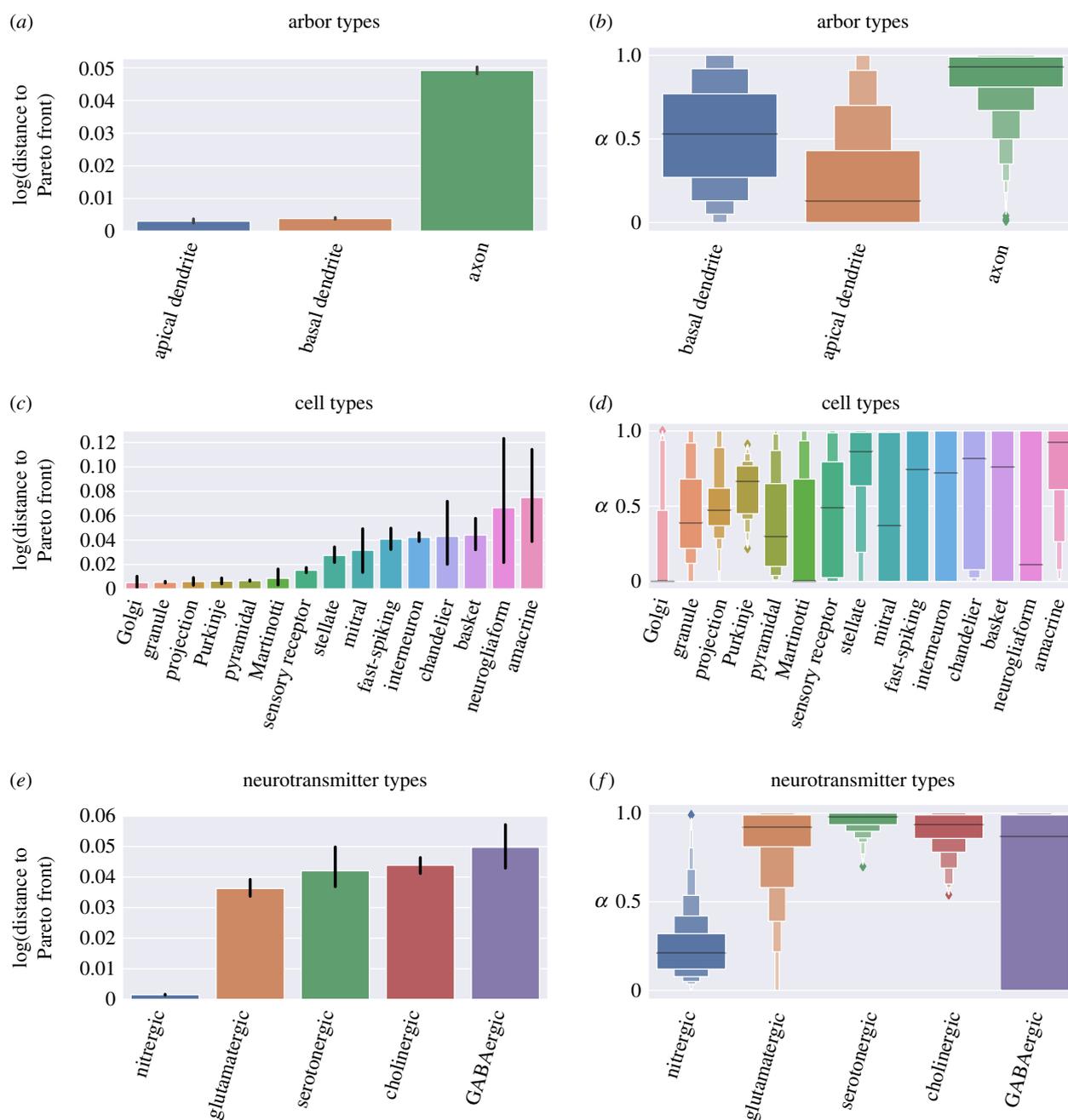


Figure 3. Structure–function differences in neural arbors. (a,b) We categorize arbors in three categories. In (a), we plot the distribution of the distances (y-axis) to the Pareto front for arbors in each category (x-axis). In (b), we show a letter plot depicting the distribution of α values for arbors in each category. The middle bar of the letter plot shows the median of the distribution, and the two innermost boxes show the 25th and 75th percentiles, respectively; the next two boxes show the 12.5th and 87.5th percentiles, respectively, and so on. Diamonds show individual outliers. (c,d) Same analysis with arbors grouped by cell type. (e,f) Same analysis with arbors grouped by neurotransmitter type.

the null hypothesis that the average value of α is the same for every pair of groups, we find that each pair of groups (axons versus apical dendrites, axons versus basal dendrites, and apical dendrites versus basal dendrites) are significantly different from each other ($p < 10^{-324}$ in all cases). Thus, apical dendrites appear to prioritize conduction delay over wiring cost; basal dendrites take on intermediate values of α with a more even weighting of both objectives; and axons appear to prioritize wiring cost over conduction delay.

Cell type. Here, we studied differences in arbors belonging to one of 15 well-studied cell types (figure 3c,d; electronic supplementary material, table S1).

The cell types closest to the Pareto front were Golgi cells (1.01 ± 0.03), granule cells (1.01 ± 0.04) and projection cells (1.01 ± 0.04) (figure 3d). By contrast, the cells that lie furthest

to the Pareto front included amacrine cells (1.29 ± 0.65) and neurogliaform cells (1.28 ± 0.86) (figure 3c). The cell types with the most extreme α values (i.e. close to 0 or 1) were Golgi cells (0.13 ± 0.32), Martinotti cells (0.30 ± 0.41), amacrine cells (0.75 ± 0.32) and stellate cells (0.74 ± 0.33).

While neuroanatomists have long appreciated the visual diversity of arbors from different neural cell types, our work quantifies and provides a quantitative signature for some of these differences based on a simple trade-off principle. Potential avenues for further work may involve comparing more detailed aspects of these cell types, such as dendritic signalling, to assess why different types of arbors may prioritize one objective over the other.

Neurotransmitter types. Here, we grouped arbors into one of five classes of neurotransmitter types that the

corresponding cell releases (figure 3e,f; electronic supplementary material, table S1).

Nitric neurons lay closest to the Pareto front (1.003 ± 0.003) and had significantly smaller values of α (0.24 ± 0.17), indicating a tendency towards optimizing for conduction delay over wiring cost. Other neurotransmitter types lay further from the Pareto front: glutamatergic (1.09 ± 0.07), serotonergic (1.11 ± 0.13), cholinergic (1.11 ± 0.05) and GABAergic neurons (1.14 ± 0.22) (figure 3e). GABAergic neurons tended to exhibit intermediate α values (0.65 ± 0.41), whereas the other neurotransmitters generally had values of $\alpha \geq 0.84$ —glutamatergic (0.84 ± 0.22), cholinergic (0.90 ± 0.11) and serotonergic (0.96 ± 0.06). Thus, nitric neurons are unique topologically: they are significantly closer to being Pareto optimal than any of the other neurotransmitters, and they prioritize minimizing conduction delay much more strongly than other neurotransmitters. These results are summarized in electronic supplementary material, table S3.

Systematic clustering of α values for arbors of different neurotransmitter types shows the value in using the Pareto front location to classify neural arbors by function, and may highlight new connections between structure and function that merit deeper investigation.

We also found a positive correlation between an arbor's distance to the Pareto front and the arbor's α value ($R = 0.42$ using Pearson's correlation coefficient; $p < 10^{-324}$ using a permutation test). In other words, the more strongly a neuron emphasized minimizing wiring cost, the further it lay from the Pareto front. This correlation is positive within every category of arbors we studied (electronic supplementary material, table S1), and the correlation is significantly different from 0 for every category ($p < 0.005$ in all cases using a permutation test).

These results show that certain categories of arbors can be classified using a single value (α), indicating how the arbor trades off between wiring length and conduction delay. Additional topological descriptors may further help classify these arbors [18]. Our results suggest that during evolution, natural selection may fine-tune arbor growth strategies based on functional requirements, while still following basic design templates that generate Pareto optimal arbors.

(d) Similarities in branching patterns of neural arbors and plant architectures

Nature is abound with branching structures, and it is natural to ask whether other such networks exhibit a similar cost-versus-performance trade-off.

Conn *et al.* [46] recently identified wiring cost and conduction delay as important design principles constraining plant shoot (above ground) architectures. Here, synapses are analogous to leaves, and the soma is analogous to the base/root of the plant. Wiring cost for plants corresponds to the amount of resources (e.g. carbon) used to build the architecture. Conduction delay corresponds to a measure of the nutrient transport efficiency (e.g. sugars, water) from the leaves to the root system, and vice versa.

Re-analysing their data, we find that while neural arbors tend to prioritize minimizing wiring cost, plant shoot architectures tend to prioritize minimizing conduction delay (electronic supplementary material, Results and figure S5). We speculate that for plants, space may not be as constrained as for neural arbors packed in the brain, and thus it is possible

that wiring conservation is more important to neurons than plants. Nonetheless, the fact that plants and neurons follow similar design principles was an unexpected result, especially since they are separated by millions of years of evolution.

4. Discussion

We studied a network design trade-off between two competing biological objective functions—minimizing conduction delay and minimizing wiring cost—interpolated between by a single parameter. We developed a graph-theoretic algorithm to generate near-Pareto optimal topologies that outperformed prior algorithms. We then showed that almost all 14 145 neural arbors analysed achieved a level of Pareto optimality significantly better than three other baseline arbors, suggesting that such an optimization is unlikely to occur by chance. We presented instances where differences in arbor type, cell type and neurotransmitter type employed different trade-offs, suggesting that different biological constraints can tune arbor topology while still obeying the same underlying trade-off principle. Finally, we compared neural arbors to plant architectures and found that both were Pareto optimal, suggesting broad similarities in branching structures across natural systems that have diverged long ago.

Our work raises three questions for future work. First, our algorithm for computing the Pareto front was designed to find nearly Pareto optimal trees; it was not meant to mimic the growth process used by neural arbors. Specifically, our algorithm used centralized computation when determining which edge to add next to the existing tree. Furthermore, it assumed that the locations of pre- or post-synaptic partners were pre-determined. In reality, arbors likely grow using local rules of computation, and create synapses in locations that are not entirely pre-determined. This raises an open challenge of finding a distributed stochastic growth algorithm to generate Pareto optimal arbors, while incorporating general rules by which neural arbors develop, including the option to add branch points, and allowing some flexibility in where synapses are made. Such an algorithm may involve modifying existing random graph models to produce desirable geometric structures [48]. Second, as discussed in related work, there are several other objectives that may constrain neural arbor topology, such as space-filling, that may help further classify arbor types and better explain instances where arbors lie further from the Pareto front. One advantage of our Pareto optimality framework is that it can naturally be extended to more than two objectives. Furthermore, our model did not include wire radii, as these data are not currently widely available for many different types of arbors. These data could also be introduced in our framework using weighted edges. Third, further improvements could be made to our greedy algorithm. For example, instead of only considering one edge to add per iteration, we could add multiple edges per time step. The algorithm would choose the optimal set of k new edges in each time-step; however, such a k -greedy algorithm would increase the run-time by a factor of $O(n^{k+1})$ steps.

Data accessibility. The code used in the study is available at: <https://github.com/arjunc12/neurons>. Tracings of neural arbors are available at: <http://neuromorpho.org/>.

Competing interests. We declare we have no competing interests.

Funding. A.C. was supported by the Chapman Foundations Management, LLC. S.N. was supported by the Pew Charitable Trusts, the

References

- Laughlin SB, Sejnowski TJ. 2003 Communication in neuronal networks. *Science* **301**, 1870–1874. (doi:10.1126/science.1089662)
- Wen Q, Chklovskii DB. 2008 A cost–benefit analysis of neuronal morphology. *J. Neurophysiol.* **99**, 2320–2328. (doi:10.1152/jn.00280.2007)
- Rivera-Alba M, Peng H, de Polavieja GG, Chklovskii DB. 2014 Wiring economy can account for cell body placement across species and brain areas. *Curr. Biol.* **24**, R109–R110. (doi:10.1016/j.cub.2013.12.012)
- Chen BL, Hall DH, Chklovskii DB. 2006 Wiring optimization can relate neuronal structure and function. *Proc. Natl Acad. Sci. USA* **103**, 4723–4728. (doi:10.1073/pnas.0506806103)
- Rivera-Alba M, Vitaladevuni SN, Mishchenko Y, Lu Z, Takemura S-Y, Scheffer L, Meinertzhagen IA, Chklovskii DB, de Polavieja GG. 2011 Wiring economy and volume exclusion determine neuronal placement in the *Drosophila* brain. *Curr. Biol.* **21**, 2000–2005. (doi:10.1016/j.cub.2011.10.022)
- Wang IE, Clandinin TR. 2016 The influence of wiring economy on nervous system evolution. *Curr. Biol.* **26**, R1101–R1108. (doi:10.1016/j.cub.2016.08.053)
- Cuntz H, Forstner F, Borst A, Häusser M. 2010 One rule to grow them all: a general theory of neuronal branching and its practical application. *PLoS Comput. Biol.* **6**, e1000877. (doi:10.1371/journal.pcbi.1000877)
- Budd JML, Kovács K, Ferecskó AS, Buzás P, Eysel UT, Kisvárdy ZF. 2010 Neocortical axon arbors trade-off material and conduction delay conservation. *PLoS Comput. Biol.* **6**, e1000711. (doi:10.1371/journal.pcbi.1000711)
- Chklovskii DB. 2004 Synaptic connectivity and neuronal morphology: two sides of the same coin. *Neuron* **43**, 609–617. (doi:10.1016/s0896-6273(04)00498-2)
- Ramon Y, Cajal S. 1904 *Textura del Sistema Nervioso del Hombre y de los Vertebrados*. Madrid, Spain: Nicolas Moya.
- Vetter P, Roth A, Häusser M. 2001 Propagation of action potentials in dendrites depends on dendritic morphology. *J. Neurophysiol.* **85**, 926–937. (doi:10.1152/jn.2001.85.2.926)
- Gasparini S, Migliore M. 2015 Action potential backpropagation. In *Encyclopedia of computational neuroscience* (eds D Jaeger, R Jung), pp. 133–137. New York, NY: Springer.
- Bekkers JM, Häusser M. 2007 Targeted dendrotomy reveals active and passive contributions of the dendritic tree to synaptic integration and neuronal output. *Proc. Natl Acad. Sci. USA* **104**, 11 447–11 452. (doi:10.1073/pnas.0701586104)
- Prömel HJ, Steger A. 2012 *The Steiner tree problem: a tour through graphs, algorithms, and complexity*. Brunswick, Germany: Vieweg+Teubner Verlag.
- Shoval O, Sheftel H, Shinar G, Hart Y, Ramote O, Mayo A, Dekel E, Kavanagh K, Alon U. 2012 Evolutionary trade-offs, Pareto optimality, and the geometry of phenotype space. *Science* **336**, 1157–1160. (doi:10.1126/science.1217405)
- Ahn Y-Y, Jeong H, Kim BJ. 2006 Wiring cost in the organization of a biological neuronal network. *Physica A* **367**, 531–537. (doi:10.1016/j.physa.2005.12.013)
- Kim Y, Sinclair R, Chindapol N, Kaandorp JA, De Schutter E. 2012 Geometric theory predicts bifurcations in minimal wiring cost trees in biology are flat. *PLoS Comput. Biol.* **8**, e1002474. (doi:10.1371/journal.pcbi.1002474)
- Teeter CM, Stevens CF. 2011 A general principle of neural arbor branch density. *Curr. Biol.* **21**, 2105–2108. (doi:10.1016/j.cub.2011.11.013)
- Sugimura K, Shimono K, Uemura T, Mochizuki A. 2007 Self-organizing mechanism for development of space-filling neuronal dendrites. *PLoS Comput. Biol.* **3**, e212. (doi:10.1371/journal.pcbi.0030212)
- Scott EK, Luo L. 2001 How do dendrites take their shape? *Nat. Neurosci.* **4**, 359. (doi:10.1038/86006)
- Panico J, Sterling P. 1995 Retinal neurons and vessels are not fractal but space-filling. *J. Comp. Neurol.* **361**, 479–490. (doi:10.1002/(ISSN)1096-9861)
- Ćimović J, Mäki-Marttunen T, Linne M-L. 2015 The effects of neuron morphology on graph theoretic measures of network connectivity: the analysis of a two-level statistical model. *Front. Neuroanat.* **9**, 76. (doi:10.3389/fnana.2015.00076)
- Puppo F, George V, Silva GA. 2018 An optimized structure–function design principle underlies efficient signaling dynamics in neurons. *Sci. Rep.* **8**, 10460. (doi:10.1038/s41598-018-28527-2)
- Hodgkin AL, Huxley AF, Katz B. 1952 Measurement of current–voltage relations in the membrane of the giant axon of loligo. *J. Physiol.* **116**, 424–448. (doi:10.1113/jphysiol.1952.sp004716)
- Rall W. 1962 Theory of physiological properties of dendrites. *Ann. NY Acad. Sci.* **96**, 1071–1092. (doi:10.1111/j.1749-6632.1962.tb54120.x)
- Rall W. 1962 Electrophysiology of a dendritic neuron model. *Biophys. J.* **2**, 145–167. (doi:10.1016/S0006-3495(62)86953-7)
- Chklovskii DB, Stepanyants A. 2003 Power-law for axon diameters at branch point. *BMC Neurosci.* **4**, 18. (doi:10.1186/1471-2202-4-18)
- Rall W, Burke RE, Holmes WR, Jack JJ, Redman SJ, Segev I. 1992 Matching dendritic neuron models to experimental data. *Physiol. Rev.* **72**, S159–S186. (doi:10.1152/physrev.1992.72.suppl_4.s159)
- West GB, Brown JH, Enquist BJ. 1999 The fourth dimension of life: fractal geometry and allometric scaling of organisms. *Science* **284**, 1677–1679. (doi:10.1126/science.284.5420.1677)
- West GB, Brown JH, Enquist BJ. 1999 A general model for the structure and allometry of plant vascular systems. *Nature* **400**, 664–667. (doi:10.1038/23251)
- Banavar JR, Damuth J, Maritan A, Rinaldo A. 2002 Supply–demand balance and metabolic scaling. *Proc. Natl Acad. Sci. USA* **99**, 10 506–10 509. (doi:10.1073/pnas.162216899)
- Arora S. 1996 Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Foundations of Computer Science, 1996. Proc., 37th Annual Symposium on*, pp. 2–11. Piscataway, NJ: IEEE.
- Karpinski M, Zelikovskiy A. 1997 New approximation algorithms for the steiner tree problems. *J. Comb. Optim.* **1**, 47–65. (doi:10.1023/A:1009758919736)
- Agrawal A, Klein P, Ravi R. 1995 When trees collide: an approximation algorithm for the generalized steiner problem on networks. *SIAM J. Comput.* **24**, 440–456. (doi:10.1137/S0097539792236237)
- Khuller S, Raghavachari B, Young N. 1995 Balancing minimum spanning trees and shortest-path trees. *Algorithmica* **14**, 305–321. (doi:10.1007/BF01294129)
- Nguyen UT, Xu J. 2007 Multicast routing in wireless mesh networks: minimum cost trees or shortest path trees? *IEEE Commun. Mag.* **45**, 72–77. (doi:10.1109/MCOM.2007.4378324)
- Alpert CJ, Hu TC, Huang JH, Kahng AB, Karger D. 1995 Prim–Dijkstra tradeoffs for improved performance-driven routing tree design. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **14**, 890–896. (doi:10.1109/43.391737)
- Chen G, Chen S, Guo W, Chen H. 2007 The multi-criteria minimum spanning tree problem based genetic algorithm. *Inf. Sci.* **177**, 5050–5063. (doi:10.1016/j.ins.2007.06.005)
- Sourd F, Spanjaard O. 2008 A multiobjective branch-and-bound framework: application to the biobjective spanning tree problem. *INFORMS J. Comput.* **20**, 472–484. (doi:10.1287/ijoc.1070.0260)
- Rees CL, Moradi K, Ascoli GA. 2017 Weighing the evidence in Peters’ rule: does neuronal morphology predict connectivity? *Trends Neurosci.* **40**, 63–71. (doi:10.1016/j.tins.2016.11.007)
- Kasthuri N, et al. 2015 Saturated reconstruction of a volume of neocortex. *Cell* **162**, 648–661. (doi:10.1016/j.cell.2015.06.054)
- Garey MR, Graham RL, Johnson DS. 1977 The complexity of computing steiner minimal trees. *SIAM J. Appl. Math.* **32**, 835–859. (doi:10.1137/0132072)

43. Shoval O, Sheftel H, Shinar G, Hart Y, Ramote O, Mayo A, Dekel E, Kavanagh K, Alon U. 2012 Evolutionary trade-offs, pareto optimality, and the geometry of phenotype space. *Science* **386**, 1157–1160. (doi:10.1126/science.1217405)
44. Da Cunha N, Polak E. 1967 Constrained minimization under vector-valued criteria in finite dimensional spaces. *J. Math. Anal. Appl.* **19**, 103–124. (doi:10.1016/0022-247X(67)90025-X)
45. Kou L, Markowsky G, Berman L. 1981 A fast algorithm for Steiner trees. *Acta Inform.* **15**, 141–145. (doi:10.1007/BF00288961)
46. Conn A, Pedmale UV, Chory J, Navlakha S. 2017 High-resolution laser scanning reveals plant architectures that reflect universal network design principles. *Cell Syst.* **5**, 53–62. (doi:10.1016/j.cels.2017.06.017)
47. Gabow HN, Myers EW. 1978 Finding all spanning trees of directed and undirected graphs. *SIAM J. Comput.* **7**, 280–287. (doi:10.1137/0207024)
48. Sporns O. 2010 *Networks of the brain*. Cambridge, MA: MIT Press.