TOPOLOGICAL SHOLL DESCRIPTORS FOR NEURONAL CLUSTERING AND CLASSIFICATION

Reem Khalil [1,2,*], Sadok Kallel [3,4], Ahmad Farhat [5], Paweł Dłotko [5],

[1]American University of Sharjah, Departement of Biology Chemistry and Environmental Sciences, Sharjah, United Arab Emirates.
[2]City College of New York, New York, United States.
[3]American University of Sharjah, Department of Mathematics, Sharjah, United Arab Emirates.
[4]Laboratoire Painlevé, Université de Lille, France.
[5]Dioscuri Centre in Topological Data Analysis, Mathematical Institute, Polish Academy of Sciences, Warsaw, Poland.

Variations in neuronal morphology among cell classes, brain regions, and animal species are thought to underlie known heterogeneities in neuronal function. Thus, accurate quantitative descriptions and classification of large sets of neurons is essential for functional characterization. However, unbiased computational methods to classify groups of neurons are currently scarce. We introduce a novel, robust, and unbiased method to study neuronal morphologies. We develop mathematical descriptors that quantitatively characterize structural differences among neuronal cell types and thus classify them. Each descriptor that is assigned to a neuron is a function of a distance from the soma with values in real numbers or more general metric spaces. Standard clustering methods enhanced with detection and metric learning algorithms are then used to objectively cluster and classify neurons. Our results illustrate a practical and effective approach to the classification of diverse neuronal cell types, with the potential for discovery of putative subclasses of neurons.

Keywords: morphology, brain, cell types, cortex, metric learning, machine learning, persistence diagrams

1. Introduction

Neuronal morphology dictates how information is processed within neurons [1], as well as how neurons communicate within networks [2]. Thus, given the large diversity in dendritic morphology within and across cell classes, quantifying variations in morphology becomes fundamental to elucidate neuronal function. The two major classes of neurons in the neocortex are principal cells (pyramidal cells), and GABAergic interneurons. Pyramidal cells play a critical role in circuit structure and function, and are the most abundant type in the cerebral cortex (70-80% of the total neuronal population) [3]. The morphology of pyramidal cells can vary substantially among cortical areas within a species [4, 5, 6, 7], and across species [8, 9]. Similarly, neocortical GABAergic interneurons are important in shaping cortical circuits, accounting for 10-30% of all cortical neurons [10, 11]. Classification of GABAergic interneurons has proved to be especially challenging due to their diverse morphological, electrophysiological, and molecular properties [12, 13]. Importantly, morphological differences among classes and subclasses of pyramidal cells and interneurons are presumed to be functionally relevant. Moreover, changes in neuronal morphology is thought to underlie various neurodevelopmental [14], and acquired [15, 16, 17, 18, 19] disorders. Thus, given the key role of pyramidal cells and interneurons in cortical function in health and disease, it is important to differentiate among their subclasses through rigorous classification tools.

Standard approaches rely on measurements of morphological features typically acquired from digital neuron reconstructions. Feature measurements are subsequently used to quantitatively assess and cluster cell classes [20], using standard supervised [13], and unsupervised [21, 22, 23] clustering algorithms. The raw quantification of features, provided by standard methods often fails to discriminate among neuronal classes that are visually very different (§2). Additionally, these methods have only been tested on select datasets. Therefore, there is a demand for more robust and general methods for discriminating among diverse neuronal cell types and larger datasets. In recent years, the field of computational topology has become increasingly more popular in the characterization of tree structures, including neurons. For example, [24] developed a new algorithm called 'Topological morphological descriptor', TMD which is based on topological data analysis (i.e. persistence diagrams) to classify families of neurons. In a more recent study by [25], the authors use TMD to classify cortical pyramidal cells in rat somatosensory cortex. The

topological classification was largely in agreement with previously published expert assigned cell types. Furthermore, [26] present a framework based on persistence homology to compare and classify groups of neurons. Nevertheless, the available methods fail to fully capture the subtle morphological differences among families of neurons as we illustrate in the present work.

Here we take a novel approach to the classification of neurons. We view a morphological feature describing a neuron as a function which takes values in the real numbers, or more generally in some relevant metric space, and varies as a function of distance from the soma. Each morphological feature, such as tortuosity, taper rate, branching pattern, etc., gives rise to what we refer to as a *Sholl descriptor*. This *Sholl descriptor* is a rule that assigns a metric element, such as a number or a persistence diagram, to every given neuron and at any given distance from its soma. Value assignments are normalized so that all neurons are represented on a comparable scale, and are isometry invariant and stable §4.10. The construction we just outlined endows the set of neurons with a *descriptor metric* for every Sholl descriptor. Our approach is useful in that every morphological feature turns the set of neurons into a metric space. The closer the neurons are in the underlying descriptor metric, the more of this feature they share. This method gives a powerful, objective, and interpretable tool to compare and analyze neuronal morphologies.

In the course of our work, we developed eight Sholl descriptors, representing eight features which we then use in both unsupervised and supervised settings to cluster and classify families of neurons. Using diverse datasets, we identify key descriptors that reveal differences and similarities between neuronal classes. Our discrimination results were significantly better in separating different neuronal cell types than clustering methods based on raw quantification of features. Certain descriptors result in complete separation of selected groups of neurons. Thus, our highly effective and powerful classification tool could be used for the identification of new neuronal cell types, ultimately enhancing our understanding of the morphological diversity and function of neurons in the brain.

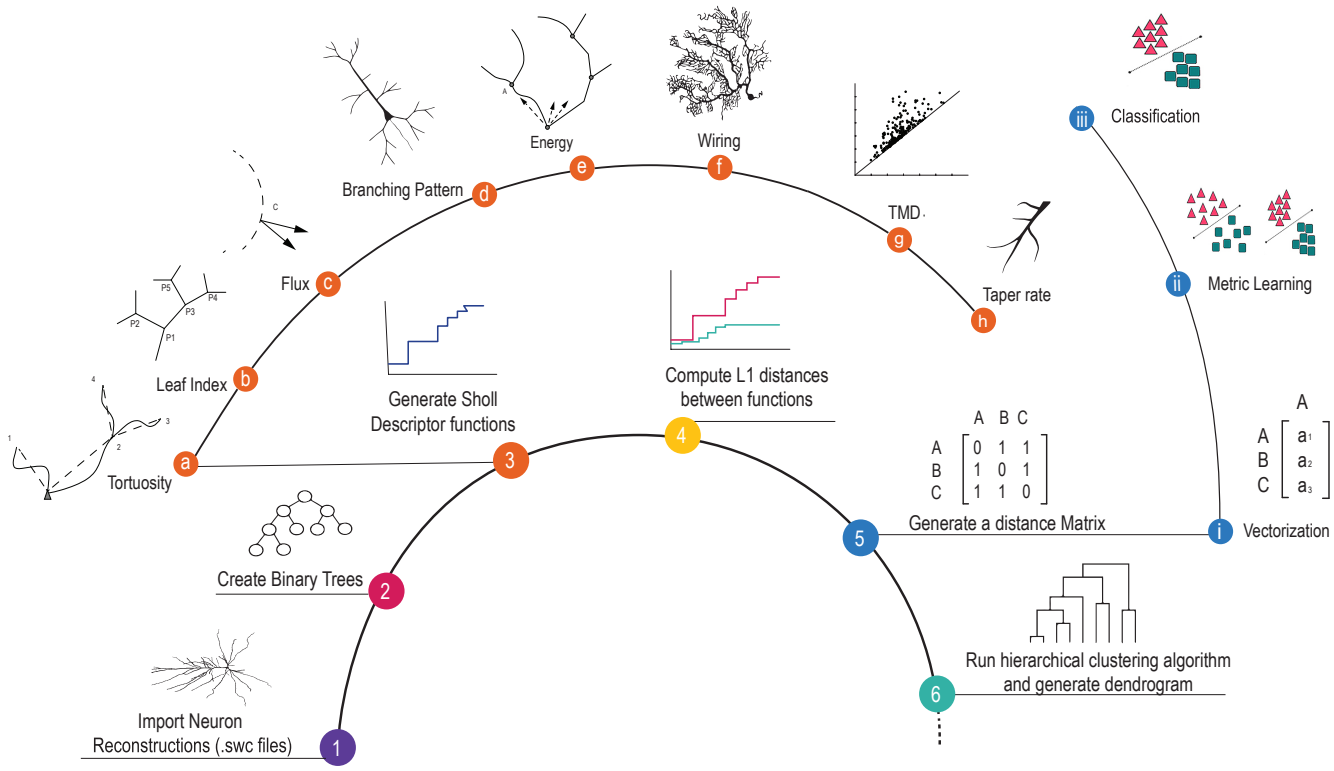

Fig. 1. A pipeline to generate and cluster morphological data. Reconstructions of neuronal cell types are encoded in Sholl descriptors which are then used for clustering and classification. A representation of each Sholl descriptor is depicted in steps **a-h**. The process of clustering different neuronal cell types is shown in steps 1-6. Once neurons are vectorized based on descriptor metrics, we apply metric learning techniques to obtain classification (steps **i-iii**).

2. Results

We developed eight descriptors based on the following morphological or morphometric features: branching pattern, tortuosity, taper rate, wiring, flux, leaf index, energy, and the parameterized TMD descriptor. A schematic of each descriptor is illustrated in steps a-h in Fig. 1, and all terms are defined in §4.3. We illustrated the discriminative accuracy of our Sholl descriptors on six different datasets, providing evidence that relevant Sholl descriptors can reliably discriminate among different classes of neurons in agreement with previously published assignment. All datasets were downloaded from Neuromorpho.org [27]. They were chosen to cover diverse types and subtypes of neurons across different regions and animal species. For each dataset, we predefined classes and subsequently performed clustering analysis in several different ways.

Detection, clustering and classification methods. In this paper we present a toolkit of descriptors, and describe their method of implementation. Sholl descriptors are run one at a time on each dataset, detection rates are computed, and dendrograms based on cluster analysis are generated. Descriptors can then be combined to optimize clustering and obtain classification.

To streamline terminology, a class C is detected at a 100% level by a descriptor ϕ means that the entire class fits in a ball under the descriptor metric, and no other neuron from any other class is within that ball. Detection reveals how much a given feature is able to single out the class C from all other classes. Detection rates are correlated with the corresponding dendrogram obtained via cluster analysis, so a high detection rate leads to the neurons in the class clustering together in the accompanying dendrogram. For convenience, we say a class has been *detected* by a descriptor ϕ if the rate of detection for that class is at least 90% (details in §4.7).

Neuronal clustering can be achieved using a single descriptor or a combination of descriptors. We present three different ways of combining the descriptors, each with its own merits. The first combination method is unsupervised in that it does not depend on the way we subdivide our dataset into classes §4.13. This combination method is used to analyze dataset 1. The second method combines features through a grid-search algorithm that produces a linear combination of descriptor metrics capable of differentiating among classes §4.14. If no such combination can be produced, the classes are indistinguishable under our morphological descriptors. This method is used to analyze dataset 5. The third combination method is a supervised technique which is achieved by means of metric learning and yields neuronal classification when applied to a large dataset. This last method is used to analyze dataset 6. This method works by first selecting relevant features based on their detection rates. Neurons are then vectorized using the descriptor metrics. A metric learning algorithm is then applied and produces an optimal metric that differentiates the classes. This method is subsequently checked for 'overfitting' §4.16. The new metric is subsequently used to tell how close in features a random neuron is to the given classes.

Feature selection and clustering: L-measure versus Sholl descriptors. As a proof of concept, we implemented our Sholl descriptors on dataset 1 which comprised three different neuronal cell types in the mouse brain: retinal ganglion cells (n=10), cerebellar purkinje cells (n=9), and interneurons in the medial prefrontal cortex (n=10). The assumption was to choose strikingly different neuronal cell types that could easily be clustered. A representative neuron from each type is shown in Fig. 2a-c. We applied seven descriptors to this set (all but taper-rate), assigned Sholl functions to neurons and computed distances for each descriptor. As shown in the detection Table 1, the performance of all descriptors was optimal in that each class was completely detected by at least two descriptors. Remarkably, the branching pattern descriptor detected all three classes, suggesting that this descriptor is sufficient in classifying this particular dataset. The parameterized TMD descriptor (TMD Sholl) performed equally well as other descriptors, but better than its classical version. The dendrogram based on the TMD Sholl is shown in Fig. 2f. The horizontal axis of the dendrogram represents all the neurons in this dataset while the vertical axis represents the distance between clusters. Interestingly, interneurons were fully detected by every single descriptor, suggesting that this cell type is characterized by a unique set of morphological features.

We compare the performance of our Sholl descriptor methods to conventional clustering techniques to determine whether these methods can clearly separate classes of neurons in this dataset. First, we used morphological parameters from our eight descriptors to represent neurons as vectors (see §4.13), and applied a hierarchical cluster analysis algorithm. Next, we extracted 10 morphological parameters for this dataset using the L-Measure software [28] and applied the same cluster analysis algorithm. To make a fair comparison, we chose specific morphological parameters from L-Measure to match features that are captured by our descriptors (i.e, number of branches, leaves, bifurcations, path distance, and Euclidean distance). Fig. 2 shows a dendrogram of the linkage distances between the 29 neurons based on L-measure extracted features (Fig. 2d) and Sholl descriptors (Fig. 2e). Neurons are color coded according to morphological type. Cluster analysis based on features captured by the descriptors results in clear separation of classes into three clusters (Fig. 2e), whereas the L-Measure method returns two clusters with significant intermingling of neurons from the three types (Fig. 2d). Therefore, our results demonstrate that our combined descriptors can outperform conventional methods in clustering different neuronal cell types.

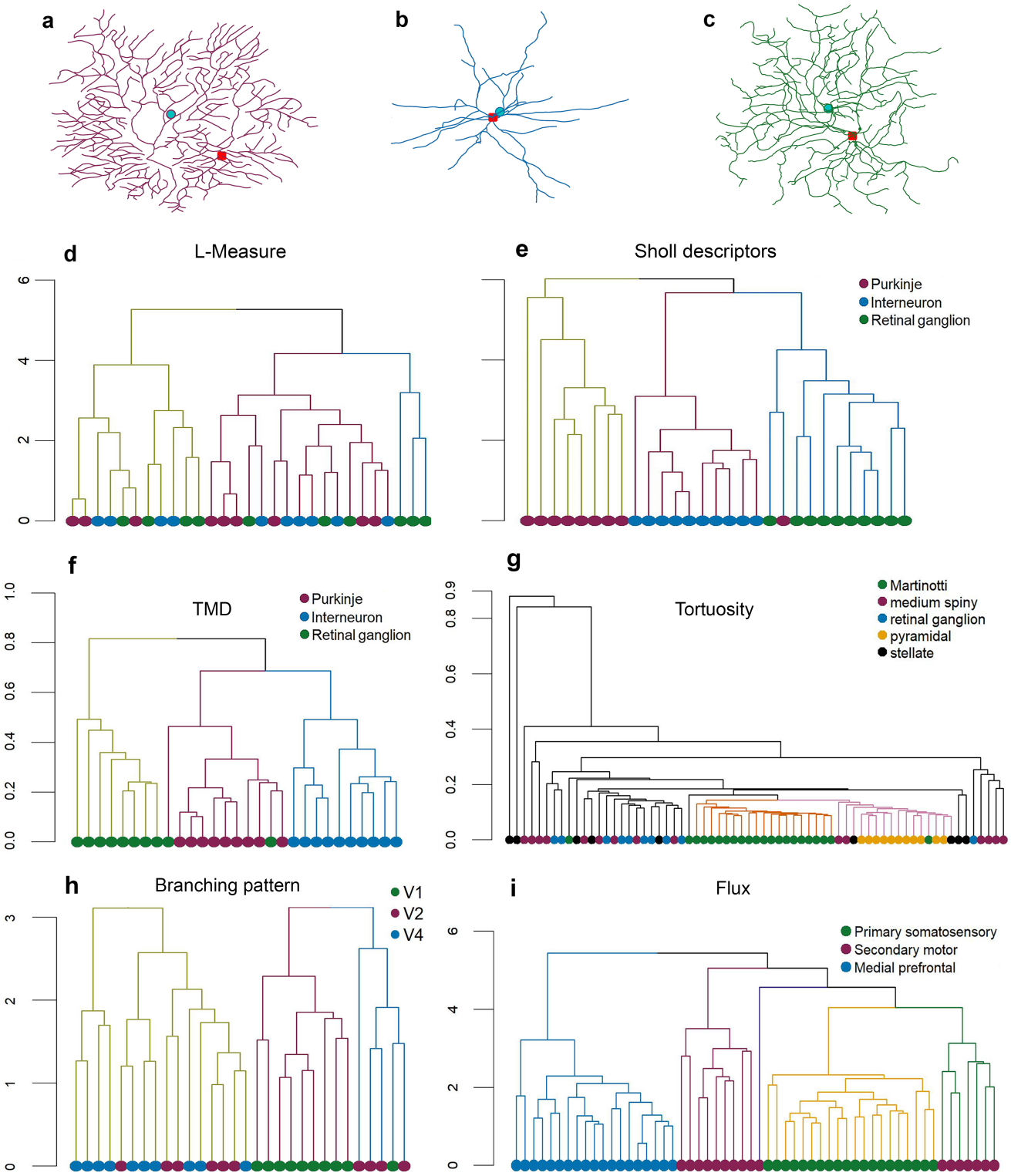

Fig. 2. Hierarchical clustering trees for all datasets. Representative neuronal reconstructions of (**a**) purkinje, (**b**) interneuron, and (**c**) retinal ganglion cell, in magenta, blue, and green colors respectively. Representative dendrograms for morphological parameters extracted from (**d**) L-measure software, (**e**) Sholl descriptors. Dendrograms based on descriptors for (**f**) TMD for dataset 1, (**g**) Tortuosity for dataset 2, (**h**) Branching pattern for dataset 3, and (**i**) Flux for dataset 4. For the reconstructions (**a,b,c**), the red dot represents the soma and the green dot is the barycenter of all nodes.

Distinguish among classes based on a single feature. Dataset 2 included 67 neurons from five different regions of the mouse brain: retinal ganglion cell (n=10), basal ganglia medium spiny (n=15), somatosensory stellate (n=9), hippocampal pyramidal (n=11), somatosensory Martinotti (n=22). This is the only dataset that included dendritic width in the reconstructions, which allowed us to implement the taper rate descriptor. The dendrogram in Fig. 2g shows the cluster analysis based on the tortuosity descriptor function. This Sholl descriptor resulted in efficient separation of neuronal cells types, particularly for the Martinotti (detection rate=91%), and hippocampal pyramidal cells (detection rate=92%). Table 1 reports the detection rates (see §4.7) for all descriptor functions for dataset 2. Detection rates highlighted in pink are above 80% while rates of 100% are highlighted in green. For comparison, the leaf index descriptor performed poorly in separating four of the five neuronal cell types, as evidenced by the low detection rates. This suggests that this particular morphological feature is largely uniform across these cell types (Martinotti, medium spiny, pyramidal, and stellate).

Dataset 3 comprises pyramidal cells in layer 3 of different cortical areas of the vervet monkey brain: primary visual cortex (V1) (n=10), V2 (n=10), and V4 (n=10). These reconstructions consisted of only basal dendrites. Prior reports have revealed regional differences in pyramidal cell morphology in the monkey brain [4, 8]. Specifically, pyramidal cell size, dendritic complexity, and spine density increase from primary visual cortex (V1) to higher order visual areas. Therefore, as a proof of concept we sought to recapitulate these findings by running our descriptors on reconstructions of pyramidal neurons from different areas in the visual cortical hierarchy. We expected at least to cluster pyramidal neurons from V1, V2, and V4, based on the branching pattern descriptor. Indeed, the dendrogram in Fig. 2h based on this descriptor reveals excellent separation of V1 neurons with some intermingling among V2 and V4 neurons. The wiring descriptor performed equally well in clustering, with excellent separation of V1 neurons and V4 neurons, and reasonable separation of V2 neurons (Table 1).

Subclustering within a neuronal class. To ensure sufficient coverage of neurons from different species, we also tested our Sholl descriptors in clustering pyramidal cells from different cortical areas in the rat brain. Therefore, dataset 4 consisted of rat pyramidal cells in layer 5 of somatosensory (n=20), secondary motor cortex (n=15), and medial prefrontal cortex (n=19). The discriminative accuracy in separating the three neuron groups with many of the descriptor functions was very high. For example, the cluster analysis based on the flux descriptor shown in the dendrogram in Fig. 2i resulted in nearly perfect clustering. The detection rate was highest in both medial prefrontal and somatosensory cortex (100% detection), followed by secondary motor cortex (94% detection). Likewise, the branching pattern, wiring, and TMD Sholl descriptors performed equally well as shown in Table 1. Remarkably, the combined descriptor approach yielded complete separation with three distinct clusters (Supplementary Fig. 12a) (see §4.14). Interestingly, the majority of pyramidal cells in secondary motor cortex formed their own distinct cluster, while several of these cells were clustered with other pyramidal cells in primary somatosensory cortex. This suggests the existence of two subpopulations of pyramidal cells in secondary motor cortex. Indeed, when we visually examined these neurons, we found striking similarities in morphology with pyramidal cells in primary somatosensory cortex. Therefore, the discriminative performance of certain descriptor is sufficient in separating neurons in different cortical areas of the rat brain, but more importantly, is powerful enough in revealing sub-clustering in a population of neurons.

Morphological aberrations. Revealing morphological aberrations resulting from neurodevelopmental and acquired disorders is an important step in understanding the pathophysiology of these diseases. Thus, unbiased methods to distinguish and separate normal neuronal morphology from aberrant morphology becomes essential. Therefore, dataset 5 included pyramidal neurons in layer 5 of rat somatosensory cortex in control (n=20) and experimental condition (n=16). This study assessed morphological changes of cortical pyramidal neurons in hepatic encephalopathy. Interestingly, the authors report that although dendritic arbors remained unchanged in rats with hepatic encephalopathy, dendritic spine density was significantly reduced [29]. Indeed, as one would expect, the detection rates based on all the descriptors was low (Table 1), suggesting that neurons from the control group and the experimental group were virtually indistinguishable. Unsurprisingly, the combined descriptor approach (Supplementary Fig. 12b) (see §4.14) resulted in intermingling of neurons from the control and experimental group. These results confirm previous findings that neuronal morphology is largely unaltered in rat cortical neurons with hepatic encephalopathy. More importantly, although the study only assessed path length and number of terminal ends in control versus experimental condition, we reveal using multiple Sholl descriptors that additional parameters related to neuronal morphology are in fact comparable between the two neuron groups. Nevertheless, although our descriptors do not reveal any structural differences between neurons in the two groups, there may be other features that differ which our descriptors do not capture. In a future study, we intend to construct additional Sholl descriptors that may potentially reveal structural differences between control and experimental neurons in this dataset.

Classification and metric learning. Finally we apply our descriptors to a relatively large dataset of 405 neurons in order to classify them. Dataset 6 is compromised of: Retinal ganglion cell (n=83), basal ganglia medium spiny (n=157), hippocampal granule (n=76), hippocampal pyramidal (n=49), and mPFC pyramidal (n=40). A classification scheme is then used which aims to (i) generate a single metric that can differentiate the classes, and (ii) assign a new neuron to one of the given classes that it shares the most features with. This was accomplished

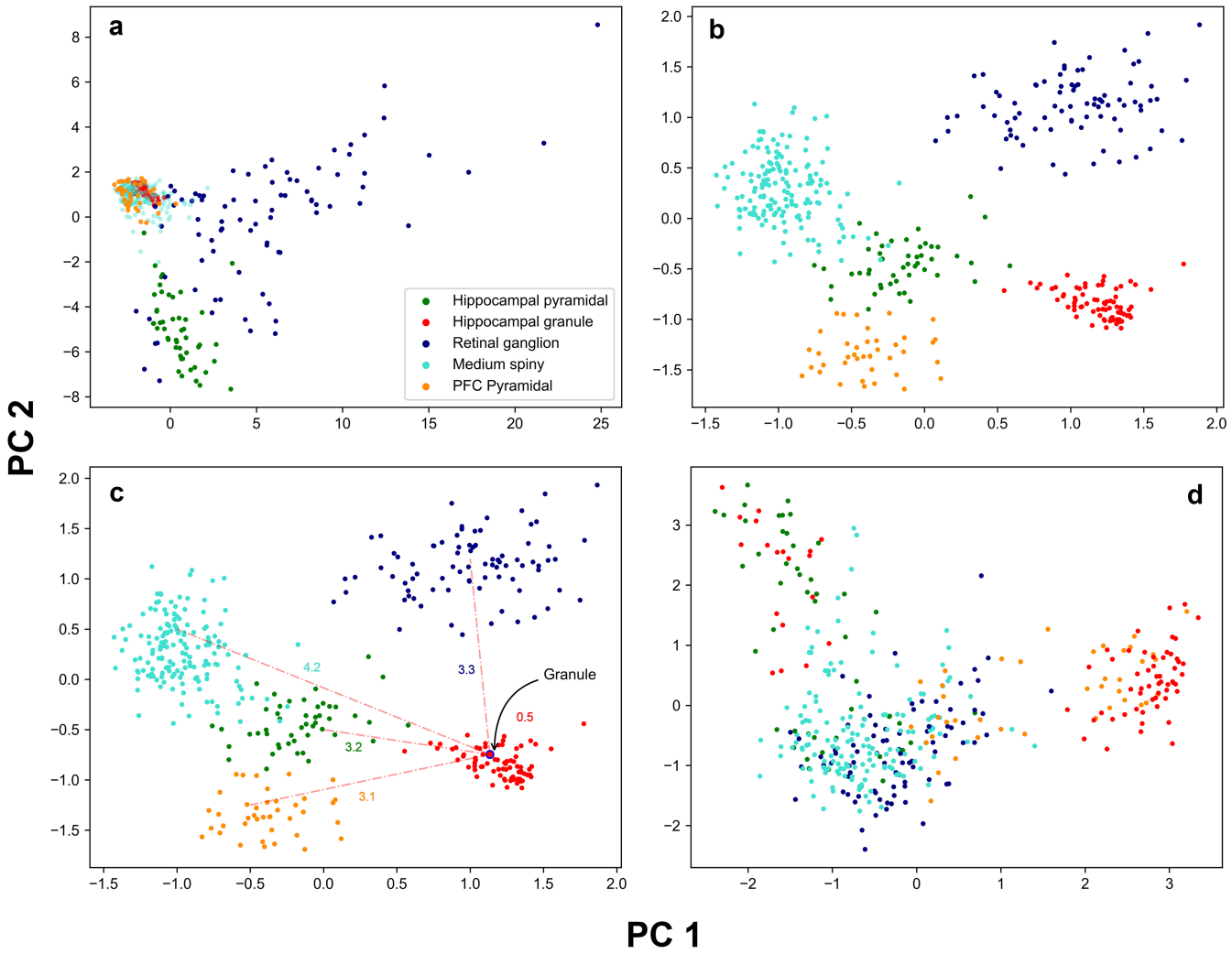

Fig. 3. Classification and Metric learning for dataset 6. Each dot represents a neuron which is color coded according to its class. PCA was performed to reduce dimensions in **(a-d)**. (**a**) Vectorized neurons in Euclidean space. (**b**) KNN classification in the new metric space. (**c**) Distances from newly introduced neuron to all 5 classes. (**d**) Overfitting test by permuting the classes.

by first implementing all Sholl descriptors on this dataset to determine which ones resulted in the best detection rates ('feature selection'). We subsequently chose these features (i.e. descriptors) for inclusion in our classification scheme. Specifically, Table 1 shows that the energy and total wiring descriptors were ineffective in distinguishing among classes in this dataset, as evidenced by low detection rates (below 80%). Therefore, these descriptors were removed and only the descriptors that performed well in separating these classes were included (flux, leaf index, branching pattern, tortuosity and TMD). Next, we vectorize the neurons based on these descriptors, resulting in classes of vectors in Euclidean space (in $\mathbb{R}^{25}$). Principal Component Analysis (PCA) is subsequently used to reduce dimensions (Fig. 3a). Neurons from different classes in this dataset were largely overlapping and poorly separated. Data are then fitted and transformed into a new metric space using the Large Margin Nearest Neighbor (LMNN) metric learning algorithm which learns a Mahalanobis distance metric in the K-Nearest Neighbor (KNN) classification setting [30] (see §4.15). This powerful new approach results in excellent separation of classes in this dataset. Fig. 3b shows the data plotted in the new transformed space. In Fig.3c, we introduce a hippocampal granule neuron (so we have *a priori* knowledge of its class) into this dataset. The KNN classifier successfully predicted the class of the newly introduced cell. Note that the distances in the new metric space from the newly introduced cell to all 5 classes are depicted in Fig. 3c. Finally, Fig. 3d shows the result of an over-fitting test where we permuted the vectors among our five classes and plotted the data using PCA in the newly transformed space. The plot shows fairly poor separation which is an indication that our selected features were geometrically meaningful in classifying this dataset (see §4.16).

Dataset	Description	Energy	Flux	Leaf index	Branching	Wiring	Tortuosity	TMD Classical	TMD Sholl	Taper rate
1	Mouse retinal ganglion	91	90	100	100	82	90	91	91	
.	Mouse cerebellum Purkinje	89	82	31	100	78	73	90	100	
.	Mouse L5 interneuron, mpfc	100	100	100	100	100	100	100	100	
2	Mouse L5 Martinotti, occipital	48	73	68	68	73	91	64	55	43
.	Mouse medium spiny, basal ganglia	40	67	59	60	60	63	47	53	87
.	Mouse retinal ganglion	100	90	82	90	90	64	91	70	60
.	Mouse pyramidal, hippocampus	62	73	50	64	50	92	50	83	82
.	Mouse stellate, somatosensory	60	36	43	33	25	30	33	33	67
3	Monkey L3 pyramidal, V1	60	91	70	100	91	70	90	91	
.	Monkey L3 pyramidal, V2	64	82	53	82	82	60	58	70	
.	Monkey L3 pyramidal, V4	46	82	43	90	91	58	70	82	
4	Rat L5 pyramidal, primary somatosensory	55	100	75	100	90	75	85	95	
.	Rat L5 pyramidal, secondary motor	71	94	48	76	94	71	82	94	
.	Rat L5 pyramidal, medial prefrontal	75	100	100	100	84	84	95	95	
5	Rat L5 pyramidal somatosensory (control)	65	67	67	67	65	73	70	67	
.	Rat L5 pyramidal somatosensory (experimental)	56	56	55	57	50	67	69	63	
6	Mouse pyramidal, subiculum	76	94	40	84	61	80	80	92	
.	Mouse granule, hippocampus	61	99	87	97	99	99	89	96	
.	Mouse retinal ganglion	84	70	90	88	78	50	69	80	
.	Mouse medium spiny, basal ganglia	72	77	90	82	66	87	81	73	
.	Mouse L2-3 pyramidal, PFC	30	48	32	36	36	88	45	46	

Table 1. Detection rates for all classes of neurons based on all descriptors. Numbers represent percentages. Green represents complete detection of a class (100%), while pink represents detection rates between 80%-100%. The taper rate was only run on dataset 2 since measurements of dendritic width were available for this dataset

.

3. Discussion

In this work we introduce a novel method of comparing descriptor functions of tree structures for classification of neuronal cell types. Importantly, we obtained substantially better clustering results when we compared the performance of our descriptors with conventional methods. By constructing a metric space valued function for a single neuron (Sholl function) to capture the evolution of a particular morphological feature as distance from the soma, we are able to compare Sholl functions for all neurons in a dataset. We illustrate that certain descriptor functions can effectively cluster classes of neurons with subtle morphological variations, as well as discriminate among widely different classes of neurons in agreement with expert assignment. Additionally, we leverage metric learning techniques to provide more robust classification. Our framework is powerful enough to separate diverse classes of neurons across different brain regions and species. Our results reveal several key findings regarding this tool kit of descriptors.

The six representative datasets used in this study were chosen to ensure morphological diversity and are thus derived from different areas, layers, and species. In dataset 2 (different types of neurons taken from different regions of the mouse brain), we show that the TMD and tortuosity descriptors performed very well in clustering this dataset. Specifically, based on the tortuosity descriptor the Martinotti and pyramidal cells each formed their own cluster. Interestingly, dendritic tortuosity has been shown to vary among different non-pyramidal neuron classes in the rat brain, whereby Martinotti cells in layer II/II and V of the frontal cortex have higher tortuosity than other cell types [31]. In the mouse brain, dendritic tortuosity increases as a function of increasing branch order on apical dendrites of hippocampal CA1 pyramidal cells [32]. Additionally, dendritic tortuosity of layer II/III pyramidal cells appears to increase from caudal to rostral regions in mouse cortex [33]. Our tortuosity descriptor is therefore powerful in distinguishing among neuron groups with non-uniform dendritic tortuosity. The upper limit for tortuosity values

appears to be 2, which is consistent with prior reports [31]. Importantly, we improved discrimination accuracy by using a combination of descriptors which effectively assigns weights to the function with the best separation results.

Anatomical studies have shown that interneuron morphology is highly diverse in the cerebral cortex. For example, interneurons with similar somatodendritic morphology may differ in axonal arborization patterns [13]. Therefore, axonal morphometric features are typically required for accurate classification of interneurons as they have been shown to capture important differences among interneuron subtypes [34]. We did not analyze axonal features in our descriptors as that could explain why Martinotti, and medium spiny neurons were largely intermingled (dataset 2). However, based on dendritic features alone some of our descriptors (tortuosity and TMD) were able to reliably distinguish interneuron subtype (Martinotti) from other neuronal cells types such as purkinje and retinal ganglion cells (dataset 2). In a future study, we will focus our efforts on interneuron subtypes in order to incorporate important axonal features into our descriptors for a more accurate classification scheme.

Prior work has revealed regional differences in pyramidal cell morphology in the monkey brain. Specifically, in the Old World macaque monkey, pyramidal cells become progressively larger and more branched with rostral progression through V1, the secondary visual area (V2), the fourth visual area (V4), and inferotemporal cortex (IT) [4, 8, 36]. Therefore, we were interested in testing whether our descriptors can detect differences in the morphology of pyramidal cells from different visual cortical areas of the vervet monkey (dataset 3), another species of Old World monkeys. Indeed, we find that the performance of the branching pattern descriptor results in excellent clustering of cells from V1, V2, and V4. This suggests distinct differences in the branching pattern of basal dendrites of pyramidal cells residing in these areas. The wiring descriptor which is a proxy for total dendritic length yielded reasonable clustering of neurons, with some intermingling of neurons from all areas. This is not surprising given that in some species such as the tree shrew, differences in pyramidal cell morphology throughout the visual cortical hierarchy is less pronounced [37]. Even in rodents, regional differences in pyramidal cell morphology appear to be less noticeable than in primates [38, 39]. Therefore, the fact that pyramidal cells in V2 are intermingled with cells in V4 in our cluster analysis based on the wiring descriptor reflects genuine similarities between these two population of cells, and further suggests inter-neuron variations within each visual cortical area.

Collectively, the results from this study highlight the robustness of our framework in quantitatively characterizing and discriminating among different neuronal cell types. Certain morphological features and thus specific descriptors are better suited in separating distinct neuronal cell types. For instance, the branching pattern descriptor appears to perform very well in detecting most neuronal cell types. This descriptor measures how far or how fast nodes appear (bifurcations) and disappear (leaves), as measured from the soma. Conversely, the energy descriptor, which reveals the distribution of nodes around the soma, appears to reliably detect retinal ganglion cells, purkinje, and interneurons. Importantly, our use of metric learning techniques resulted in more optimal classification. Progress in the development of unbiased clustering methods to distinguish among groups of neurons will further our understanding of the relationship between brain structure and function. The toolkit of morphological descriptors introduced here, and the development of new methods will potentially lead to the discovery of novel sub-classes of neurons [40]. Additionally, our descriptors will aid efforts to uncover differences between normal and aberrant neuron morphology which is commonly associated with various disease states. For instance, changes in dendritic morphology have previously been described in a number of disease states, including Alzheimer's disease [15], schizophrenia [41], and mental retardation [42]. Given that our tool kit of descriptors discriminated among different types of cells as well as revealing subclasses of cells, its utility may be extended to the study of brain diseases potentially identifying which subtypes may be affected in various disease states.

4. Methods

In this study we developed an unsupervised clustering and supervised classification framework based on relevant morphological features to differentiate among and classify different neuronal cell types. Given a topological or spatial feature of neurons, denoted by the greek letter ϕ, such as branching pattern, tortuosity, total wiring, etc, we associate to such a feature a "Sholl descriptor". Specifically, a Sholl descriptor is a map from the set of neurons to a metric space of functions, with reasonable properties (Definition 4.3). We then use this assignment to construct a metric d_ϕ on the set of all neurons §4. In other words, we endow the set of neurons with a metric space structure for every topological feature ϕ. This (Sholl) metric measures how far neurons are in that metric for that particular feature, so that the closer the neurons are under that metric, the more of that feature they share. The fundamental idea of a Sholl descriptor is presented in Fig. 4. In this case, a branching pattern function is presented for a simple tree. The value of the descriptor for that neuron is the step function on the right.

Sholl descriptors form a toolkit to analyze neuronal morphology. Descriptors combined with standard hierarchical clustering methods, a detection algorithm (also used for feature selection), grid search, and metric learning functions are used to cluster and classify a dataset of neurons.

1. Analysis based on a single descriptor:

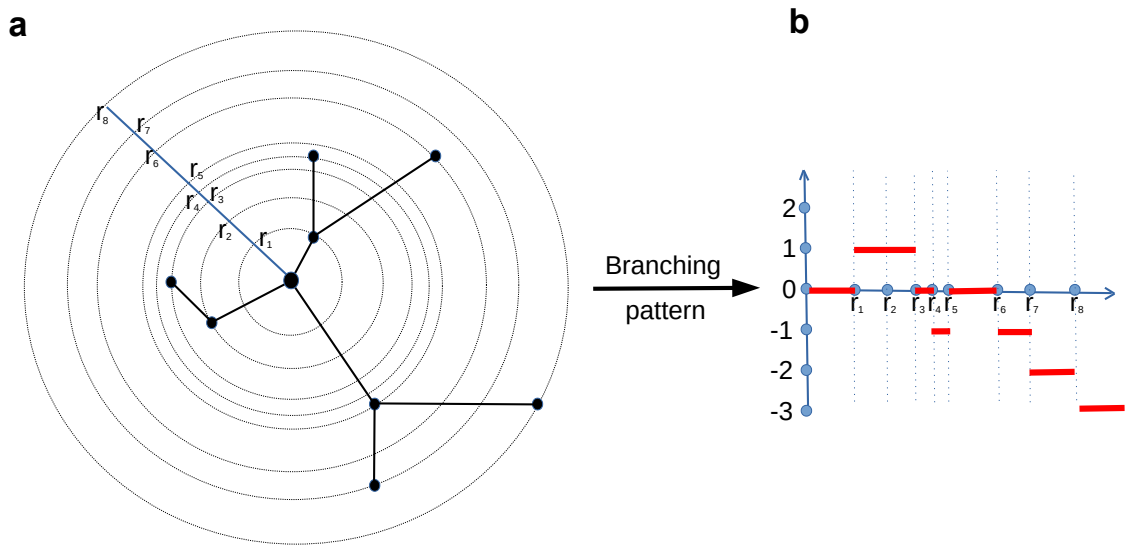

Fig. 4. Sholl descriptor using branching pattern (see §4.9.1). (**a**) A representation of a tree. (**b**) the corresponding branching function as a function of distance from soma.

- (Clustering) Given a dataset of unlabeled neurons we can run a particular descriptor in order to cluster them according to that descriptor. For example, if $\phi = T$ is tortuosity, we can set the distance matrix for the associated Sholl metric d_T and then run standard hierarchical clustering to obtain dendograms. The obtained dendrogram reveals whether neuronal cell types differ according to their tortuosity (i.e. cluster together), or if their tortuosity is comparable (cells from different neuron types will be intermingled).
- (Detection) This method assesses the performance of a given descriptor in identifying a desired feature within a dataset. For example, we can run a given descriptor ϕ on a dataset of neurons to detect which set of neurons under which label is "grouped together" under the descriptor ϕ. So if $\phi = T$ is branching pattern, we can represent three types of neurons in a dataset with colors red, green, and lavender (Supplementary Fig.11). This allows us to determine if d_ϕ detects the red neurons within a certain percentage, that is the detection rate of red neuron within a ball in that metric. Detection rates can be used as feature selection when running classification schemes.

Typically, when a given dataset of neurons which is distributed among classes $C_1, \ldots, C_k$ (and thus labeled), we can run the detection algorithm one Sholl descriptor at a time to allow us to differentiate at least one class, but it leaves the other classes undifferentiated. In other words, using the metric d_ϕ associated to the descriptor ϕ, one or more classes are singled out (i.e. neurons will cluster) while the remaining classes may be indistinguishable. However, the other classes may be differentiated by other descriptors. Therefore, by combining descriptors, we can obtain a complementary combined effect that can better separate the classes.

2. Analysis based on a combination of descriptors (details in §4.12):

- (Vectorization and unsupervized clustering) A given neuron can be converted into a vector using our descriptor functions and metrics. There are different approaches that may be used, however we implement§4.13. Using the feature vectors, we then run a standard hierarchical clustering algorithm.
- (Classification) Given a dataset of neurons distributed among a number of classes, and a number of morphological features, we can determine which class a newly introduced neuron is associated with (i.e. shares the most features with). More precisely, suppose we are given classes of neurons $C_i, 1 \leq i \leq n$, and morphological descriptors $\phi_1, \ldots, \phi_k$ which can be measured for all neurons. Therefore, given a newly introduced neuron N, we can determine which class, with measurable likelihood, this neuron belongs to. Using metric learning [30] we can determine which features are comparable among the classes.
- (Differentiation) By linearly combining descriptor metrics, we can separate classes and obtain a new clustering metric. This approach is similar to metric learning in that it will separate different classes, unless the classes are truly indistinguishable.

4.1. **Notation and Terminology.**

- Capital letter N represents a neuron seen as a tree in 3-space.

- A class C of neurons is a set containing a selection of neurons of a particular type.
- A node in a neuron N is either the soma, bifurcation point, or a termination point

$$\{\text{Nodes}\} = \{\text{Bifurcations}\} \cup \{\text{Soma}\} \cup \{\text{Terminations}\}$$

  A branchpoint can be used interchangeably with bifurcation point. A leaf can be used interchangeably with a termination point.
- The number of terminal nodes in a tree is denoted as degree which is a proxy for tree complexity. The number of branches of a tree is twice the degree of that tree minus one, while the number of bifurcations is degree minus 1.
- Radial distance means Euclidean distance as measured from a point to the soma.
- Path distance is the distance along dendrites.
- Two nodes are parent-child related if they are adjacent on a branch. The node closer to the soma in path distance is called the parent, and the node farther away from the soma is called a child.
- A branch is part of the dendrite between the father and any of its children (at most two).
- $R(N)$ is the "span" of the neuron, that is the largest radial distance of any of the nodes (typically a leaf). A neuron has span $R(N)$ if it can fit in a ball of radius $R(N)$ and in no smaller ball.
- $L(N)$ is the length of the longest dendrite stemming from soma and ending at a termination point.
- A neuronal feature is denoted by the Greek letter ϕ. It is always topological or morphological in nature, and its associated (Sholl) descriptor will also be denoted by ϕ.
- A feature ϕ gives rise to a metric on the set of neurons which is denoted by d_ϕ.

4.2. **Representation of neurons.** We model a neuron N as a collection of rooted binary trees embedded in 3 space $\mathbb{R}^3$, all having a common root (the soma). These rooted treed are also called the "primary" trees. In this paper, all 3D neuronal reconstructions are acquired from the public repository NeuroMorpho.org [27]. The morphological structure of individual neurons is retrieved from an SWC file which contains a digital representation of the neuron as a tree structure that consists of points/markers. Each marker has associated properties such as 1) its 3D spatial coordinates, 2) its radius denoting the thickness of the branch segment at a specific 3D location 3) a node type indicating whether it is soma, axon or dendrite, and 4) one parent marker to which it directly connects through neuronal arbors. Well-defined geometric constructions on neurons need to be invariant under the affine isometry group of $\mathbb{R}^3$, and since the soma is always at the origin of any reference frame, it is sufficient to only consider invariance by rotations and reflections fixing the soma, and disregard translational invariance. Geometric constructions that only depend on conformal measures, like angle and distance, can result in interesting geometric and topological invariants for neurons.

4.3. **Sholl descriptors.** Below we detail the construction and definition of Sholl descriptors.

Definition 4.1. *A "Sholl descriptor" is any rule that associates to a given neuron (seen as a tree embedded in $\mathbb{R}^3$) a compactly supported function whose independent variable is the distance from the soma, either path or radial, and whose values are in a metric space X. We further require this function to be both isometry invariant and stable.*

More precisely, for a given neuron $N \subset \mathbb{R}^3$, a Sholl descriptor $\tilde{\phi}$ associates a function

$$\tilde{\phi}_N : [0, \infty[\longrightarrow X$$

which is supported on either $[0, R(N)]$ or $[0, L(N)]$ (for definitions, see §4.1). Here X is a metric space, and ϕ is stable in the sense of 4.10. Isometry invariance means that if N' is obtained by rotating or reflecting N about respectively a line or a plane passing through the soma, then $\tilde{\phi}_N$ and $\tilde{\phi}_{N'}$ are identical functions. As we will explain below, our constructions will be independent of scale, so we can consider that all of our neurons are normalized to be in a ball of radius 1 in space (see §4.4). So succinctly, *a Sholl descriptor associates to every neuron N a Sholl function $[0, 1] \to X$, with X a metric space, satisfying the stability and isometry invariance properties.*

The following are Sholl descriptors that we discuss in this paper (details in §4.3).

(1) **Branching pattern**: This is an integer valued descriptor given by the number of bifurcations from which we subtract the number of leaves, all within a given radius r from the soma. As the radius changes, this number changes. We get a different Sholl descriptor if we consider the same quantity (number of bifurcations - number of leaves) but within path distance r from the origin.
(2) **Tortuosity**: Tortuosity between two nodes is measured as the quotient of the path distance by the Euclidean distance. The tortuosity descriptor measures the mean tortuosity between all adjacent nodes within radius r from the soma.
(3) **Flux**: This associates to a given radius r the sum of the angles between dendrites and normal directions to the sphere of radius r centered at the soma, at the points where the dendrites intersect with that sphere.The flux construction is related to, and can be viewed as an extension of the root-angle construction in [43]. This

construction considers, at any given leaf, the angle between the radial normal and the main branch ending at that point.

(4) **Taper Rate**: This is based on the width of dendritic segments at bifurcations, taken as a function of path distance to the soma. Dendritic tapering is a measure of the change in width along a dendritic segment from node to node.

(5) **Leaf Index**: This construction counts the number of terminations emanating from every node, plotted as a function of radial distance. If the node is taken to be the soma, this number is the total number of leaves, if the node is taken to be a leaf, the value is one (that leaf).

(6) **Energy**. By viewing the nodes as charged electrons, they generate a vector field. The resulting combined field vector at the soma (superposition principle) is now measured. This vector changes as the number of nodes is increased, giving us a Sholl descriptor. This descriptor detects the position of the soma with respect to the nodes. If we divide space in octants, with the soma in the origin, then the more nodes present in the same octant, the greater the energy.

(7) **Total Wiring**. This construction measures the total wiring of a neuron within a given sphere of radius r centered at the origin. This is the sum of the path distances of all dendritic segments within that sphere.

(8) **TMD**. This is the Topological Morphological Descriptor of [24] redesigned to be a Sholl function. The metric space target in this case is the space of persistent diagrams with the Wasserstein metric.

4.4. **Normalization.** Given a Sholl descriptor $\tilde{\phi}$, we can normalize it so that all function are supported on $[0,1]$. Let $\tilde{\phi}_N : [0,\infty[\longrightarrow X$ be the descriptor function for a neuron N. If $\tilde{\phi}_n$ is supported on $[0, R(N)]$, where $R(N)$ is the span of the neuron, that is the radial distance to the furthest point in N from the soma (see §4.1), define the corresponding normalized descriptor ϕ_N by :

$$\phi_N : [0,1]\longrightarrow X$$
$$\phi_N(r) = \tilde{\phi}_N(rR(N))$$

If $\tilde{\phi}_N$ is a function of path length, and thus supported on $[0, L(N)]$, then we normalize in the same way, with $L(N)$ replacing $R(N)$. All our descriptors are normalized and supported on $[0,1]$. The constructions we provide are such that all real-valued Sholl functions we consider are step functions. Fig. 5 illustrates these step functions for a very simple granule cell in mouse olfactory bulb.

4.5. **Functional Metrics.** Each Sholl descriptor defines a metric on the set of neurons. Let $\mathcal{N}$ be a given set of neurons. Let ϕ be a normalized Sholl descriptor which associates to each neuron N a descriptor function $\phi_N : I\longrightarrow X$, $I = [0,1]$, and X a metric space. We assume that no two neurons can be identical with respect to any feature we define. We thus have an inclusion (i.e. an injective map)

$$\mathcal{N} \hookrightarrow \mathrm{Map}(I, X) \quad , \quad N \longmapsto \phi_N$$

Let d be any metric on the space of functions $\mathrm{Map}(I, X)$. It induces a metric on $\mathcal{N}$ by setting

$$d_\phi(N_1, N_2) := d(\phi_{N_1}, \phi_{N_2}) \tag{1}$$

The distance we choose to work with is the "L^1 distance"

$$d(\phi_{N_1}, \phi_{N_2}) := \int_0^1 d_X(\phi_{N_1}(r), \phi_{N_2}(r))dr \tag{2}$$

All ϕ_N constructed in this paper are real-valued *step functions*, except for the Sholl-TMD in §4.9.8 . We give the formula for the distance in this case. Let ϕ_1, ϕ_2 be two step functions with jumps at radii $r_1, \ldots, r_q$ and $s_1, \ldots, s_\ell$ respectively. This means that ϕ_1 is constant on $[r_i, r_{i+1}[$, and similarly ϕ_2 is constant on $[s_j, s_{j+1}[$. Let

$$\{t_1, \ldots, t_{q+\ell}\} = \{r_1, \ldots, r_q\} \cup \{s_1, \ldots, s_\ell\}$$

and order the t_i's by increasing magnitude so we can assume

$$0 = t_0 < t_1 < t_2 \leq \cdots < t_{q+\ell}$$

Then the L^1-distance between the step functions is given by

$$d_\phi(N_1, N_2) := d(\phi_{N_1}, \phi_{N_2}) := \int_0^1 |\phi_{N_1}(r) - \phi_{N_2}(r)|dr = \sum_{i=0} (t_{i+1} - t_i)|\phi_1(t_i) - \phi_2(t_i)| \tag{3}$$

Other choices of metrics we can work with for real valued Sholl functions are the L^p metrics for $p > 1$ or the Sup metric $d(\phi_{N_1}, \phi_{N_2}) := \mathrm{Sup}_{r\in[0,1]}(\phi_{N_1}(r); \phi_{N_2}(r))$. This last metric is known to induce the compact-open topology on the space of functions when X a compact regular metric space. Once we can measure functional distances between descriptor functions, we can measure "Sholl distances" between neurons as indicated in (1). These distances are then used to cluster and classify neurons.

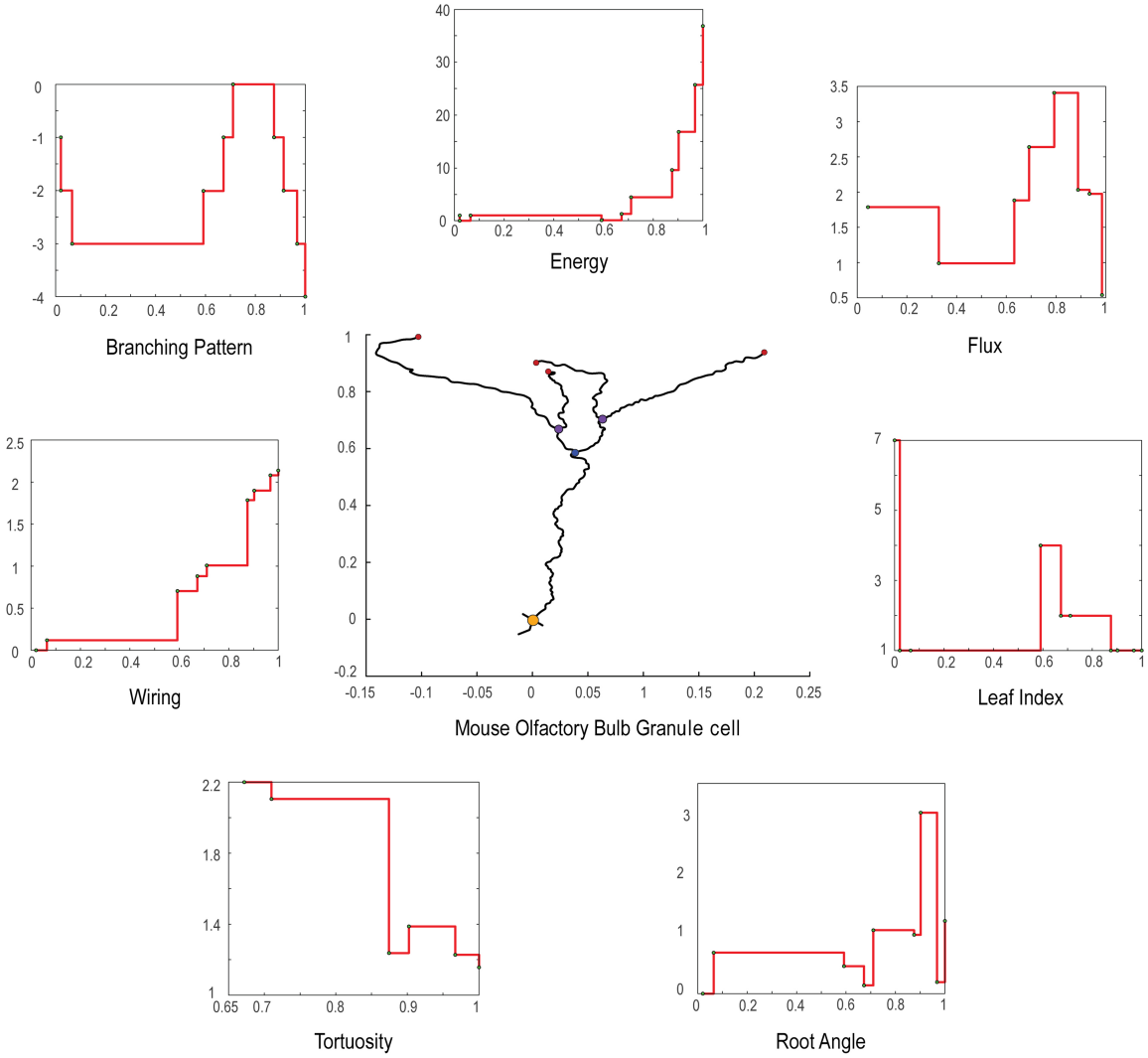

Fig. 5. Representative neuronal reconstruction of a granule cell in mouse olfactory bulb with the corresponding step functions for seven real-valued Sholl descriptors. This neuron was chosen because of its simple branching pattern which consists of 3 bifurcation points and 7 leaves. Notice that all neurons have been normalized to be included in a ball of a radius 1 centered at the soma. The step function for the "taper rate" descriptor is missing for lack of data. Also, the "TMD" descriptor is missing here, as it is not real-valued and cannot be easily presented.

4.6. **Clustering.** The ultimate goal is to find measurable and quantifiable morphological differences between classes of neurons. When given a Sholl descriptor, and a selection of neurons $N_1, \ldots, N_q$, the standard procedure is to generate a distance matrix associated to the descriptor

$$[d_\phi(N_i, N_j)]_{1 \leq i,j \leq q}$$

This is a symmetric matrix with positive entries, and zeros along the diagonal. Any such distance matrix produces a dendrogram using standard hierarchical clustering algorithms. It is not reasonable to expect a single descriptor to cluster faithfully a given set of classes of neurons (Supplementary Fig. 8). The advantage of developing multiple descriptors based on various morphological features reveals which features are uniform and which are different among classes of neurons. Combining descriptors to differentiate between classes of neurons is another method we use. This combination can be achieved at the level of distance matrices, or at the level of Sholl descriptors since these form a vector space of functions (in fact an algebra). Indeed, given two normalized Sholl descriptor functions $\phi_1, \phi_2 : I \to \mathbb{R}$, we can take linear combinations. This sum is also stable, as defined in §4.10, if we start with stable descriptors.

4.7. **Detection and Feature Selection.** Let $C_1, \ldots, C_k$ be k distinct classes of neurons. Each class consists of neurons to be compared with neurons from other classes. We say that a descriptor ϕ has at least an $n\%$ level of

detection of a class C_i if there is some ϵ ball B_ϵ^ϕ in the d_ϕ metric so that more than $n\%$ of all elements of C_i are within B_ϵ^ϕ, and of all elements in B_ϵ^ϕ, more than $n\%$ are from C_i.

Example 4.2. Suppose we have three classes of neurons C_1, C_2, C_3 each consisting of 5 neurons. Let ϕ be a given Sholl descriptor, and suppose there is an ϵ ball in the d_ϕ-metric, that contains 4 elements of C_1 and 2 elements from $C_2 \cup C_3$. This ball contains $\frac{4}{5} = 0.8$ of the total of all C_1-neurons (i.e. 80%), while $\frac{4}{6} = 0.66$ or 66% of all neurons in this ball are C_1-neurons. We say that the descriptor ϕ has detected C_1 to a 66% level at least, which is the lower percentage from among 80% and 66%. If for smaller ϵ, we still have as many neurons from C_1 in the smaller ball but we lose one neuron from C_2, then detection is now at 80%.

The detection algorithm is described in the supplementary material. We also use detection as a method for feature selection when we run several descriptors on a given set of classes. The set of descriptors with detection rates that are less than a certain percentage are deemed ineffective in differentiating among these classes and can thus be excluded from further analysis (see §4.15).

4.8. **Combination of Descriptors and Classification.** . A single descriptor may detect features within a family of neurons, but it alone may not be able to differentiate between many classes at once. The idea of "combining" several descriptors together into one single descriptor offers a more effective tool in differentiating between classes (this is what we call *classification*). The Sholl descriptor metrics are perfectly well-suited to provide such a classification. We have devised three combination methods, each being applicable within its own specific context. This was discussed at the beginning of §2 and the details can be found in the supplementary material §4.12.

4.9. **The Sholl Descriptors.** Given a neuron N viewed as an embedded tree in space, we define

$$N_r := (N \cap B_r)_c = \text{connected component of } N \cap B_r \text{ containing the soma} \tag{4}$$

Here B_r is the ball of radius r around the soma.

4.9.1. *The Branching Pattern Descriptor.* This morphological descriptor detects patterns that results from the distribution of branches and leaves relative to the soma (see Fig. 4). Dendrites emanate radially from the soma, branching in a binary way. Branches and leaves appear and disappear as we move away from the soma, and a measure of this birth and death of branches and leaves gives rise to a function of the radius we call the "branching pattern" function.

Let N be a neuron which we view as a collection of single rooted binary trees in $\mathbb{R}^3$, with the common root being the Soma. Label $B_1, \ldots, B_q$ the branch points of N and label all leaves by $L_1, \ldots, L_k$. Let $r > 0$ be the radial distance measured away from the Soma. Order the branch points and leaves by increasing r, so that if r_i indicates the distance of the i-th node to the soma, we have $0 < r_1 < r_2 < \cdots < r_q$ (equal radii can be removed by an infinitesimal perturbation).

Fixing a neuron N as before, associate to each $r \in I$ the number $\alpha(r)$ defined by

$$\alpha(r) = \#\{B_i \mid r_i \leq r\} - \#\{L_j \mid r_j \leq r\}$$

Let $R(N)$ be the span of the neuron N and define the function

$$[0, R(N)] \longrightarrow \mathbb{R}^+ \ , \ r \longmapsto \alpha(r_i) \text{ if } r \in [r_i, r_{i+1}[$$

The normalized version takes the form

$$\begin{aligned} \phi_N &: [0,1] \longrightarrow \mathbb{R}^+ \\ \phi_N(r) &= \alpha(r_i) \text{ if } r \in \left[\frac{r_i}{R(N)}, \frac{r_{i+1}}{R(N)}\right[\end{aligned}$$

This defines our branching-pattern descriptor ϕ. Note that

$$\phi_N(1) = -\#\text{primary branches of N}$$

since the number of primary branches is the difference between the number of leaves and the number of bifurcation points.

Example 4.3. In Fig. 6, Tree structures of two different neurons are chosen (A) pyramidal and (B) stellate. The corresponding Sholl descriptor functions reveal obvious difference (C). The red curve which depicts the branching pattern of the pyramidal cell reveals that branching occurs rapidly close to the soma, but much slower as you move further away from the soma. Conversely, branching for the stellate cell is changing uniformly and steadily as you move away from the soma. Both neurons have similar branching counts: neuron (A) has 21 bifurcations and 30 leaves, while neuron (C) has 32 bifurcations and 49 leaves. This gives that $\phi_{N_1}(1) = -9$ and $\phi_{N_2}(1) = -17$ as depicted.

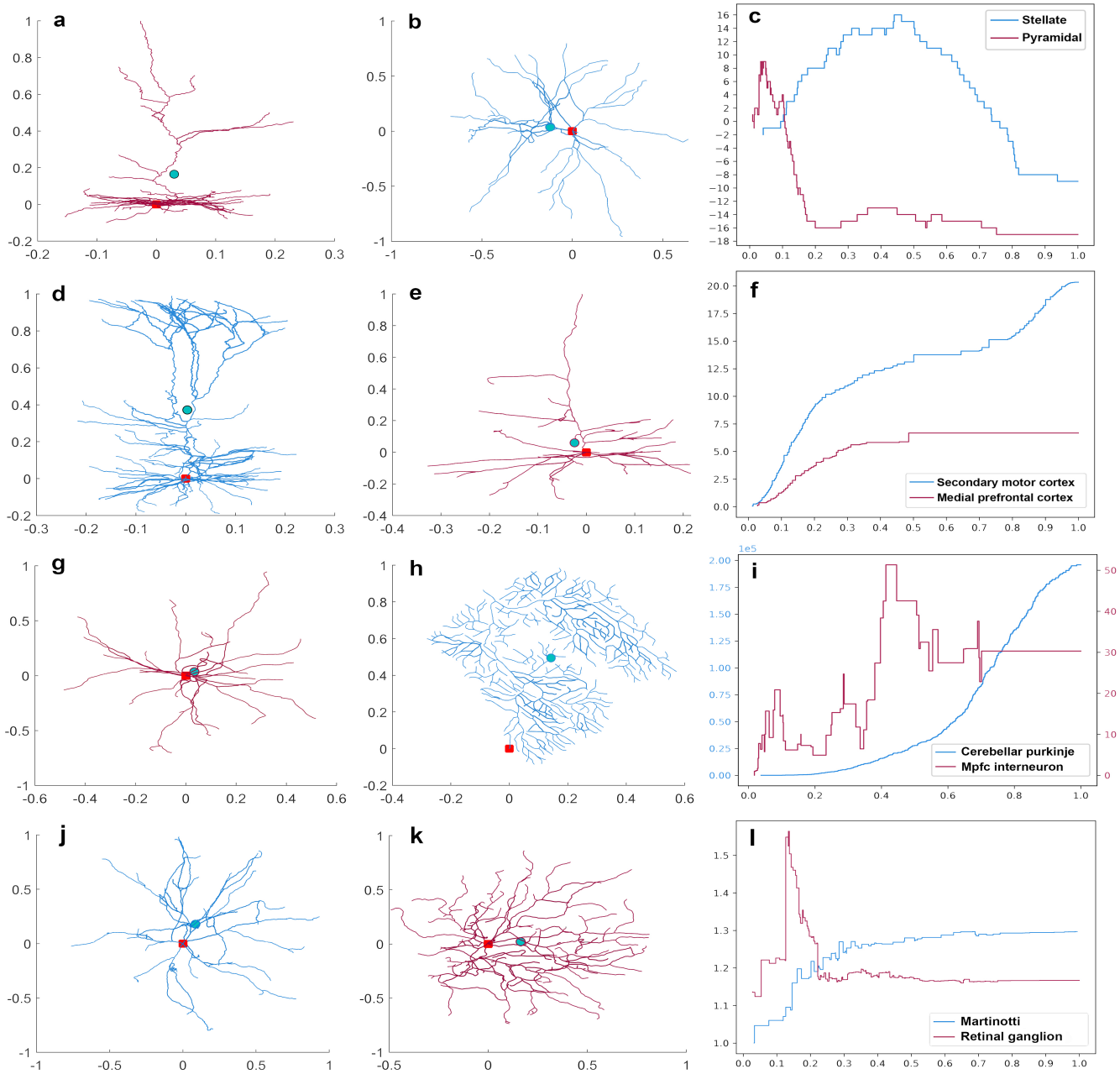

Fig. 6. Representative neurons and the corresponding step functions. (**a**) Pyramidal and (**b**) stellate cell. (**c**) The step functions for each neuron generated from the branching pattern Sholl descriptor. The branching Sholl functions show that neuron (**a**) is branching quickly near the soma, and leaves appear much closer to the soma than for neuron (**b**). The number of primary branches for each neuron is the value of the corresponding function at 1. (**d**) Pyramidal cell in secondary motor cortex and (**e**) in mPFC. (**f**) The step functions for neurons (**e**) and (**d**) generated from the wiring Sholl descriptor. The sharp increase towards the end for the wiring Sholl function for cell **d.** reveals the existence of an apical tuft. (**g**) Interneuron and (**h**) Purkinje cell. (**i**) The step functions for each neuron generated from the energy Sholl descriptor. The two energy Sholl functions show completely distinct features. We obtain maximal energy values for purkinje cells (a defining feature). The total energy value at the soma for neuron (**g**) is 200000 units, while this value is 50 for neuron (**h**). (**j**) Martinotti and (**k**) Retinal ganglion cell. (**i**) The step functions for each neuron generated from the tortuosity Sholl descriptor. The red dot represents the soma and the green dot is the barycenter of all nodes

4.9.2. *Tortuosity Descriptor.* Represent a neuron by an embedded tree in space, and label its nodes by $P_1, \ldots, P_n \in \mathbb{R}^3$. For any two nodes, we can consider both path distance and Euclidean distance between them. If P_i is a parent node and P_j is the child node, let $b_{i,j}$ be the dendritic path distance between these nodes, and let $d_{i,j}$ be the length of the segment $[P_i, P_j]$. The ratio of both distances is

$$\delta_{i,j} = \frac{b_{i,j}}{d_{i,j}} \quad \text{branch tortuosity} \tag{5}$$

Consider a neuron N with n nodes §4.1. Let (parent, child) be a pair of adjacent nodes. There are exactly n such pairs coinciding with the number of branch segments. Define the average tortuosity of N to be the average sum

$$\alpha(N) = \frac{1}{n}\sum_{(i,j)} \delta_{i,j} \quad , \quad (i,j) \text{ running over all (parent-child) pairs} \tag{6}$$

It is clear that $1 \leq \alpha(N)$ for all choices of N. The (non-normalized) Sholl descriptor function associated to this construction is now given as follows: order the the nodes of N by increasing radii as before. Then define

$$\begin{array}{rcl} T_N : [0, R(N)] & \longrightarrow & \mathbb{R} \\ r & \longmapsto & \alpha(N_{r_i}) \quad , \quad r_i \leq r < r_{i+1} \end{array}$$

where $B(r)$ is the ball of radius r around the soma. We then take the normalized version §4.4.

4.9.3. *Taper Rate Descriptor.* We start with a neuron N and list all *path distances* of the nodes to the soma in increasing order $0 < \ell_1 < \cdots < \ell_k$. Each node has a dendritic thickness (or width) that tapers as we move away from the soma along the dendrite. We can measure the tapering rate as a function of path distance. More precisely, define

$$TP_N(r) = \text{dendrite thickness at the node at } \ell_i, r \in [\ell_i, \ell_{i+1}[$$

and then take the associated normalized Sholl descriptor by dividing ℓ_i by the length of the longest dendrite. This is a Sholl function whose variable is path length and not radial distance.

4.9.4. *Flux Descriptor.* We define the Sholl descriptor F and the associated flux functions $F_N : [0,1] \longrightarrow \mathbb{R}$ for a given neuron N. Let $N'_r = N \cap B_r$, where B_r is a ball of radius r centered at the soma. Notice that N'_r can be different from N_r (4) if there are dendrites that leave B_r and then enter again. If a dendrite crosses the boundary sphere S_r at a point $P \in N \cap S_r$, we identify the parent of N (inside the sphere) and the child of N (outside the sphere). So the parent and child are on either sides of the sphere. The direction vector $\overrightarrow{AB}$ from father to child points outward if A is inside, and points inward if A is outside the sphere. Consider the segment $[A, B]$ and let C be the point on the segment that cuts the sphere. We then assign the value

$$f_N(P) = \frac{\overrightarrow{AB}}{|AB|} \cdot \frac{\overrightarrow{OC}}{|OC|}$$

This is the cosine of the angle between the unit vector along $\overrightarrow{AB}$ and the normal to the sphere going through C. This value is maximal if $\overrightarrow{AB}$ is aligned with the radial vector at C and the angle is zero.

To define the total flux function, order the nodes of N as before by increasing values of their distances from the soma $0 < r_1 < \ldots < r_k$. For every $r = \dfrac{r_i + r_{i+1}}{2}$, take the sphere of that radius r, look at all dendrites intersecting that sphere at $P_1, \ldots, P_k$ and add up the values obtained from the construction outlined above. This value is

$$F_N(r) = \sum_i f_N(P_j) \quad , \quad P_j \in N \cap S_r$$

This gives rise again to a step function $F_N : [0,1] \longrightarrow \mathbb{R}$ by setting

$$F_N(r) := F_N\left(\frac{r_i + r_{i+1}}{2}\right) \quad \text{if} \quad r \in \left[\frac{r_i + r_{i+1}}{2}, \frac{r_{i+1} + r_{i+2}}{2}\right[$$

Additionally, if $A = (a_1, a_2, a_3)$ is the parent marker and $B = (b_1, b_2, b_3)$ the child marker, such that either $|OA| < r < |OB|$ or $|OB| < r < |OA|$, that is on different sides of the sphere, the point of intersection $C = (c_1, c_2, c_3)$ of that sphere with the segment $[A, B]$ is obtained by setting $r^2 = c_1^2 + c_2^2 + c_3^2$ and $C = (1-t)A + tB$, then solving for t through a quadratic. The flux value at C is

$$f_N(C) = \frac{c_1(b_1 - a_1) + c_2(b_2 - a_2) + c_3(b_3 - a_3)}{\sqrt{(c_1^2 + c_2^2 + c_3^2)((b_1 - a_1)^2 + (b_2 - a_2)^2 + (b_3 - a_3)^2)}}$$

4.9.5. *The Leaf Index Descriptor.* From each node grows a new dendritic tree with a number of terminal points. Counting the number of these terminal points for each node P gives the "leaf index" of P, and we write it as $\mathrm{li}(P)$. When P is a leaf, we set $\mathrm{li}(P) = 1$. Figure 1b depicts the construction of this descriptor.

As before, given a neuron N, order its nodes $P_1, \ldots, P_k$ by increasing order of distance to the soma $0 < r_1 < \cdots < r_k$, and define the Leaf Index Sholl Descriptor as follows:

$$\begin{array}{rcl} LI_N : [0,1] & \longrightarrow & \mathbb{N} \\ \dfrac{r}{R(N)} & \longmapsto & \mathrm{LI}_N(r_i) \ \text{ if } r \in [r_i, r_{i+1}[\end{array}$$

Evidently $LI(1) = 1$ which is the value at the furthest leaf, while $LI(0)$ is the total number of leaves. This is again a step function and distances between leaf index Sholl functions can be given by the standard formula (3). The next figure gives an example of a neuron and its associated leaf index Sholl function.

4.9.6. *Total Wiring Descriptor.* "Total wiring" is a morphological feature which measures the total dendritic length of neurons. When used as a Sholl descriptor, it gives total length of dendrites, but also their density as we move away from the soma.

Given a neuron N, let $t\ell(N)$ be the total length of all dendrites of N. If N_r is part of the neuron within a sphere of radius r from the soma (4), then

$$\begin{array}{rcl} TL_N : [0,1] & \longrightarrow & \mathbb{R}^+ \\ \dfrac{r}{R(N)} & \longmapsto & t\ell(N_r) \end{array}$$

This is a normalized Sholl function, which always starts at value 0 and ends up at value $TL_N(1) = t\ell(N)$ which is the total wiring of the neuron. As for other Sholl functions, we will only consider the step function version of this construction, where once more one defines for $r \in [0,1]$,

$$TL_N(r) = t\ell(N_{r_i R(N)}) \ \text{ if } \ r \in [r_i, r_{i+1}[$$

where $0 < r_1 < \ldots < r_n = 1$ are the normalized radial distances of the nodes listed in increasing order.

4.9.7. *Energy Descriptor (Nodal Distribution).* Given a neuron N, consider all its nodes as cloud points in 3D space viewed as charged particles. The charge each node carries will be proportional to the thickness of the branch at that point. These charged nodes affect the space around them through the electric field they generate. This electric field is a well-defined map

$$E_N : \mathbb{R}^3 \setminus \{\text{nodes}\} \longrightarrow \mathbb{R}^3$$

So taking the intensity of the vector field at each point of $\mathbb{R}^3 \setminus \{\text{nodes}\}$ gives us a measure of how space is being affected by the neuron. This also gives a measure of how the nodes are distributed in space as we will later illustrate in the case of Purkinje cells.

Let $\zeta := \{P_1, \ldots, P_n\}$ be the nodes of N, with P_i having charge q_i. This charge q_i is chosen to be the width of the dendrite at point p_i. Each point $P_i(x_i, y_i, z_i)$ of ζ contributes an electric vector field which is normalized to have length q_i and which is of the form

$$E_i(x,y,z) = q_i \left(\frac{x - x_i}{|PP_i|}, \ \frac{y - y_i}{|PP_i|}, \ \frac{z - z_i}{|PP_i|} \right)$$

where $|PP_i| = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}$. By superposition, the node configuration N gives rise to a vector field $E_N(x,y,z) = \sum_i q_i F_i(x,y,z)$ with square intensity

$$|E_N(x,y,z)| \ = \ \left(\sum q_i \frac{x - x_i}{|PP_i|} \right)^2 + \left(\sum q_i \frac{y - y_i}{|PP_i|} \right)^2 + \left(\sum q_i \frac{z - z_i}{|PP_i|} \right)^2$$

Let $O(2)$ be the group of orthogonal matrices. This group acts on $\mathbb{R}^3$, and thus on the set of neurons. If $A \in O(2)$ and $N \subset \mathbb{R}^3$ a neuron, we write $A(N)$ the image of N under this action.

Lemma 4.4. *The intensity at the soma, $|E_N(0,0,0)|$, is $0(2)$-invariant.*

Proof. We show first that $|E_{A(N)}(P)| = |E_N(A^{-1}(P))|$ for $P \in \mathbb{R}^3$. The nodes of N are $\{P_1, \ldots, P_n\}$. Write $E_{A(N)}(P) = \sum E_{A(P_i)}(P)$. Since $E_{A(P_i)}(P) = AE_{P_i}(A^{-1}(P))$, and since A is linear and preserves lengths, it follows that

$$\begin{array}{rcl} |E_{A(N)}(P)|^2 & = & |\sum E_{A(P_i)}(P)| = |\sum AE_{P_i}(A^{-1}(P))| = |A \sum E_{P_i}(A^{-1}(P))| \\ & = & |\sum E_{P_i}(A^{-1}(P))| = |E_N(A^{-1}(P)| \end{array}$$

At the soma $P = (0,0,0)$, $A^{-1}(P) = P$, so that $|E_{A(N)}(P)| = |E_N(P)|$, which is what is claimed. □

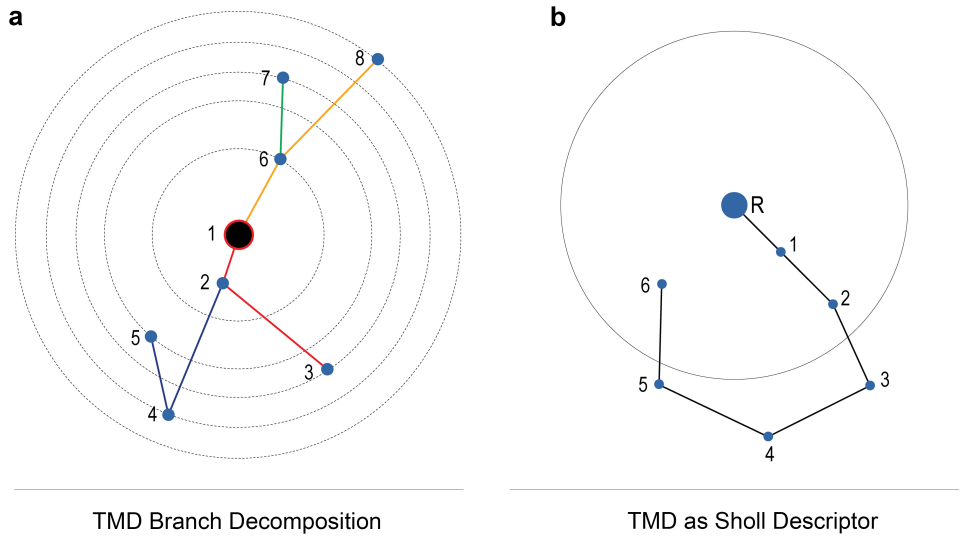

Fig. 7. (**a**) Example of TMD-path decomposition on a simple planar tree. The soma marked with 1 is the root. Equicentered circles reveal the distances of nodes from the root. The furthest node is node 8. The paths from the TMD-path decomposition are: $\{[5,4,2],[3,2,1],[8,6,1],[7,6]\}$ (**b**), The tree T with a single path x starting at the root R. When using TMD as a Sholl-type descriptor by considering TMD of $T \cap B(R,r)$ we will only see the final barcode $[0, d(R,6)]$ for $r \geq d(R,4)$. For the radii r between $d(R,6)$ and $d(R,4)$ the endpoint of the persistence interval will be equal r. When r reaches $d(R,4)$ the endpoint of the persistence interval it will then jump down to $d(R,6)$.

Our Sholl descriptor associates to every neuron the map

$$\begin{array}{rcl} \phi_N : [0,1] & \longrightarrow & \mathbb{R}^+ \\ r & \longmapsto & |E_{N_r}(0,0,0)| \end{array}$$

where N_r is as in (4). This map adds the unit vectors at the soma, one for each node, and takes the magnitude. We can also think of energy as the effect of the nodal distribution around the soma. If all nodes are on one side of a plane going through the soma, then their contributions is greatest (eg. Purkinje cells have very large energy values), as opposed to nodes that are evenly distributed around the soma. In this latter case, several cancellations occur and the energy value tends to be small.

4.9.8. *The Topological Morphological Descriptor (TMD).* Let $T \subset \mathbb{R}^3$ be a tree with a root R. A *path* is any continuous sequence of edges in T. Each path x have unique initial $b(x)$ and terminal $d(x)$ vertices. The TMD is based on a method of decomposing a given tree T into a collection of paths such that the sum of those paths is the whole tree T. In addition, any two paths from that decomposition will either have empty an intersection, or their intersection is the endpoint of one of them (and in this case a branching point of T).

The TMD path decomposition is obtained using the following procedure; All the paths from the TMD-path decomposition starts at the leaves of T. They continue along the tree, towards the root, until they reach a node n of degree 3 or higher in T. In the node n all the paths except from one terminate. The path that continues through the node n is the one with the initial node further away from R (soma)[1]. Once a path reaches the root R, it does not continue any further (it terminates). For example, a TMD-path decomposition is presented in Figure 7a.

Given a TMD-path decomposition as described below, we associate a collection of pairs of numbers inspired by *persistent diagram*, to this decomposition. For that purpose, a path x having the initial and terminal vertices $b(x)$ and $d(x)$ correspond to persistence interval $[d(R,b(x)), d(R,d(x))]$. Using the terminology from persistent homology, we say that the path x is *born* at the radius $d(R,b(x))$ and *dies* at the radius $d(R,d(x))$. The collection of all such birth-death pairs is then used as a signature of a tree. As it has the same structure as persistent diagram, we further adopt various metrics from persistent homology to compare such diagrams.

To fit the TMD into the scheme of the current paper we will now turn it into a Sholl descriptor having values in the space of persistence diagrams. For that purpose let T_r be the connected component of $T \cap B(R,r)$ containing R. Let us make two simple observations:

[1]In case of two or more paths satisfying this condition, the one that continue is picked up randomly.

(1) Note that the TMD-path decomposition of T restricted to T_r is a valid TMD-path decomposition of T_r. To see that, let us consider a branching node n in T, such that $d(R,n) \leq r$, and the same node in T_r. The paths from n to the leaves in T_r will either be the same as in T, or they will be cut short in T_r by $B(R,r)$. In both cases, the path that does not terminate at n in T will also be the path with the initial point further away from R in T_r. It is possible that more than one path in T_r joining in n will be cut short by $B(R,r)$. However in this case we can choose to continue in T_r the same path that continues in T. Consequently, a TMD-path decomposition in T_r can be obtained by appropriate restriction of the TMD-path decomposition in T.

(2) Suppose we consider a path x in T giving rise to the persistence interval $[d(R,b(x)), d(R,d(x))]$. Then the path x will be present in T_r for $r \geq d(R,d(x))$. However it may happen that the path x contains points that are further away from R than $d(R,b(x))$ and will be cut in those points by $B(R,r)$. This will happen when the path x turns around as presented in the Figure 7b. In that instance, the interval $[d(R,b(x)), d(R,d(x))]$ in T_r for certain values of r will have a larger value of the first coordinate than the corresponding interval in T. Therefore, while the first (birth) coordinate of the interval corresponding to x in T_r may be longer than the interval corresponding to x in T. The actual length can be obtained from the coordinates of degree-2 vertices in x.

Those two observations allows for quick computation of the Sholl version of TMD descriptor, i.e. a TMD of a tree T_r, also denoted as $TMD(T,r)$. Firstly the TMD-path decomposition P of T is computed. Subsequently, for a given radius r, a subset $P' \subset P$ containing all the paths x such that $d(x) \leq r$ is selected. The paths in $x \in P'$ are transversed to find the point f_x in there which is inside $B(R,r)$ and furthest away from its center. Once found, the pair $(d(R,f_x), d(x))$ is added to the $TMD(T,r)$.

The algorithm described above uses the radial distance from the soma to construct T_r. When an intrinsic distance is used instead both in TMD and in construction of T_r, the Sholl version of the descriptor is even easier to obtain, as each path $x \in P$ such that $d(x) \leq r$ will give rise to a pair $(d(x), \min(b(x), r))$ in $TMD(T,r)$.

Unlike other Sholl descriptors, $TMD(T,r)$ has the range in the space of persistence diagrams which is a much richer mathematical structure than real numbers. Yet, it is still possible to compute distances between the functions $TMD(T,r)$ and $TMD(T',r)$. Let us assume that both functions has been computed for a discrete set of values $0 = r_0 < r_1 < r_2 < \ldots < r_n$. Then a distance between $TMD(T,r)$ and $TMD(T',r)$ can be approximated by:

$$\sum_{i=0}^{n-1} (r_i - r_{i-1})\ d_{diag}(TMD(T,r_i), TMD(T',r_i))$$

where d_{diag} denotes any distance between persistence diagrams, e.g. p-Wasserstein distance.

4.10. **Stability.** In this subsection we define stability and then verify that all Sholl descriptors are stable. We address this issue by verifying that our descriptors are reasonably sensitive to small perturbations of input neurons. More precisely, if two reconstructions of the same neuron vary slightly, they will result in different tree representations. A descriptor is "stable" if, when applied to either tree, it gives results that also vary slightly (i.e. the variation is controlled).

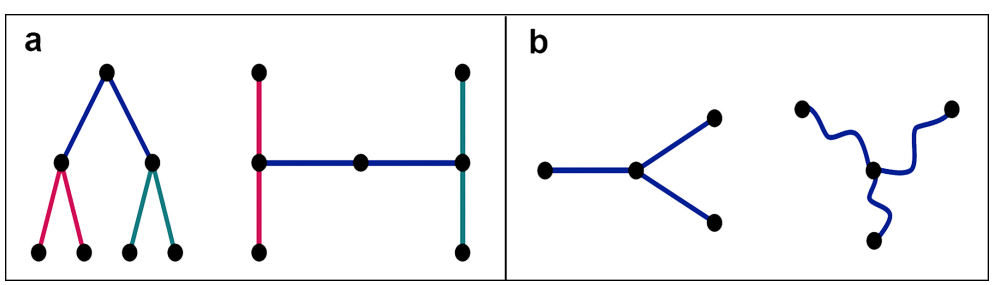

Fig. 8. Representative isomorphic trees with entirely different (**a**) branching pattern and (**b**) tortuosity descriptors

.

Two different reconstructions of the same neuron produce two different trees embedded in $\mathbb{R}^3$. All reasonable reconstruction schemes should produce isomorphic trees, resulting in the same number of primary dendrites and the same number of bifurcations. We can measure the distance between two reconstructed trees under the Hausdorff metric and use it as a measure of closeness. Two such reconstructions are expected to be close in the Hausdorff metric, requiring that our descriptor depends "continuously" on this metric. However, this is not a good notion, as trees that are very close in the Hausdorff metric may still have very different morphological properties (like lengths

of branches, number of nodes, etc). Figure 9 gives an illustration of two trees (**b**) and (**c**) close, in a Hausdorff sense, to the initial tree (**a**) note the tree in (**a**) is represented in gray and overlapping with the tree in (**b**), and is depicted below the main branches in (**c**). Clearly the trees (**b**) and (**c**) have distinct morphological features compared to the tree (**a**) and they should not be considered similar.

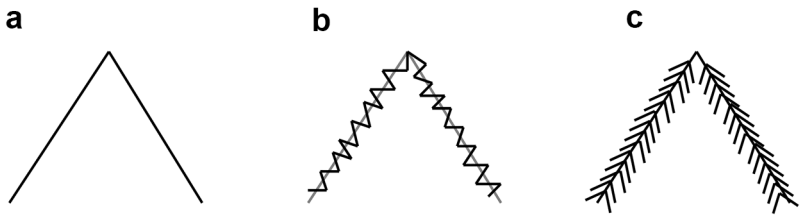

Fig. 9. Representative tree (**a**) and similar trees (**b**) and (**c**) that are close to tree (**a**) in the Hausdorff metric.

Our next definition is adapted from ([45], §2) who utilizes it for rectifiable curves and in the context of knot theory. We will assume that the dendrites are piecewise smooth paths in $\mathbb{R}^3$; meaning that the branches between nodes can be parameterized as C^1-differentiable paths in space.

We say that two neurons N and N' are (δ, θ)–close if N' can be obtained from N by a smooth 1-1 map Ψ supported on an open neighborhood of N so that corresponding points x and $\Psi(x)$ are within δ and the norm differences $||v - d\Psi_x(v)|| < \theta$ for all $x \in U$ and $v \in T_x\mathbb{R}^3$, where $d\Psi_x$ is the differential of Ψ at x. We recall this is a linear map between tangent spaces $d\Psi_x : T_x\mathbb{R}^3 \to T_{\Psi(x)}\mathbb{R}^3$ mapping a vector v to $Jac_x(\Psi)(v)$, where $Jac_x(\Psi)$ is the 3×3 Jacobian matrix of partial derivatives evaluated at x. Let's now make this construction a bit more precise. For the sake of simplicity we will assume Ψ is defined on all of $\mathbb{R}^3$.

Definition 4.5. We say that N and N' are (δ, θ)-close if there exists an ambiant diffeomorphism $\Psi : \mathbb{R}^3 \to \mathbb{R}^3$ such that $|\Psi(x) - x| < \delta$ for all x, and the Frobenius norm $||I - Jac_x\Psi|| < \theta$ for every $x \in \mathbb{R}^3$. In contrast with the definition in [45], we not only require the angles between corresponding vectors to be close, but also their norms. This is precisely the essence of the inequality $||I - Jac_x\Psi|| < \theta$.

Remark 4.6. Let's just observe that our definition is related to the C^1-topology of functions in the following way. If we view a branch γ of N as a smooth path $[0, 1] \to \mathbb{R}^3$, then it is (δ, θ)-close to $\Psi \circ \gamma$ if both paths are C^1-close. The definition of C^1-closeness doesn't involve the existence of Ψ, so its easier, but it cannot be defined globally on neurons, as opposed to just branches, since neuronal trees are not manifolds.

We now define "stability".

Definition 4.7. A Sholl descriptor ϕ is *stable* if for any $\epsilon > 0$, there exists $\eta > 0$ so that for $\delta < \eta, \theta < \eta$,

$$N \text{ and } N' \text{ are } (\delta, \theta)\text{-close} \implies d_\phi(N, N') < \epsilon$$

According to this definition, a small perturbation or deformation of the neuron which "moves the points by as little as δ" and "distorts the branches by as little as θ", yields a small change in the descriptor ϕ.

Let N be a neuron represented as a spatial tree, and let ϕ be a Sholl descriptor. The nodes for N are sorted according to increasing distances from the soma

$$0 < r_1 < \cdots < r_k \leq 1$$

These distances are radial or dendritic depending on the descriptor. We make the assumption that a deformation of a neuron does not introduce new bifurcations, and so the leaf index is completely unchanged by deformation. It is evidently stable.

We start by verifying the stability of the branching pattern descriptor. This descriptor is only based on the distribution of nodes, and so only the δ constant matters. Let's now move the nodes of N by a distance δ, and by that we mean there is a homeomorphism $\Psi : N \to N'$, taking node P_i to P'_i so that for all i, $|r_i - r'_i| \leq \delta$. We ensure that $\delta < \min_i |r_i - r_{i+1}|$ so that the formula for functional distance takes the form (3)

$$d_\phi(N, N') = d^{L^1}(\phi_N, \phi_{N'}) = \sum_{i=0} (t_{i+1} - t_i)|\phi_N(t_i) - \phi_{N'}(t_i)|$$

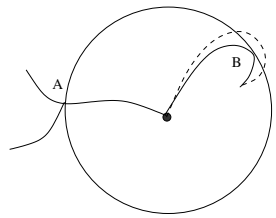

Fig. 10. Instability behavior

.

where $(t_i, t_{i-1}) = (r_j, r'_j)$ or (r'_j, r'_{j-1}) or (r_j, r_{j-1}) for some j. In the first case, $|\phi_N(t_i) - \phi_{N'}(t_i)| = 1$, and in the other two cases it is 0. This means that

$$d_\phi(N, N') \leq \sum_{i=1} |r_i - r'_i| = k\delta$$

where k is the number of nodes. By choosing δ small enough, this is less than any desired ϵ.

The taper and energy descriptors depend only on the node distribution, and are thus stable. Note that for the taper rate, we assume that the width of a dendrite at a given node is the same in any reconstruction (this is not a varying feature in our definition of stability), so this descriptor only depends on nodes, and it is stable.

To see stability of the TMD with the radial distance to the soma let us observe that the TMD-path decomposition may change when the position of nodes is perturbed. This will happen when the endpoints of two paths that merge in a bifurcation point b are at almost the same distance from the soma. In this case a perturbation of the endpoints of those paths may result in swapping the branch that continues up from b with the one that terminates there. However, since the TMD only gathers the values of distances from the soma, the endpoints of persistence intervals will move by at most δ which directly translate in stability of the descriptor.

As for the Tortuosity descriptor, we recall that T associates to every N and $r \in [0, 1]$ the average tortuosity of N_r which is the connected component of N containing the soma inside the ball of radius r. The stability for T holds under one condition, and so we refer to it as "conditional stability". We discuss this condition next and remark that it is almost always realized so that in practice and generically T behaves stably. To understand this condition, we must observe that instability can occur in the following situation illustrated in Figure 10.

Here r is chosen to be the radial distance of the node A. The twisted dendrite touches the sphere tangentially at B. When measuring tortuosity of N_r, we only consider the term δ_{SB} (5) which is the tortuosity from the soma S to B considered as a leaf. If a small perturbation causes the branch to be inside the sphere, this term is replaced by the term δ_{SC} which is the tortuosity of the entire branch inside $B(r)$ from S to C. This leads to a sudden increase in tortuosity which can potentially lead to instability. However this only happens if indeed a sphere through a node is tangent to a branch, which is a rare instance.

Let $\gamma : [0, 1] \longrightarrow \mathbb{R}^3$ be a smooth space curve. Then its length is given by $\int_0^1 |\gamma'(t)|dt$. We say that the two paths γ_1 and γ_2 are (δ, θ) close if there is a (δ, θ)-diffeomorphism taking γ_1 to γ_2; that is $\gamma_2 := \Psi \circ \gamma_1$.

Lemma 4.8. *Let γ_1 be a given smooth curve in $\mathbb{R}^3$. For every $\epsilon > 0$, $\exists \eta$ with $\delta < \eta, \theta < \eta$, such that for every curve γ_2 that is (δ, θ)-close to γ_1,*

$$\left| \int_0^1 |\gamma'_2(t)| - |\gamma'_1(t)|dt \right| < \epsilon$$

Proof. Let γ_2 be (δ, θ)-close to γ_1, meaning there is a (δ, θ)-diffeomorphisme Ψ taking γ_1 to γ_2. Since γ_1 is differentiable, $|\gamma'_1(t)|$ is bounded uniformly on $[0, 1]$, say by $M > 0$. We can write

$$\begin{aligned}
\left| \int_0^1 |\gamma'_1(t)| - |\gamma'_2(t)|dt \right| &\leq \int_0^1 ||\gamma'_1(t)| - |\gamma'_2(t)||dt \\
&\leq \int_0^1 |\gamma'_1(t) - \gamma'_2(t)|dt \\
&\leq \int_0^1 |\gamma'_1(t) - Jac_t\Psi(\gamma'_1(t)|dt \quad \text{here } Jac_t\Psi := Jac_{\gamma_1(t)}\Psi \\
&= \int_0^1 |(I - Jac_t\Psi)(\gamma'_1(t)|\, dt \\
&\leq M \int_0^1 ||I - Jac_t\Psi||dt \leq M\vartheta
\end{aligned}$$

where we know by definition that $||I - Jac_t\Psi|| \leq \theta$ on the domain of Ψ. By choosing $\eta = \frac{\epsilon}{M}$, we obtain our claim. □

Let Ψ be (δ, θ)-diffeomorphism taking N to N'. The map Ψ maps nodes to nodes, branches to branches necessarily. Lemma 4.8 shows that by controlling (δ, θ) we can control the lengths of branches. It is clear that T is stable away from the stated condition, meaning that if spheres through the nodes of N and N' are not tangent to dendrites, then $d_T(N, N') = d_T(N, \Psi(N)) < \epsilon$ for any chosen $\epsilon > 0$, once δ, θ are chosen sufficiently small.

Finally we discuss the stability of the flux descriptor. The stability in these cases hinges on controlling the variation of angles in any neuron deformation. This is a direct consequence of the following lemma.

Lemma 4.9. *Let $v_1, v_2 \in T_x\mathbb{R}^3$ be unit vectors, $x \in N$ (x will be typically a node in our case). Then for any $\epsilon > 0$, $\exists \eta > 0$ such that for any (δ, θ)-diffeomorphism Ψ with $\theta < \eta$,*

$$|\angle(v_1, v_2) - \angle(d\Psi_x(v_1), d\Psi_x(v_2))| < \epsilon$$

Proof. We fix x and drop it from the notation. We write $d_u\Psi(v_i) = \dfrac{d\Psi(v_i)}{|d\Psi(v_i)|}$ the normalized vector. Using the cosine-angle formula $|v_1 - v_2|^2 = |v_1|^2 + |v_2|^2 - 2|v_1||v_2|\cos\angle(v_1, v_2)$, we can write

$$|d_u\Psi(v_1) - d_u\Psi(v_2)|^2 = 2 - 2\cos\angle(d_u\Psi(v_1), d_u\Psi(v_2))$$

$$|v_1 - v_2|^2 = 2 - 2\cos\angle(v_1, v_2)$$

By taking the difference, we see immediately that we can make the angles $\angle(v_1, v_2)$ and $\angle(d\Psi_x(v_1), d\Psi_x(v_2))$ arbitrarily close by making their cosines arbitrarily close or equivalently by making $|d_u\Psi(v_1) - d_u\Psi(v_2)|$ and $|v_1 - v_2|$ arbitrarily close. We check this last part.

$$\begin{aligned}
& ||d_u\Psi(v_1) - d_u\Psi(v_2)| - |v_1 - v_2|| \\
\leq\ & |d_u\Psi(v_1) - v_1| + |d_u\Psi(v_2) - v_2| \\
=\ & \left|d\Psi\left(\frac{v_1}{|d\Psi(v_1)|}\right) - v_1\right| + \left|d\Psi\left(\frac{v_2}{|d\Psi(v_2)|}\right) - v_2\right| \quad \text{by linearity of } d\Psi \\
\leq\ & \left|d\Psi\left(\frac{v_1}{|d\Psi(v_1)|}\right) - \frac{v_1}{|d\Psi(v_1)|}\right| + \left|d\Psi\left(\frac{v_2}{|d\Psi(v_2)|}\right) - \frac{v_2}{|d\Psi(v_2)|}\right| + \left|\frac{v_1}{|d\Psi(v_1)|} - v_1\right| + \left|\frac{v_2}{|d\Psi(v_2)|} - v_2\right|
\end{aligned}$$

Since for any vector $v \in T_x\mathbb{R}^3$, $|d\Psi(v) - v| < \theta$, we can extend the string of inequalities above to

$$\begin{aligned}
||d_u\Psi(v_1) - d_u\Psi(v_2)| - |v_1 - v_2|| &\leq 2\theta + \frac{|1 - |d\Psi(v_1)||}{|d\Psi(v_1)|} + \frac{|1 - |d\Psi(v_2)||}{|d\Psi(v_2)|} \\
&\leq 2\theta + 2\frac{\theta}{1 - \theta}
\end{aligned}$$

By making θ small, the left term can be made arbitrarily small and this is enough to yield our claim. □

4.11. **The Detection Algorithm.** The detection construction and motivation was discussed in §4.7. Supplementary Fig. 11 illustrates this construction on three classes labeled A, B and C.

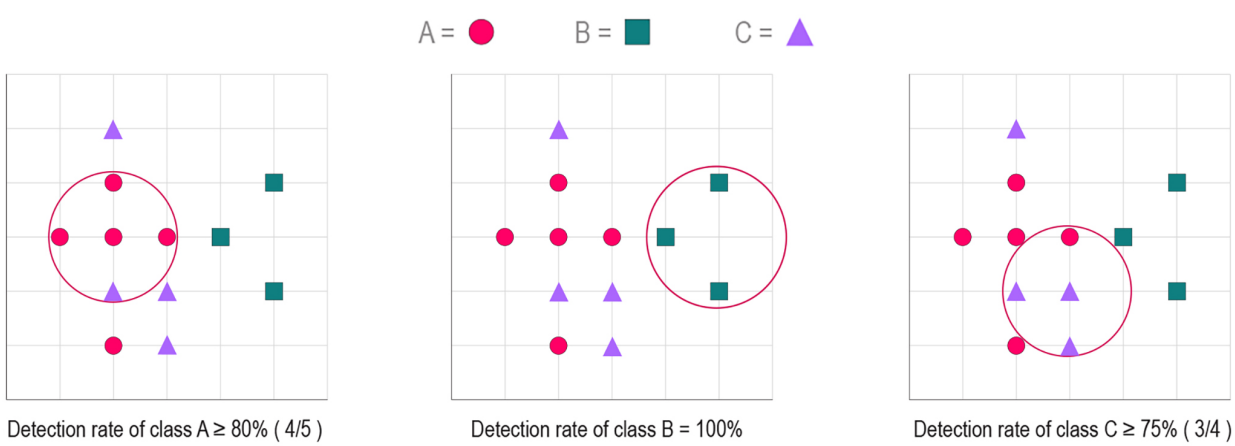

Fig. 11. Method used to determine detection rate. Each circle is the boundary of a disk in the Euclidean metric.

In this section we provide a simple algorithm to estimate the level of detection of a Sholl descriptor ϕ on a set of neurons divided into classes $C_1, \ldots, C_n$. Recall that associated to ϕ we have a metric d_ϕ on the set of neurons (1). To compute the level of detection of C_1, we proceed as follows:

(1) Assume $C_1 = \{N_1, N_2, \ldots, N_m\}$ has m neurons.
(2) Pick a neuron N_i from class C_1, one at a time, for $i \in \{1, \ldots, m\}$.
(3) Look at all distances d_ϕ from N_i to all neurons in all the classes. Divide the obtained distances into two sets: distances of N_i to all neurons in C_1 (these are called the internal distances), and distances from N_i to all other neurons not in C_1 (that is to C_2 up to C_n, and these are called the external distances).
(4) Order all internal distances increasingly $d_1 = d_\phi(N_1, N_1) = 0 < d_2 < ... < d_m$.
(5) Now proceed iteratively: let $\epsilon = d_m$ and consider all external distances smaller than d_m. Count them, say there are c_m of them. Then we have two ratios:
$m/m = 1 \leftrightarrow 100\%$ (meaning all neurons in C_1 are within distance d_m from N_1 which is of course the case), and $\frac{m}{c_m+m}$ (this gives the total percentage of neurons from the class C_1 among all neurons that are a distance d_m from N_1).
In this case, the smaller of the two is $\beta_m = \frac{m}{c_m+m}$.
(6) Consider next the ball of radius d_{m-1} centered at N_1 and let c_{m-1} be the number of external neurons that are within that ball. We set β_{m-1} to be the smallest of $\frac{m-1}{m}$ and $\frac{m-1}{c_{m-1}+m-1}$.
(7) Proceed iteratively the same way through $\epsilon = d_{m-2}, ..., d_2$ and get rates $\beta_{m-2}, \ldots, \beta_2$. One defines

$$\beta(N_1) = \min\{\beta_2, \ldots, \beta_m\}$$

(8) Repeat the same procedure for $N_2, \ldots, N_m$ and obtain the proportions $\beta(N_i)$ for $1 \leq i \leq m$. The level of detection of C_1 by the descriptor ϕ is then simply

$$\det_\phi(C_1) = \max\{\beta(N_1), \ldots, \beta(N_m)\} \times 100\%$$

For instance, if we obtain $\det_\phi(C_1) \geq 75\%$, this implies there is a ball in the d_ϕ metric that contains at least 75% of all the internal neurons in C_1, and within that ball, at least 75% of all neurons are from C_1. A detection rate of 100% means perfect detection whereby a ball contains all of C_1 and no other neurons from the other classes $C_2, \ldots, C_m$.

4.12. **Combination of Descriptors and Classification.** Our objective is to build a toolbox of distances $d_1, \ldots, d_n$ that can be used to discriminate among trees (classes of neurons) according to a given morphological feature. One aim is to understand, for two or more classes of trees, which morphological features differentiate them. For example, C_1 can be a class that represents neurons from an experimental group with a neurological disease, while C_2 is a class of neurons from a control group. In this regard, one may want to know which morphological features are different between these two classes.

4.13. **Vectorization: Unsupervized Classification.** The starting point is a set $C = \{N_1, \ldots, N_k\}$ of neurons. Given a descriptor ϕ and a neuron N, consider its area value $a(\phi_N) = \int_0^1 |\phi_N(r)|dr$. The extreme values of ϕ_N are meaningful. We will assume ϕ is one of the descriptors below:

- Branching: Then $\phi_N(1)$ is the (opposite) of the number of primary branches.
- Tortuosity: then $\phi_N(1)$ is the average tortuosity of N.
- Leaf index: then $\phi_N(0)$ is the total number of leaves of N.
- Energy (respectively flux and wiring): then $\phi_N(1)$ is respectively the total energy vector at the soma (respectively flux and total wiring of N).

We can list all of our descriptors $\phi_1, \ldots, \phi_9$, ϕ_1 being leaf index, and associate to N the vector

$$V_N = \langle (a(\phi_N^1), \phi_N^1(0)), (a(\phi_N^2), \phi_N^2(1)), \ldots, (a(\phi_N^9), \phi_N^9(1)) \rangle \ \in \mathbb{R}^{18}$$

If a descriptor data is not available (eg. taper rate), then the corresponding pair of entries is omitted from the vector. For Dataset 1, we vectorized based on seven descriptors (all but TMD-Sholl and taper rate).

4.14. **Combination of Metrics.** Here we present a greedy grid–based search procedure that can reveal which features are 'fundamentally' different between classes. For that purpose, we consider a new distance d being the following linear combination:

$$d = \alpha_1 \ d_1 + \alpha_2 \ d_2 + \ldots + \alpha_n \ d_n$$

We will consider the constants α_i sampled from a uniform grid. For the fixed choice of $\alpha_1, \ldots, \alpha_n$ let $I_k = \max_{x,y \in C_k} d(x,y)$ be the maximal distance d between objects in C_k for $k \in \{1,2\}$. Let $E = \{\min_{x \in C_1, y \in C_2} d(x,y)\}$ be the minimal distance d between elements from two different classes. We then consider the ratio

$$sc_{\alpha_1,\ldots,\alpha_n} = \frac{E}{I_1 + I_2}$$

and select $\alpha_1, \ldots, \alpha_n$ from our grid of points that maximize $sc_{\alpha_1,\ldots,\alpha_n}$. Note that this greedy grid search has exponential complexity with n, being the number of considered distances.

The obtained weights $\alpha_1, \ldots, \alpha_n$ give an idea of the relative importance of different distances in the separation of classes C_1 and C_2. consequently, when distances can be interpreted geometrically, the weights may help identify the geometrical features that are important in the separation of the classes, and those that are not.

This idea can be generalized for a multi–class problem. There are various ways this can be achieved. However, we will only present one approach. Let us have k classes $C_1, \ldots, C_k$ and define $I_j = \max_{x,y \in C_j} d(x,y)$ (maximal internal distance in C_j) and $E_{i,j} = \{\min_{x \in C_i, y \in C_j} d(x,y)\}$ (minimal distance between classess C_i and C_j). Then the multi-class score is given by:

$$sc_{\alpha_1,\ldots,\alpha_n} = \frac{\max\{E_{i,j}, i,j \in \{1,\ldots,k\}\}}{\sum_{i=1}^{k} I_i}$$

The remaining of the procedure described above is not changed. We can show that separation is meaningful by conducting a "permutation test" (see §4.16).

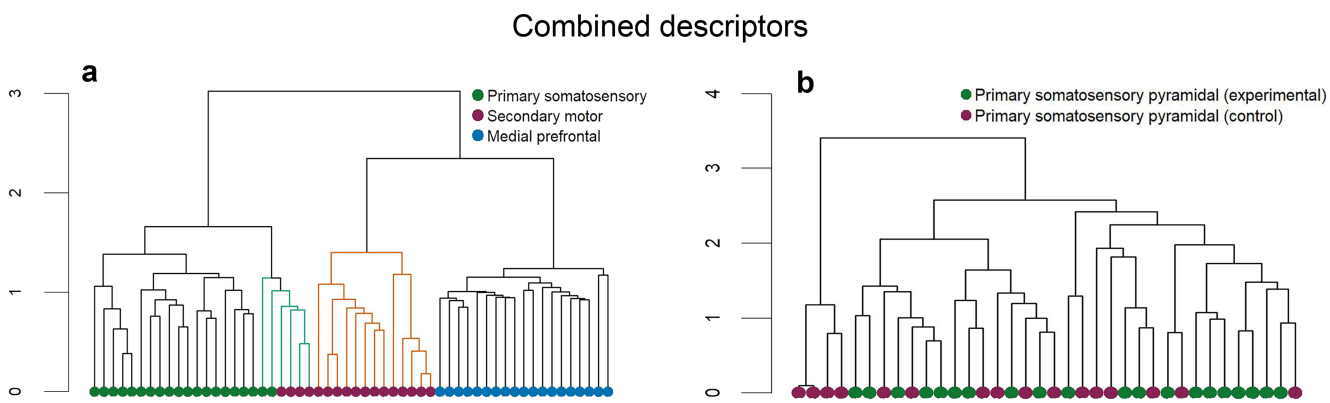

Fig. 12. Representative dendrograms based on the combination of metrics.(**a**) Combined dendrogram for dataset 4 and (**b**) dataset 5.

4.15. **Metric Learning: Supervised Classification.** Starting with classes of neurons $C_1, \ldots, C_k$, a Sholl feature ϕ, and a given random neuron N, we wish to know how close in feature is N to any of these classes, or in other words, how much of the feature ϕ does N share with any of these classes?

More generally, given classes $C_1, \ldots, C_k$, a neuron N and a number of features $\phi_1, \ldots, \phi_m$, we can use the combined effect of the features to "classify" the C_i's. Our approach is to devise an optimal metric $D_{\phi_1,\ldots,\phi_m}$ such that N is close to the class C_i if its distance under this metric to this class is small (relative to other distances to other classes). This is accomplished via the relatively recent technique of *Metric Learning* (see [44]) and machine-implemented in [30]. We explain how this is achieved and successfully used to solve the following two problems:

(i) Under the metric learned $D_{ML} = D_{\phi_1,\ldots,\phi_m}$, classes that share the largest number of features (i.e. are within shorter Sholl distances) are closer than those that share fewer or none.
(ii) Given a neuron picked from outside our dataset $\mathcal{C} = C_1 \cup \cdots \cup C_k$, the metric-learned D_{ML} determines with good confidence in which suitable class this neuron fits.

The starting point is a set of vectors $v_i \in \mathbb{R}^n$, each of which is labeled by an integer ℓ_i. Ideally one would hope that the Euclidean metric $d(v_i, v_j) = \sqrt{v_i^T v_j}$ does "separate" the classes, meaning that vectors in the same class are close and those in different classes remain relatively distant. This is hardly the case in practice, so one seeks a modification of this Euclidean metric which has this separation property. A standard approach is to introduce a "matrix of weights" M, which is $n \times n$, positive, so that $D_{ML}(v_i, v_j) = \sqrt{v_i^T M v_j}$ defines a new metric on $\mathbb{R}^n$ (so called Mahalanobis metric) with better separating properties with respect to the chosen classes. More precisely, D_{ML} maximizes the sum of distances between points with different labels while keeping the sum of distances between those with similar labels small. Note that since M can be written as LL^T, the associated D_{ML} metric has the following

interpretation: it is the distance obtained by first moving vectors via L in $\mathbb{R}^n$, then taking their Euclidean distance. Various machine learning algorithms have been implemented to find this optimal M. This approach is entirely supervised since we need the classes to train the matrix entries and thus the metric.

In the context of this paper, the given data are neurons distributed among chosen classes, say $C_1, C_2, \ldots, C_k$. Typically, a class C will comprise neurons from a particular region in the brain, an experimental condition, or developmental stage. Therefore, all neurons in that class share a desired property. Let $\Omega = \phi_1, \ldots, \phi_m$ be a family of Sholl descriptors. For $\phi \in \Omega$, and $N \in \mathcal{C} := C_1 \cup C_2 \cup \cdots \cup C_k$, we can consider the average distance of N to C_i for the descriptor ϕ_j given by

$$a_{\phi_j,i}(N) = \frac{1}{|C_i|} \sum_{N' \in C_i} d_{\phi_j}(N, N')$$

As ϕ runs over Ω, we obtain the vector in $\mathbb{R}^{mk}$ (here $n = mk$)

$$v(N) := (a_{\phi_1,1}(N), \ldots, a_{\phi_1,k}(N), a_{\phi_2,1}(N), \ldots, a_{\phi_2,k}(N), \ldots, a_{\phi_m,1}(N), \ldots, a_{\phi_m,k}(N)) \tag{7}$$

where the first k entries give the average distances of N to all k classes in the metric d_{ϕ_1}, and the last k entries give the average distances of N to all k classes in the metric d_{ϕ_m}. Notice that each neuron $N \in \mathcal{C}$ comes with a unique label i if $N \in C_i$. The vector in (7) depends on the ordering on the C_i's and ϕ_j's, but the final outcome will not. In all cases, and for the remaining constructions in this section, an order on classes $C_1, \ldots, C_k$ and on descriptors $\phi_1, \ldots, \phi_m$ is always chosen before we start running any algorithm, and this order is preserved throughout the process.

Starting with a dataset of neurons, and given m descriptors, we obtain a set of labeled vectors in $\mathbb{R}^{mk}$, one for each neuron. The class C_i of neurons will correspond to a class C'_i of vectors. The labels are in $\{1, 2, \ldots, k\}$. This setup is precisely what metric learning (ML) requires for implementation, and data can be run through a supervised ML algorithm [30]. The end result is a metric D_{ML}, depending on $d_{\phi_1}, \ldots, d_{\phi_k}$, which differentiates between the classes $C'_1, \ldots, C'_k$.

The solutions to the two problems (i) and (ii) raised at the beginning of this section are now evident. They are summarized below.

The Classification Scheme: Start with m classes $C_1, \ldots, C_k$ and k descriptors $\phi, \ldots, \phi_m$.

- Run metric learning on the vectorized classes to obtain a new metric D_{ML}. The new metric is validated after being tested for "overfitting" (see §4.16). A good metric gives good separation of the vectorized classes.
- Given a neuron N, vectorize it as in (7) and then take its average distances to the classes C'_i as in $D_{ML}(N, C'_1), \ldots, D_{ML}(C'_k)$. These distances are compared.
- The random neuron N shares these given features the most with a particular class C_j if $D_{ML}(N, C'_j)$ is smallest among $D_{ML}(N, C'_1)$,..., $D_{ML}(C'_k)$. This can be phrased in terms of percentages.
- (feature selection) Run each descriptor on the classes. If the detection rates are lower than 80% on all classes, the descriptor can be considered "noisy" and is subsequently excluded. Repeat the process above with non noisy descriptors.

4.16. **Overfitting.** Both grid search §4.14 and metric learning §4.15 provide efficient tools to differentiate classes of neurons. Yet, the fact that these methods return clear separation between two or more classes of trees is not sufficient to conclude that the separation is geometrically meaningful.

Example 4.10. Let us consider four vertices of a square: $A = (1, 1)$, $B = (-1, 1)$, $C = (-1, -1)$ and $D = (1, -1)$. Suppose that the first class consists of points A and D, while the second class is composed of points B and C. Metric learning will seek to place elements of different classes far away and those of the same class close together. This can be achieved by a metric function $d((x_1, y_1), (x_2, y_2)) = \sqrt{A(x_1 - x_2)^2 + a(y_1 - y_2)^2}$ by making A large and a small. Such a perturbed Euclidean metric can be obtained both by the grid search and metric learning. Yet, it is clear that the division to the first and second class is somewhat arbitrary; In fact, putting points A and B to the first class and C and D to the second one is geometrically similar. Alternatively, separation of these classes can be achieved by the same distance function with A small and a large, and therefore can also be found by the methods we present here.

In order to detect cases similar to the one described in Example 4.10, we will use a procedure that is similar to a *permutation test*. Namely, after obtaining separation of the given classes, we will repetitively permute all the labels of data points and run the grid search / metric learning for the data with the permuted labels. We will check how frequently a good separation between the permuted labels is obtained. If that happens often, then the separation between the initial classes is not valid. However, of it is not the case, then we have additional verification that the separation of the original classes is meaningful.

4.17. **Software.** MATLAB v2019a (The Mathworks Inc., Natick,MA. RRID: SCR_001622), Python v3.7 (Python Software Foundation. RRID:SCR_008394) and R v4.0.3 (R Foundation for Statistical Computing. RRID:SCR_010279) were used for computations. Code for the data analysis is available at https://github.com/reemkhalilneurolab/morphology

5. Code availability

Code for the data analysis was deposited to the Github repository and is available at https://github.com/reemkhalilneurolab/morphology

6. Funding

This work is supported by grants from the Biosciences and Bioengineering Research Institute (BBRI) and Faculty Research Grant (FRG), American University of Sharjah (AUS). P.D. acknowledges the support of Dioscuri program initiated by the Max Planck Society, jointly managed with the National Science Centre (Poland), and mutually funded by the Polish Ministry of Science and Higher Education and the German Federal Ministry of Education and Research.

7. Acknowledgements

We thank M. Rowaizak for his help with figure 1 and 2 illustrations.

8. Competing interests

The authors declare no competing financial interests.

9. Author contributions

S.K, R.K. and A.F. conceptualized the project; S.K, A.F., and P.D designed the computational analysis; P.D. and A.F. performed all the coding. S.K, R.K., P.D and A.F. interpreted the results; R.K. and S.K wrote the initial draft paper; S.K, R.K., P.D and A.F. revised the initial draft and wrote the final paper.

References

[1] Vetter P, Roth A, Hausser M., *Propagation of action potentials in dendrites depends on dendritic morphology.*,J Neurophysiol **85** (2001),926–937 .

[2] Brette R. et al. *Simulation of networks of spiking neurons: A review of tools and strategies.* , J Comput Neurosci **23** (2007), 349–398.

[3] DeFelipe J, Fariñas I, *The pyramidal neuron of the cerebral cortex: morphological and chemical characteristics of the synaptic inputs* , Prog Neurobiol **39(6)** (1992), 563-607.

[4] Lund JS, Yoshioka T, Levitt JB, *Comparison of intrinsic connectivity in different areas of macaque monkey cerebral cortex* , Cereb Cortex **3** (1993), 148–162.

[5] Elston GN, Rosa MGP, *The occipitoparietal pathway of the macaque monkey: comparison of pyramidal cell morphology in layer III of functionally related cortical visual areas* , Cereb Cortex **7** (1997), 432–452.

[6] Elston GN, Rosa MGP, *Basal Pyramidal cells, patches, and cortical columns: a comparative study of infragranular neurons in TEO, TE, and the superior temporal polysensory area of the macaque monkey* , J Neurosci **20(RC117)** (2000), 1–5.

[7] Luebke, J. I., Medalla, M., Amatrudo, J. M., Weaver, C. M., Crimins, J. L., Hunt, B, *Age-related changes to layer 3 pyramidal cells in the rhesus monkey visual cortex* , Cereb. Cortex **25** (2015), 1454–1468.

[8] Elston GN, Rosa MGP, *Morphological variation of layer III pyramidal neurons in the occipitotemporal pathway of the macaque monkey visual cortex* , Cereb Cortex **8** (1998), 278–294.

[9] Elston GN, *cognition and the cell: new insights into the pyramidal neuron and prefrontal function* , Cereb Cortex **13(11)** (2003), 1124-1138.

[10] Beaulieu C. (1993). *Numerical data on neocortical neurons in adult rat, with special reference to the GABA population.* Brain research, **609(1-2)**, 284–292. https://doi.org/10.1016/0006-8993(93)90884-p

[11] Gabbott, P. L., Bacon, S. J. (1996). *Local circuit neurons in the medial prefrontal cortex (areas 24a,b,c, 25 and 32) in the monkey: II. Quantitative areal and laminar distributions.* The Journal of comparative neurology, **364(4)**, 609–636. https://doi.org/10.1002/(SICI)1096-9861(19960122)364:4¡609::AID-CNE2¿3.0.CO;2-7

[12] Petilla Interneuron Nomenclature Group, Ascoli, G. A., Alonso-Nanclares, L., Anderson, S. A., Barrionuevo, G., Benavides-Piccione, R., Burkhalter, A., Buzsáki, G., Cauli, B., Defelipe, J., Fairén, A., Feldmeyer, D., Fishell, G., Fregnac, Y., Freund, T. F., Gardner, D., Gardner, E. P., Goldberg, J. H., Helmstaedter, M., Hestrin, S., Yuste, R. (2008). *Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex.* Nature reviews. Neuroscience, **9(7)**, 557–568. https://doi.org/10.1038/nrn2402

[13] DeFelipe J, López-Cruz PL, Benavides-Piccione R, Bielza C, Larrañaga P, Anderson S, *New insights into the classification and nomenclature of cortical GABAergic interneurons* , Nat Rev Neurosci **14(3)** (2013), 202:16.

[14] Hrvoj-Mihic, B., Hanson, K. L., Lew, C. H., Stefanacci, L., Jacobs, B., Bellugi, U., & Semendeferi, K, *Basal Dendritic Morphology of Cortical Pyramidal Neurons in Williams Syndrome: Prefrontal Cortex and Beyond* , Frontiers in neuroscience **11** (2017), 419.

[15] Arendt T, Zvegintseva HG, Leontovich TA, *Dendritic changes in the basal nucleus of Meynert and in the diagonal band nucleus in Alzheimers disease—A quantitative golgi investigation* , Neuroscience **19** (1986), 1265–1278.

[16] Kaufmann WE, Moser HW, *Dendritic anomalies in disorders associated with mental retardation* , Cereb Cortex **10** (2000), 981–991.

[17] Verret, L., Mann, E. O., Hang, G. B., Barth, A. M., Cobos, I., Ho, K., et al. (2012). *Inhibitory interneuron deficit links altered network activity and cognitive dysfunction in Alzheimer model.* Cell 149, 708–721. doi: 10.1016/j.cell.2012.02.046

[18] Marin, O. (2012). *Interneuron dysfunction in psychiatric disorders.* Nat. Rev.Neurosci. 13, 107–120. doi: 10.1038/nrn3155

[19] Williams, P.A., Thirgood, R.A., Oliphant, H., Frizzati, A., Littlewood, E., Votruba, M., Good, M.A., Williams, J., and Morgan, J.E, *Retinal ganglion cell dendritic degeneration in a mouse model of Alzheimer's disease* , Neurobiol. Aging **34** (2013), 1799–1806.

[20] Gillette T.A., Ascoli G.A., *Measuring and Modeling Morphology: How Dendrites Take Shape.*, . In: Le Novère N. (eds) Computational Systems Neurobiology. Springer, Dordrecht

[21] Guerra L, McGarry LM, Robles V, Bielza C, Larranaga P, Yuste R, *Comparison between supervised and unsupervised classifications of neuronal cell types: a case study* , Dev Neurobiol **71(1)** (2011), 71-82.

[22] Polavaram, S., Gillette, T. A., Parekh, R., Ascoli, G. A. (2014). *Statistical analysis and data mining of digital reconstructions of dendritic morphologies.* Frontiers in neuroanatomy, 8, 138. https://doi.org/10.3389/fnana.2014.00138

[23] Hasse JM, Bragg EM, Murphy AJ, Briggs F, *Morphological heterogeneity among corticogeniculate neurons in ferrets: quantification and comparison with a previous report in macaque monkeys* , J Comp Neurol **527(3)** (2019), 546-557.

[24] Lida Kanari, Pawel Dlotko, Martina Scolamiero, Ran Levi, Julian Shillcock, Kathryn Hess, Henry Markram *A Topological Representation of Branching Neuronal Morphologies* , Neuroinformatics **16(1)** (2018), 3-13.

[25] Kanari, L., Ramaswamy, S., Shi, Y., Morand, S., Meystre, J., Perin, R., Abdellah, M., Wang, Y., Hess, K., Markram, H. (2019). *Objective Morphological Classification of Neocortical Pyramidal Cells. Cerebral cortex* (New York, N.Y. : 1991), **29(4)**, 1719–1735. https://doi.org/10.1093/cercor/bhy339

[26] Li Y, Wang D, Ascoli GA, Mitra P, Wang Y., *Metrics for comparing neuronal tree shapes based on persistent homology.* , PLoS One (2017), 12(8):e0182184.

[27] Ascoli GA, Donohue DE, Halavi M, *NeuroMorpho.Org: A central resource for neuronal morphologies* , J Neurosci **27(35)** (2007), 9247-51.

[28] Scorcioni, R., Polavaram, S., Ascoli, G. *L-Measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies.* Nat Protoc **3**, (2008) 866–876

[29] Chen JR, Wang BN, Tseng GF, Wang YJ, Huang YS, Wang TJ, *Morphological changes of cortical pyramidal neurons in hepatic encephalopathy*, BMC Neurosci **15(15)** (2014), 1-12.

[30] W. de Vazelhes, C.J. Carey, Y. Tang, N. Vauquier, A. Bellet, *metric-learn: Metric Learning Algorithms in Python*, Journal of Machine Learning Research 21 (2020) 1–6.

[31] Kawaguchi Y, Karube F, Kubota Y, *Dendritic branch typing and spine expression patterns in cortical nonpyramidal cells* , Cereb. Cortex **16(5)** (2006), 696-711.

[32] Keith RE, Azcarate JM, Keith MJ, Hung CW, Badakhsh MF, Dumas TC, *Direct Intracellular Signaling by the Carboxy terminus of NMDA Receptor GluN2 Subunits Regulates Dendritic Morphology in Hippocampal CA1 Pyramidal Neurons.*, Neuroscience. **396** (2019), 138-153

[33] Ruth Benavides-Piccione, Farid Hamzei-Sichani, Inmaculada Ballesteros-Yanez, Javier DeFelipe and Rafael Yuste, *Dendritic Size of Pyramidal Neurons Differsamong Mouse Cortical Regions*, Cerebral Cortex (2005) doi:10.1093/cercor/bhj041

[34] Mihaljević, B., Benavides-Piccione, R., Guerra, L., DeFelipe, J., Larrañaga, P., Bielza, C. (2015). *Classifying GABAergic interneurons with semi-supervised projected model-based clustering.* Artificial intelligence in medicine, **65(1)**, 49–59. https://doi.org/10.1016/j.artmed.2014.12.010

[35] H. Cuntz , F. Forstner, A. Borst, M. Häusser, *One Rule to Grow Them All: A General Theory of Neuronal Branching and Its Practical Application*, PLOS Computational Biology, (2010).

[36] Elston, G. N., Tweedale, R., Rosa, M. G. (1999). *Cellular heterogeneity in cerebral cortex: a study of the morphology of pyramidal neurons in visual areas of the marmoset monkey.* The Journal of comparative neurology, **415(1)**, 33–51. https://doi.org/10.1002/(sici)1096-9861(19991206)415:1¡33::aid-cne3¿3.0.co;2-m

[37] Elston, G. N., Elston, A., Casagrande, V., Kaas, J. H. (2005). *Areal specialization of pyramidal cell structure in the visual cortex of the tree shrew: a new twist revealed in the evolution of cortical circuitry.* Experimental brain research, **163(1)**, 13–20. https://doi.org/10.1007/s00221-004-2131-7

[38] Benavides-Piccione, R., Hamzei-Sichani, F., Ballesteros-Yáñez, I., DeFelipe, J. Yuste, R. (2006). *Dendritic size of pyramidal neurons differs among mouse cortical regions.* Cereb. Cortex, 16, 990–1001.

[39] Ballesteros-Yáñez, I., Benavides-Piccione, R., Elston, G.N., Yuste, R. DeFelipe, J. (2006). *Density and morphology of pyramidal cell dendritic spines in the mouse neocortex.* Neuroscience, 138, 403–409.

[40] Boldog, E., Bakken, T. E., Hodge, R. D., Novotny, M., Aevermann, B. D., Baka, J., Bordé, S., Close, J. L., Diez-Fuertes, F., Ding, S. L., Faragó, N., Kocsis, Á. K., Kovács, B., Maltzer, Z., McCorrison, J. M., Miller, J. A., Molnár, G., Oláh, G., Ozsvár, A., Rózsa, M., Tamás, G. (2018). *Transcriptomic and morphophysiological evidence for a specialized human cortical GABAergic cell type.* Nature neuroscience, 21(9), 1185–1195. https://doi.org/10.1038/s41593-018-0205-2

[41] Flores G, Morales-Medina JC, Diaz A. *Neuronal and brain morphological changes in animal models of schizophrenia.* Behav Brain Res. 2016;301:190-203. doi:10.1016/j.bbr.2015.12.034

[42] Kaufmann, W. E., Moser, H. W. (2000). *Dendritic anomalies in disorders associated with mental retardation.* Cerebral cortex (New York, N.Y. : 1991), 10(10), 981–991. https://doi.org/10.1093/cercor/10.10.981

[43] A. D. Bird, H. Cuntz, *Dissecting Sholl Analysis into Its Functional Components*, Cell Reports

[44] A. Bellet, A. Habrard, M. Sebban, *Metric Learning*, Morgan and Claypool Publishers, ISBN-10 : 1627053654, ISBN-13 : 978-1627053655.

[45] E. Denne and J. M. Sullivan,*Convergence and Isotopy Type for Graphs of Finite Total Curvature*