Neuroinform (2011) 9:347–369
DOI 10.1007/s12021-011-9103-4

ORIGINAL ARTICLE

Models and Simulation of 3D Neuronal Dendritic Trees Using Bayesian Networks

Pedro L. López-Cruz · Concha Bielza · Pedro Larrañaga · Ruth Benavides-Piccione · Javier DeFelipe

Published online: 9 February 2011

Abstract Neuron morphology is crucial for neuronal connectivity and brain information processing. Computational models are important tools for studying dendritic morphology and its role in brain function. We applied a class of probabilistic graphical models called Bayesian networks to generate virtual dendrites from layer III pyramidal neurons from three different regions of the neocortex of the mouse. A set of 41 morphological variables were measured from the 3D reconstructions of real dendrites and their probability distributions used in a machine learning algorithm to induce the model from the data. A simulation algorithm is also proposed to obtain new dendrites by sampling values from Bayesian networks. The main advantage of this approach is that it takes into account and automatically locates the relationships between variables in the data instead of using predefined dependencies. Therefore, the methodology can be applied to any neuronal class while at the same time exploiting class-specific properties. Also, a Bayesian network was defined for each part of the dendrite, allowing the relationships to change in the different sections and to model heterogeneous developmental factors or spatial influences. Several univariate statistical tests and a novel multivariate test based on Kullback–Leibler divergence estimation confirmed that virtual dendrites were similar to real ones. The analyses of the models showed relationships that conform to current neuroanatomical knowledge and support model correctness. At the same time, studying the relationships in the models can help to identify new interactions between variables related to dendritic morphology.

Keywords Pyramidal cells · Virtual dendrites · Morphology simulation · Dendritic structure · Bayesian networks

P. L. López-Cruz (✉) · C. Bielza · P. Larrañaga
Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo sn, 28660, Boadilla del Monte, Madrid, Spain
e-mail: pedro.lcruz@upm.es

C. Bielza
e-mail: mcbielza@fi.upm.es

P. Larrañaga
e-mail: pedro.larranaga@fi.upm.es

R. Benavides-Piccione · J. DeFelipe
Laboratorio Cajal de Circuitos Corticales, Universidad Politécnica de Madrid and Instituto Cajal (CSIC), Campus de Montegancedo sn, 28223, Pozuelo de Alarcón, Madrid, Spain

R. Benavides-Piccione
e-mail: rbp@cajal.csic.es

J. DeFelipe
e-mail: defelipe@cajal.csic.es

Introduction

Dendritic morphology is essential for understanding neuronal connectivity and is a crucial feature in information processing and brain function. Pyramidal neurons are key elements in the functional organization of the cerebral cortex, where they are the most frequent neuronal type (70–85%) and the main source of cortical excitatory synapses. Another important feature of pyramidal cells is that their dendritic surfaces are covered by spines, which represent the major postsynaptic elements of excitatory synapses (Feldman 1984; White 1989; DeFelipe and Fariñas 1992; Yuste and

Bonhoeffer 2004; Spruston 2008). Moreover, the structure of the dendritic tree itself affects the process of integration, whereas its size influences the topographic sampling map and the mixing of inputs (Wen et al. 2009). The branching patterns of the dendritic trees are related to the processing of synaptic inputs (Koch et al. 1982; Koch and Segev 2000; Häusser and Mel 2003) and define the electric behaviour of the neurons (Mainen and Sejnowski 1996; Vetter et al. 2001; Chen 2009). Different parts of the dendrites can operate semi-independently according to the spatial location of synaptic connections (Shepherd 2004). As a result, there is considerable interest in the analysis of the microanatomy of pyramidal cells since it constitutes an excellent tool for better understanding cortical information processing.

Despite recent advances in molecular biology and new discoveries related to neuronal development, current knowledge about neuron structure is still incomplete, and it is hard to find a set of anatomical traits that unambiguously define a neuron type (Ascoli et al. 2008). Computational stochastic models have been used for the last two decades to measure geometric parameters of real neuronal arborizations and simulate virtual neuron morphologies. These simulations can be used to identify the basic structures and important features in neuronal classes, to study neuronal development and neurite outgrowth or to examine relationships between morphology and neuronal function. As 3D reconstruction of real cells is a time- and resource-consuming process, the data compression and amplification that can be achieved with these techniques are also important advantages (Ascoli et al. 2001; Ascoli 2007). However, a major obstacle to the creation of virtual neurons is method validation because data on the complete dendritic tree of real neurons is rather scarce. Indeed, labeled processes are frequently incomplete because, during the tissue slicing procedures some parts of the neuron morphology are missing in a varying degree, depending on the thickness of the sections and the relative localization of the labeled neuron within the slice. Usually, this problem can only be overcome using serial sections to reconstruct the cell in 3D. However, neuronal processes are not always easy to trace and they may at times get lost in the background noise (DeFelipe 2008). Together, these obstacles make it very laborious and time-consuming to obtain meaningful measurements from neurons. In this study, we have used data from fully reconstructed basal dendrites of pyramidal cells. Indeed, there is a spatial segregation of different inputs into different regions of the dendritic arborization, which can be divided into two major compartments: the apical dendrite with its collateral branches and the dendritic tuft, and the basal dendrites. The basal dendritic arbors of pyramidal cells represent about 90% of the dendritic length of cortical pyramidal neurons from layers II/III and V (Larkman 1991). The whole basal dendritic arbor can be fully reconstructed in single horizontal sections (Elston and Rosa 1997). Thus, they are particularly valuable for validating the simulated virtual neurons.

Existing models for simulating dendritic morphology can be grouped in two categories: growth and reconstruction models (Van Pelt and Uylings 2005). Growth models try to capture the behavior of growth cones during neuron development and, thus, are able to simulate dendritic structure at its different stages of maturity. These models usually consider that the neurite tips elongate and taper as they grow away from the soma until a bifurcation occurs or the neurite ends. They estimate the probabilities for each of these events taking into account some of the different factors involved in neuron development, e.g., molecular gradients (Hentschel and van Ooyen 1999), electric field presence (Robert and Sweeney 1997), neuritic tension (Li and Qin 1996), segment length or centrifugal order (Van Pelt et al. 2001), neurotrophic particles (Luczak 2006), etc. One of the recent works that implements this kind of model (Koene et al. 2009) has simulated complete networks of neurons. The probability functions include complex elements, e.g., the influence of competition between dendrites when deciding if a bifurcation should occur, the distance between dendrites and axons when establishing synaptic connections, etc.

On the other hand, reconstruction models measure relevant variables from real neurons and use their statistical distributions to describe the dendritic tree structure. Then, a simulation algorithm samples the distributions to output virtual dendrites that should be indistinguishable from real ones. Donohue and Ascoli (2005a, b) propose an algorithm that samples 2D virtual dendrites from the univariate marginal statistical distributions of some basic parameters (segment length, width and bifurcation probability). Later, they consider conditional relationships between the variables and three other fundamental parameters (centrifugal order, segment radius and path length) and compare the models at length (Donohue and Ascoli 2008). They use common predefined parametric distributions, like Gaussian, Gamma or uniform distributions, to fit the data.

Parametric distributions might not accurately fit the real distributions, and other models use non-parametric approaches to avoid that problem. Lindsay et al. (2007) apply kernel density estimation (KDE) to simulate 2D dendritic structures taking into account conditional

relationships between features. Torben-Nielsen et al. (2008b) use conditional KDE to generate dendrites and angle information is included to obtain 3D simulations.

Variations of L-Systems (Rozenberg and Salomaa 1980) have also been used in neuronal morphology simulations because of their ability to create branching structures (Hamilton 1993; Ascoli and Krichmar 2000). Recently, evolutionary computation has been used to output L-Systems that generate virtual neurons that are similar to a single real one (Torben-Nielsen et al. 2006, 2007, 2008a). They also include a postprocessing step to filter out non-biologically plausible neurons.

The above models only measure univariate marginal probability distributions or define a priori conditional relationships ad hoc (e.g., Lindsay et al. 2007; Anwar et al. 2009). Considering variables to be independent keeps models simpler, which makes them easy to analyze. However, the independence assumption does not hold since complex interactions with extracellular elements and intrinsic factors have been widely reported for real neurons (McAllister 2000; Scott and Luo 2001). Other works define relationships between model parameters according to some predefined criterion and study the simulated neurons to check whether or not the hypotheses are correct (e.g., Samsonovich and Ascoli 2003; Donohue and Ascoli 2008). This methodology is more likely to be biased towards expert knowledge and disregards important information that could be inferred from the data.

In this paper, we present a novel methodology for the 3D simulation of dendritic trees based on a computational model called Bayesian networks (Pearl 1988; Koller and Friedman 2009). The goal is to simulate virtual dendrites that are visually and statistically indistinguishable from real ones. The whole process is summarized in Fig. 1. Our proposal can be classed as a reconstruction model, i.e., we measure key features in 3D reconstructions of real pyramidal neurons and estimate their joint probability distribution to define the Bayesian networks used to build up the model. We then simulate from the Bayesian networks to output a set of virtual dendrites, which we compare with the original data to verify the model's ability to capture the dendritic tree structures. Statistical tests are performed to check whether or not the variables included in the model have the same distribution in the original and simulated data. Other emergent features measured from the whole dendritic trees and branches, not used in model learning, are also compared, e.g., total dendritic length, asymmetry index, etc. Wilcoxon rank-sum, Kolmogorov–Smirnov and Kullback–Leibler divergence-based tests are used to compare each variable independently. We also propose a multivariate test that uses a Kullback–Leibler divergence estimation to compare the joint probability distribution over a set of variables. Finally, examples of real and simulated dendrites are shown for visual comparison.

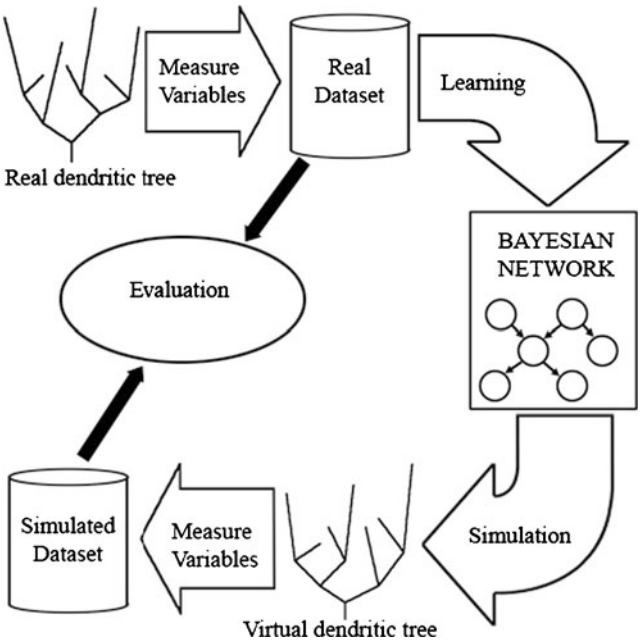

Fig. 1 Our reconstruction model approach

This approach has a number of advantages over previous models. First, Bayesian networks use the conditional (in)dependencies between the variables to model the joint probability distribution. In fact, the possible use of Bayesian networks to consider the relationships between morphological variables has already been noted (Torben-Nielsen et al. 2008b). Since these statistical relationships are found automatically by analyzing the data, the whole process is data-driven and widely applicable. Thus, instead of changing the model to consider different kinds of relationships and analyzing the results to gather relevant information, we let the data speak. The resulting model is studied to gain insights into the processes underlying dendritic morphology. We believe that this approach is less constrained by a priori assumptions based on current biological knowledge and is not affected by disagreements between domain experts.

Second, the model learns and uses a Bayesian network for each part of the dendrite. This way, the relationships can change to take into account the dendritic tree location. This is an important characteristic, since there is evidence supporting the idea that heterogeneous parts of the dendrites could be regulated by different developmental factors (Donohue and Ascoli

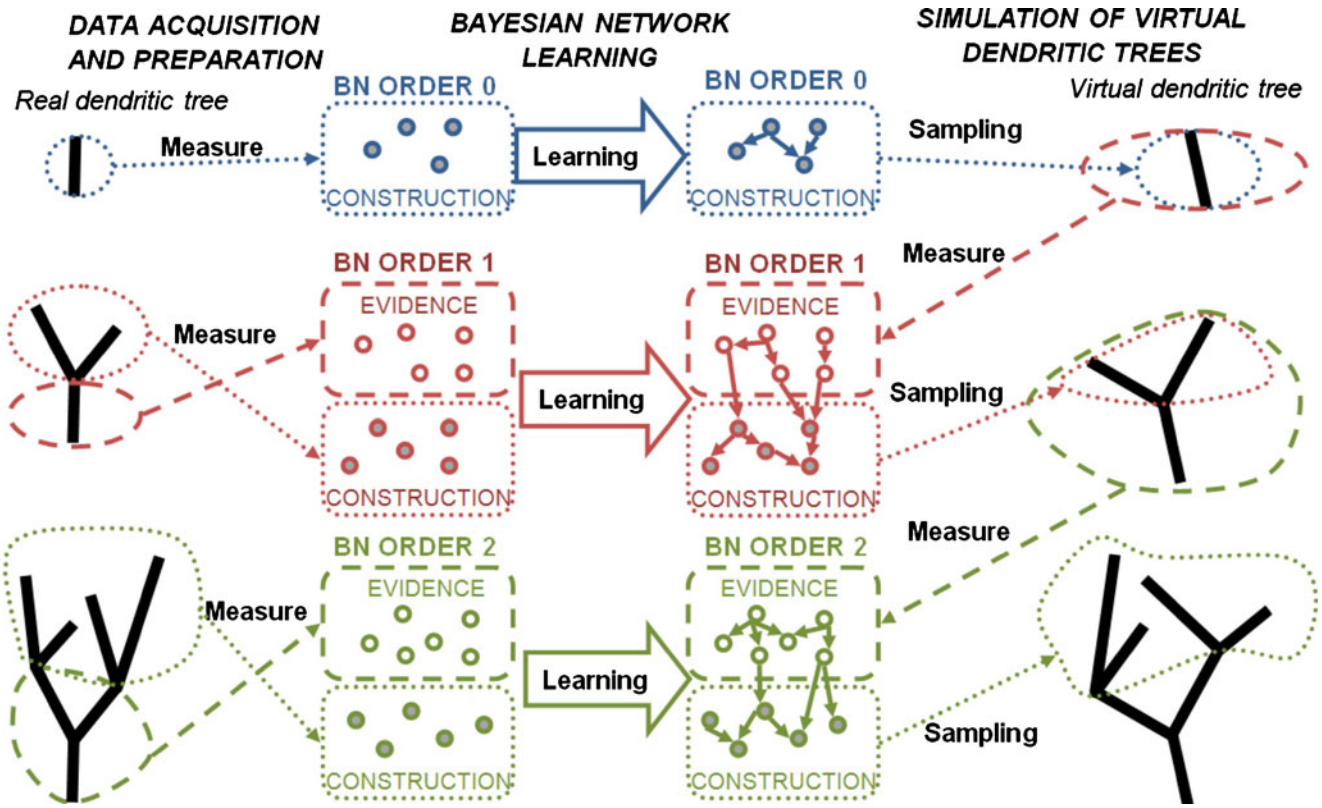

Fig. 2 Application of Bayesian networks to the modeling and simulation of basal dendritic trees. The figure shows the measuring of variables from the real trees, the learning of the Bayesian networks and the sampling of values to simulate the virtual dendritic trees

2005a). The model is flexible enough to capture and exploit low-level relationships.

Third, the model uses an extensive set of variables that include commonly used dendritic tree measurements along with new features to model context factors, such as the morphology of the subdendrite or the distance to the nearest segment. Finally, the use of several univariate tests and a novel multivariate test makes the evaluation more robust and reliable. A supplementary material website has been set up with additional results (see "Information Sharing Statement").

Methods

We provide details about each of the steps involved in the methodology presented in this paper. The data acquisition and preparation section explains how the variables used in the model were measured from the real dendrites and the data were preprocessed. We introduce the Bayesian networks paradigm and describe the algorithm used for learning from data. Then, the simulation algorithm that uses the Bayesian networks to generate virtual 3D dendrites is shown. Finally, the evaluation methodology comparing virtual and real dendrites is presented. Figure 2 shows a schematic overview of the whole process.

Data Acquisition and Preparation

We have used a set of 3D reconstructions of 90 pyramidal neurons from the mouse neocortex (two BC57 Black mice, 2 months old). These neurons were labeled with Lucifer Yellow using an intracellular injection method that covers the full extent of the basal dendritic arbor. The neurons were located in layer III of different cortical regions: the secondary motor cortex (M2), the secondary somatosensory cortex (S2) and the lateral secondary visual cortex and association temporal cortex (V2L/TeA). Therefore, three databases[1] of reconstructions were built according to their cortical area. The whole basal dendritic trees of the neurons were traced using the Neurolucida package (Glaser and Glaser 1990, MicroBrightField) and stored in digital

[1]The term "database" refers to the sets of 3D reconstructions of basal dendrites from each of the three cortical areas. The term "dataset" is used to refer to the values of the variables measured for each pair of sibling segments in those reconstructions.

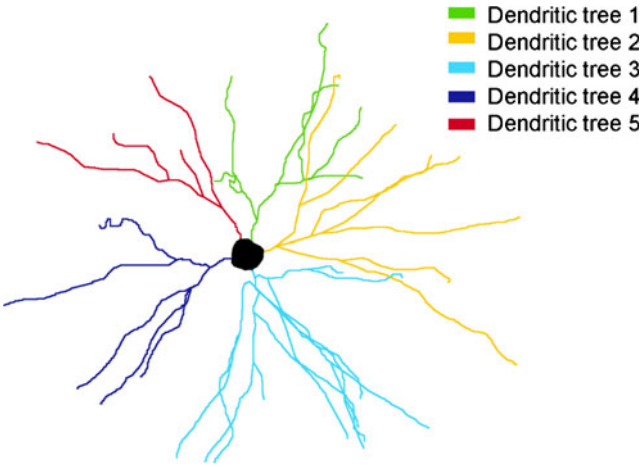

Fig. 3 Basal dendritic arbor of a pyramidal neuron from the secondary motor cortex. Each dendritic tree is drawn in a different color

files in the ASC Neurolucida format. The tissue preparation and injection process are detailed in Benavides-Piccione et al. (2006). The reconstructions are publicly available at www.neuromorpho.org (Ascoli et al. 2007) as part of DeFelipe's laboratory archive. Each basal dendritic arbor is made up of approximately 6 (mean $\pm$ SD, 5.7 ± 0.9, range 4–8) main trunks, which are in turn made up of several dendrites, as shown in Fig. 3. For the sake of simplicity and unless otherwise stated, we called these single trunks of basal dendritic arbors *dendritic trees*.

The 3D reconstructions were made up of the Cartesian coordinates of the points where the dendrites branched. Each dendritic tree was isolated and its coordinates were moved and rotated such that the root point was placed at the coordinate system origin and the root segment was on the vertical axis. We considered a segment as the straight line between two branch points. The dendritic trees with multifurcations (branch points that are the source of three or more segments) were discarded.

A set of 41 variables was measured for each pair of sibling segments (Table 1). Some of the variables were selected because they have been widely used to describe dendritic morphology (Hillman 1979; Verwer et al. 1992; Uylings and van Pelt 2002; Brown et al. 2008), whereas other new variables have been included to capture context influence and neuritic competition. Two types of variables were identified: evidence (E) and construction (C) variables. Construction variables define the morphology of a segment. In the simulation step, construction variables are sampled by the model to incrementally build the virtual dendritic trees. On the other hand, evidence variables measure the part of the dendritic tree morphology previous to a pair of sibling segments. Evidence variables are measured

Table 1 Variables measured from the real dendritic trees and used for learning the model

No.	Type	Variable	No.	Type	Variable
1	E	Subtree degree (no. endings)	22	E	Neighbor distance
2	E	Subtree no. bifurcations (no. nodes)	23	E	Neighbor inclination
3	E	Subtree total length	24	E	Neighbor azimuth
4	E	Subtree width	25	E	Neighbor extension
5	E	Subtree height	26	E	Neighbor angle
6	E	Subtree depth	27	E	Parent segment length
7	E	Subtree box volume	28	E	Parent segment inclination
8	E	Subtree max distance between nodes	29	E	Parent segment azimuth
9	E	Subtree max distance to soma	30	E	Root segment length
10	E	Subtree max length	31	E	Root segment inclination
11	E	Subtree min length	32	E	Root segment azimuth
12	E	Subtree max order	33	E	Segment centrifugal order
13	E	Subtree min order	34	C	Left segment length
14	E	Subdendrite length	35	C	Left segment inclination
15	E	Subdendrite width	36	C	Left segment azimuth
16	E	Subdendrite height	37	C	Left segment bifurcates
17	E	Subdendrite depth	38	C	Right/root segment length
18	E	Subdendrite box volume	39	C	Right/root segment inclination
19	E	Subdendrite distance to soma	40	C	Right/root segment azimuth
20	E	Subdendrite inclination	41	C	Right/root segment bifurcates
21	E	Subdendrite azimuth			

We distinguished two variable types: evidence variables (E), which provide information about the subtree and subdendrite, and construction variables (C), which describe the segment length, orientation and bifurcation

during the simulation process and used as information to accurately sample the construction variable values.

- Evidence variables (E), which provide information about the context of the segments and how the dendritic tree is constructed (variables 1–33). These features include (Fig. 4):
  - Morphological data from the subtree (variables 1–13). Given two sibling segments with order $= x$, the subtree is defined as the part of the dendritic tree considering all the segments with order $< x$. For example, Fig. 4a and b show a pair of sibling segments with a centrifugal order value of 5 (in red). The subtree (blue area) includes all the segments with an order value from 0–4. This information could be interesting for considering dendritic size when deciding if a segment should branch or to control the spread and direction of the dendritic tree.
  - Variables that describe the subdendrite (variables 14–21). Given two sibling segments, the subdendrite is the path from the soma to the segments' branching point. The dotted area in Fig. 4a represents the subdendrite for the two sibling segments (in red), including the segments along the path with lower centrifugal order, i.e., the segments in the path with order 0–4. These data could be used as a way to capture neuron tropism and determine segment direction or to bound segment length.
  - Information about the nearest segment (neighbor in Fig. 4c) in the dendritic tree that is not a part of the subdendrite (22–26). These variables have been inspired by previous studies about the branching patterns of trees in the field of ecology (Sumida et al. 2002; Miina and Pukkala 2002), where several competition indexes and measures are given to model tree growth when the influence of neighboring trees is acknowl-

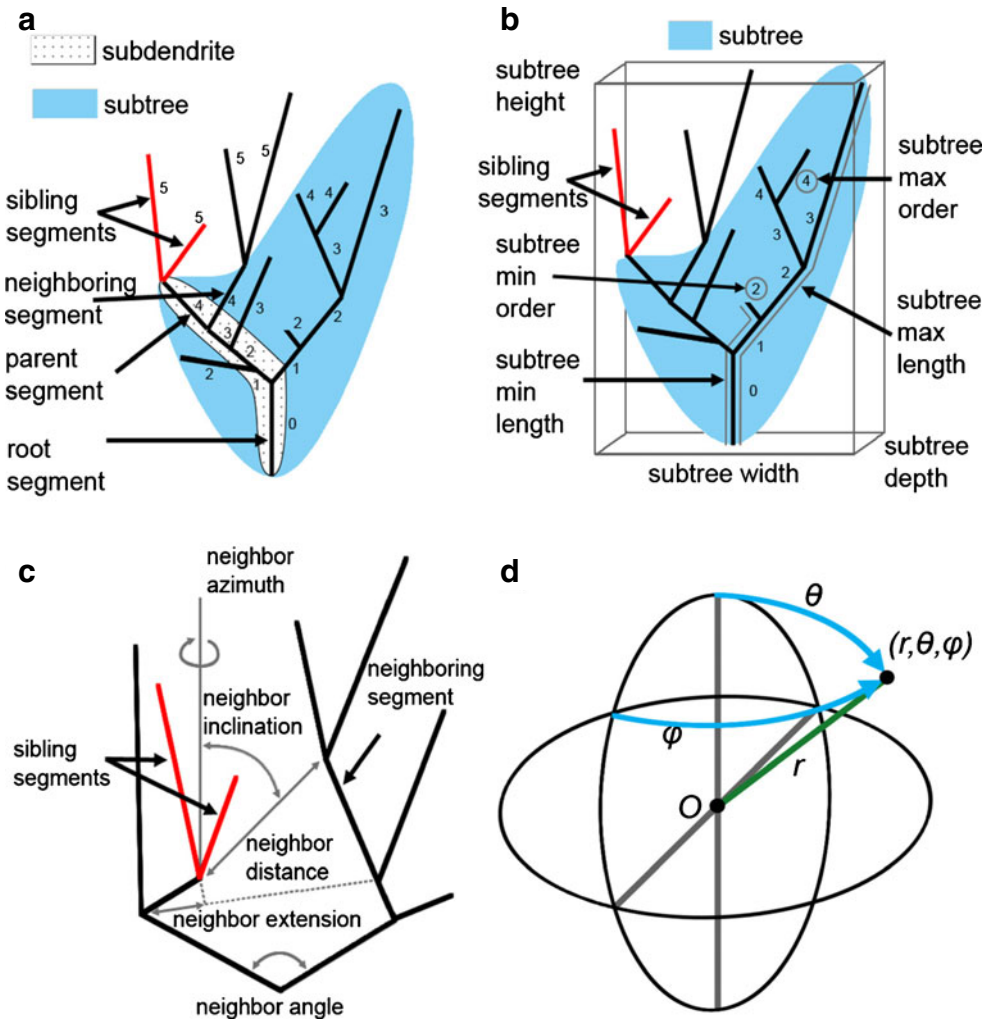

Fig. 4 Scope of the variables used in the model. **a** shows the subtree (*blue area*), subdendrite (*dotted area*), parent and root segments for the two sibling segments (*in red*). The numbers refer to the centrifugal order of the segments. **b** illustrates the variables measured from the sibling segments subtree. **c** shows the variables related to the segment closest to the segment starting point (neighbor). **d** shows a spherical coordinate system where a segment is defined by the spherical coordinates (r, θ, φ) of its end point taking the starting point as the origin: r is the Euclidean distance between the two points, θ is the inclination angle and φ is the azimuth angle

edged. Following these approaches, we calculated five variables to take into account possible competition for resources between different branches, e.g., distance to the neighboring segment, angle between the subbranches of the two segments, etc.
  - Variables 27–29 describe parent segment morphology and enable interactions between the three segments that are involved in a bifurcation (Fig. 4a). These variables are included to consider relationships between consecutive segments.
  - Root segment variables (30–32) correspond to the first segment of the dendrite that grows away from the soma (Fig. 4a). These variables could also help to capture dendritic tropism. Subdendrite information, along with parent and root segment measures, have already been used to model neurite growth direction (Samsonovich and Ascoli 2003).
  - The centrifugal order of the segment (branch order), i.e., the number of bifurcations along the path to the soma (variable 33).
- Construction variables (C) completely specify segment morphology (variables 34–41). This group determines whether or not the segments branch, as well as the spherical coordinates of each end point taking the starting point as the origin (Fig. 4d). A distinction between the two segments in a bifurcation was made based on the azimuth angle of their end points. We defined the left segment in a bifurcation as the one having a higher absolute azimuth value. Therefore, the right segment is the one with a smaller absolute value of the azimuth angle.

We formed four datasets by centrifugal order of segments: root segments (order 0), first-order segments (order 1), second-order segments (order 2) and segments with a higher order (order > 2). Finally, each of the 41 variables in each dataset was discretized by mimicking their histograms. Two or three discrete values were defined for each variable trying to preserve the shape of the empirical distributions while ensuring that enough data was available in each interval to accurately estimate Bayesian network probabilities.

Basics of Bayesian Networks

Bayesian networks (also known as probabilistic networks or belief networks) are the representation of choice for uncertainty in artificial intelligence (Koller and Friedman 2009). In this work, the model is made up of a Bayesian network with discrete variables for each one of the datasets.

Bayesian networks are a kind of probabilistic graphical model with two main elements: the graphical component and the probabilistic component. Formally, a Bayesian network can be defined as a pair $B = \langle G(\mathbf{X}, \mathbf{A}), \mathbf{P} \rangle$:

- The graphical part $G(\mathbf{X}, \mathbf{A})$ is a directed acyclic graph (DAG) used to capture the structure of the problem. A graph is defined by a set of nodes ($\mathbf{X}$) and a set of edges connecting the nodes. A DAG is a graph that contains only directed edges (called arcs, $\mathbf{A}$) without a cycle, i.e., there is no directed path $X_i \rightarrow ... \rightarrow X_k$ where $X_i = X_k$. In a Bayesian network, the variables in the problem domain are represented by the nodes ($\mathbf{X}$) in the DAG, and the probabilistic conditional (in)dependence relationships between the variables are codified with arcs ($\mathbf{A}$) connecting the nodes.
- The probabilistic component $\mathbf{P}$ includes the conditional probability distributions $P(X_i|\mathbf{\Pi}_i)$ associated with the variables $X_i, i = 1, ..., n$ in the problem. For each variable X_i, we define the set of its parents as $\mathbf{\Pi}_i = \{Y \in \mathbf{X} | (Y, X_i) \in \mathbf{A}\}$.

A Bayesian network encodes a factorization of the joint probability distribution over all the variables in $\mathbf{X}$:

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i|\mathbf{\Pi}_i) \quad (1)$$

These models have been widely used to solve different kinds of problems (classification, regression, simulation...) because they can compactly represent the problem domain, and factorization enables efficient computations that would be intractable otherwise. Moreover, these models can perform any type of reasoning: predictive, diagnostic, bidirectional, abductive...

Bayesian Network Learning and Model Construction

There are several methods for learning both the graphical structure and probabilistic distributions of Bayesian networks (Heckerman 1996; Krause 1998). Two main approaches can be identified for network structure learning:

- Detecting conditional independencies between variables. This approach is based on performing statistical tests to the data in order to identify the conditional independencies between the variables. Then, the Bayesian network structure (DAG) can be generated from the set of independencies.

- Applying a score+search method. These methods use a score function to estimate the quality of a model and a search algorithm to find the best structure according to that score.

We focused on the second approach (score+search). The Bayesian Information Criterion, BIC (Schwarz 1978), was used as a score function to evaluate the network structures. This score measures the log-likelihood ℓ of a Bayesian network ($B = \langle G, \mathbf{P}\rangle$) given a dataset ($D$) with M cases. A penalization is included to take into account the structure's dimension (number of parameters to completely define the Bayesian network) and avoid very complex networks. The BIC score is given by the expression

$$BIC(B : D) = \ell(\langle G, P\rangle : D) - \frac{\dim(G)}{2} \ln M. \qquad (2)$$

We would like to learn the model based only on the data, so every network structure has to be considered. However, the number of possible networks for any real problem makes an exhaustive evaluation infeasible. Therefore, a search method is necessary to find and explore a subset of reasonably good structures, where the quality of a network is given by the BIC score, which should be maximized.

In this research, the K2 heuristic search algorithm (Cooper and Herskovits 1992) was applied to efficiently examine the space of network structures. Table 2 outlines the K2 algorithm at a high level.

The maximum number of parents (u) allowed for each variable was set to three in our experiments. K2 needs the variables in $\mathbf{X}$ to be ordered (step 1). Leray and Francois (2006) proposed using the *maximum weight spanning tree algorithm* (MWST) to compute such ordering. First, each possible edge is given a weight that corresponds to the BIC score variation when the variables are related. Then, Prim's algorithm uses those weights to build an undirected tree, and a root node is selected so that the edges can be oriented. The centrifugal order (variable 33) was always used as the root node. Finally, a topological sorting method was applied to partially order the variables. Evidence variables were forcibly placed at the top of the list, followed by the construction attributes.

After ordering the variables, K2 incrementally builds the Bayesian network (Table 2). At each iteration, one variable is selected according to the ordering of variables (step 4). All the previous variables in the ordering are considered as candidate parents. The candidate variable that gives the highest BIC score is selected as a parent, until the score decreases or the maximum number of parents allowed per variable is reached. The process is repeated until all variables have been considered. Evidence variables can condition construction variables and give information about them because they appear first in the ordering.

Once the structure is known, the probabilistic component of the Bayesian network can be calculated. Estimating each variable probability distribution amounts to counting the relative frequency of each value given its parent values. This is equivalent to the maximum likelihood estimate

$$P(X_i = x_i | \mathbf{\Pi}_i = \boldsymbol{\pi}_i) = \frac{\text{freq}(X_i = x_i, \mathbf{\Pi}_i = \boldsymbol{\pi}_i)}{\text{freq}(\mathbf{\Pi}_i = \boldsymbol{\pi}_i)}. \qquad (3)$$

The Bayesian network is completely defined when the DAG and the probability distributions of all its variables have been specified. Bayes Net Toolbox for Matlab was used to run these algorithms (Murphy 2001), and a Bayesian network was learned for each of the four datasets. Thus, the model comprises four Bayesian networks that capture the relationships between the variables at the different levels of the dendritic trees (see Fig. 2).

Simulation Algorithm for Generating Virtual Dendritic Trees

The simulation process uses the Bayesian networks included in the model to generate the virtual dendritic trees. An iterative algorithm (Table 3) measures the

Table 2 K2 search algorithm

Given a dataset $\mathcal{D}$ and a maximum number of parents u.
1. $Order(\mathbf{X})$.
2. Create an empty Bayesian network $B = \langle G(\mathbf{X}, \mathbf{A}), P\rangle$.
3. $BIC_{\max} = BIC(B : \mathcal{D})$.
4. Repeat for each variable $X_i \in Order(\mathbf{X})$:
4.1. If the number of parents of X_i is equal to u, go to next variable in $Order(\mathbf{X})$ (step 4).
4.2. Find $X_j \in \{X_1, ..., X_{i-1}\}$ that maximizes $BIC(B' = \langle G(\mathbf{X}, \mathbf{A}'), P\rangle : \mathcal{D})$ where $\mathbf{A}' = \mathbf{A} \cup (X_j, X_i)$.
4.3. If $BIC(B' : \mathcal{D}) < BIC_{\max}$, go to next variable in $Order(\mathbf{X})$ (step 4). Else, set $B = B'$, $BIC_{\max} = BIC(B' : \mathcal{D})$ and go to step 4.1.

Table 3 Simulation algorithm for generating virtual dendritic trees

Repeat while there are incomplete dendrites
1. Select the appropriate Bayesian network depending on the centrifugal order of the segment to be sampled.
2. Measure evidence variables from the dendritic tree built so far.
3. Discretize the variables and set their values in the Bayesian network.
4. Sample the construction variable values from the Bayesian network.
5. Transform the spherical coordinates of the segments back to Cartesian coordinates to build the segments.
6. If a segment bifurcates, consider that the dendrite is still incomplete. Else, the dendrite has ended.

evidence variables and uses that information to increasingly build the virtual dendrite. The algorithm simulates the dendritic tree in a breadth-first way according to the centrifugal order (see Fig. 2), i.e. first the root segment is created, then order-1 segments are generated, followed by order-2 segments for the previous branching segments, etc.

In the Data Acquisition and Preparation, the variables were discretized and the Bayesian networks were learned with those discrete values. Therefore, the values sampled from the Bayesian networks were also discrete (step 4 in Table 3). These discrete values had to be converted back to continuous values in order to build a virtual dendritic tree.

Bayesian network learning prevents the discretization process from making a high number of bins and ensures that enough data is available for accurately estimating the probability distributions. However, a low number of discrete values produces wide intervals with complex and heterogeneous data distributions. Therefore, it is not appropriate to use a central tendency measure as the mean or the median to convert a discrete into a continuous value. Parametric fitting of the data to some theoretical distribution was also

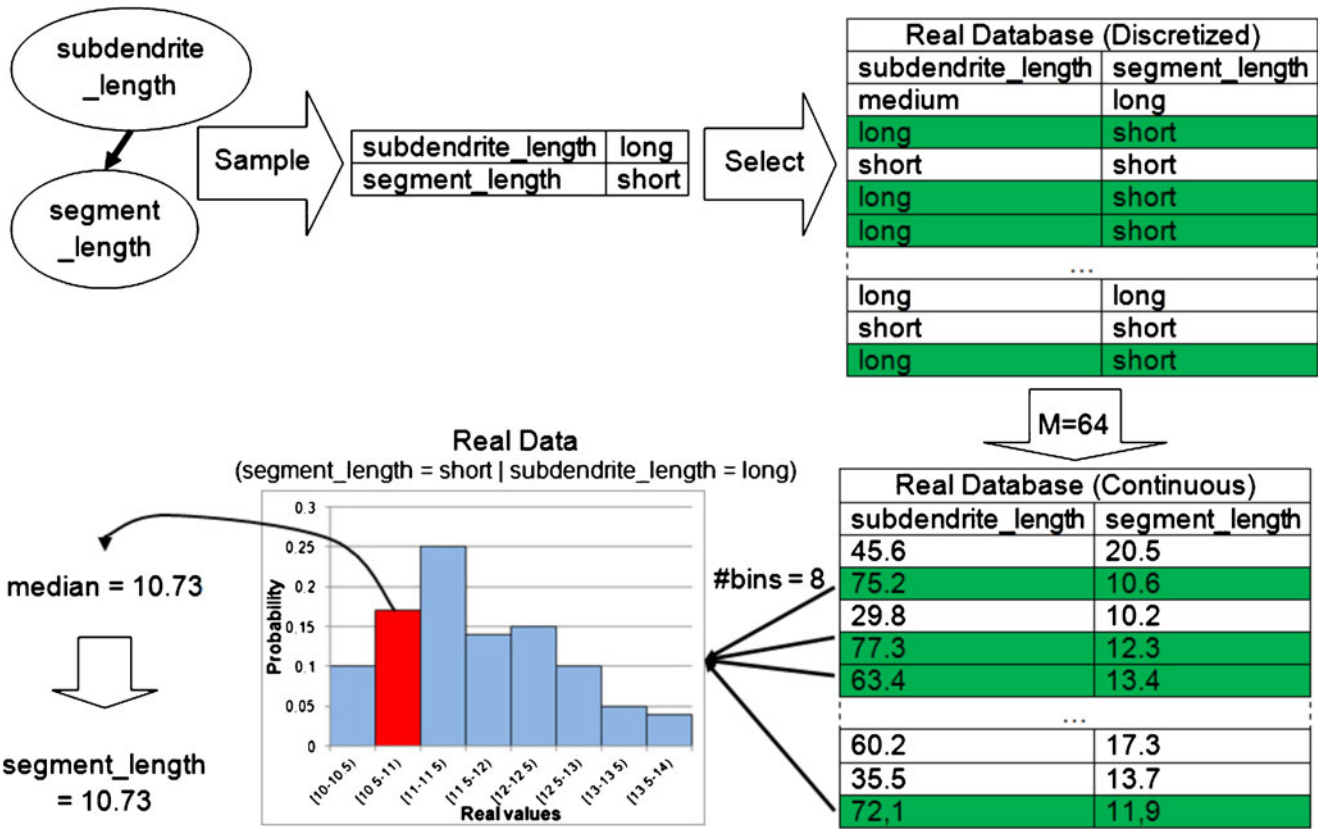

Fig. 5 Example of the transformation of discrete values to continuous values. The conditional histogram of the real continuous values that were discretized to the values *subdendrite_length=long* and *segment_length=short* is calculated. Then, one bin is selected randomly and the median of its values is the corresponding continuous value

avoided because of the high complexity of the models that work with such distributions and a low goodness of fit due to unusual data shapes.

Thus, we applied a method that samples a continuous value exploiting the original data without making any assumption about its shape. Figure 5 shows an example where, as the result of the sampling, the construction variable *segment_length* is given the value *short* and the parent variable *subdendrite_length* has the value *long*. The method was based on the conditional histograms of the real continuous values. For each simulated construction variable $X_i \in \mathbf{X}, i \in \{34, ..., 41\}$ that takes the discrete value x_i and whose parents values are $\mathbf{\Pi}_i = \boldsymbol{\pi}_i$, the values $(X_i = x_i, \mathbf{\Pi}_i = \boldsymbol{\pi}_i)$ were found in the discretized real dataset. Then, the corresponding continuous real values were retrieved (M is the number of selected samples). We built a histogram with those continuous values using $\sqrt{M}$ equal-width bins. One bin was then selected randomly, where the bin probability was proportional to the number of data in the interval. The final continuous value corresponding to the discrete value x_i was the median of the data in the selected bin.

The continuous values were finally converted to the Cartesian coordinates used to simulate the virtual dendrites (step 5 in Table 3).

Evaluation Methodology

The last step in the methodology is to check if the virtual dendrites are statistically and visually indistinguishable from the real ones. This would mean that the model is able to accurately capture the processes underlying dendritic morphology and can be analyzed to extract relevant knowledge. Van Pelt and Uylings (1999) compared real and simulated dendritic trees using both optimized parameters included in the model and predicted variables that emerged as an outcome of their growth model. Later on, Ascoli et al. (2001) proposed three ways of validating a reconstruction model: comparing real and simulated statistical distributions of the variables used in the model, comparing real and simulated emergent parameters not included in the model, and performing a visual inspection by experts in neuroanatomy.

We can evaluate each variable independently using univariate statistical tests. However, since Bayesian networks model the joint probability distribution over all the variables in the problem, a method that compares this multivariate distribution in real and simulated data would be desirable. Therefore, a test based on Kullback–Leibler divergence that can be applied to both univariate and multivariate data was designed. Kullback–Leibler divergence (Kullback and Leibler 1951) measures the "distance" from a true probability distribution $p(x)$ to a reference distribution $q(x)$ and is a commonly used technique to quantify the difference between distributions. We refer to these tests as the univariate and multivariate KL tests, respectively.

In this research, we counted the number of dendritic trees in the original database and simulated the same number of virtual dendritic trees from the model. Each run was repeated 100 times to consider statistical variability. After that, a sign test was performed for each test to check if the number of rejections was significant in the 100 repetitions. A significance level of $\alpha = 0.05$ was used for all statistical tests.

Univariate Analysis For each feature, we applied three univariate statistical tests to assess whether or not the simulated and original variables were significantly different. The three tests are non-parametric so they can be applied without making any assumption about the shape of data:

- The two-sample Wilcoxon rank-sum test checks if two samples have an equal median, and can also be used to test for equal distribution of samples (Wilcoxon 1945; Fay and Proschan 2010).
- The Kolmogorov–Smirnov test checks the hypothesis that two samples come from the same underlying distribution.
- The univariate KL test uses the KL divergence value to compare two univariate marginal probability distributions $p(x)$ and $q(x)$ on $\mathbb{R}$. The KL divergence is defined in Eq. 4:

$$KL(p||q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx. \tag{4}$$

Computing the expression in Eq. 4 when a closed form cannot be found is not a trivial task, so we used the bioDist package (Ding et al. 2010) for R statistical software (R Development Core Team 2009) to estimate the KL divergence for each continuous feature. The design of the statistical test that uses the divergence values is detailed below in this section.

Multivariate Analysis The above tests are univariate, i.e. they only consider each variable independently. However, a test using all the variables at the same time would be useful as we could compare the joint probability distributions on real and simulated data. We used the multivariate KL estimator for continuous data

in Eq. 5, which is based on k-nearest neighbors (kNN) density estimation (Wang et al. 2006):

$$\widehat{KL}(p||q) = \frac{n}{N_p}\sum_{i=1}^{N_p}\log\frac{\upsilon_{D_q}(i)}{\rho_{D_p}(i)} + \log\frac{N_q}{N_p - 1}. \quad (5)$$

This expression estimates the divergence between the densities p and q from two datasets D_p and D_q with n-dimensional samples of sizes N_p and N_q, respectively. The term $\rho_{D_p}(i)$ represents the Euclidean distance from the sample $\mathbf{x}_i \in D_p$ to its nearest neighbor in $D_p \setminus \{\mathbf{x}_i\}$. On the other hand, $\upsilon_{D_q}(i)$ is the Euclidean distance from $\mathbf{x}_i \in D_p$ to its nearest neighbor in D_q. Each feature was first normalized to the interval [0,1] to avoid the different scales in the features affecting the distance measure.

We performed a statistical test based on a bootstrap method (Efron and Tibshirani 1986) that uses the KL divergence to check whether or not the simulated and real distributions are different. If the real dataset D_r contains N samples, two datasets D_{r1} and D_{r2} of size N are sampled with replacement from D_r, and both univariate (Eq. 4) and multivariate (Eq. 5) KL divergences are calculated. This process was repeated 100 times and the 95-percentile was stored as a threshold for each set of KL divergence values. When applying the KL test, the divergence value between the simulated and real dataset had to be higher than the threshold for the null hypothesis stating distributions are equal to be rejected. A sign-test was applied to check if the number of rejections was significant.

Finally, the model can be visually validated by comparing the pictures of real and virtual dendritic trees. Since it is not possible to portray 3D data through 2D projections, a website has been set up with examples of real and simulated dendritic trees (see "Information Sharing Statement").

Results

This section examines simulation results with different databases of basal dendritic trees. We established three databases of neurons according to their cortical area, and each dendritic tree was considered independently. Table 4 shows the number of dendritic trees included in each database. This section illustrates the statistical comparison for the three areas. The analysis of the Bayesian network structures focused on the M2 area, since the basal trees in this area have more complex branching patterns and dimensions (Benavides-Piccione et al. 2006). The Bayesian networks structures for the S2 and V2L/TeA areas are available at the supplementary material webpage (see "'Information Sharing Statement").

Table 4 Number of dendritic trees in each database

Region	Database	#Dendritic trees
Motor cortex	M2	104
Somatosensory cortex	S2	103
Lateral visual and association temporal cortex	V2L/TeA	156

Number of trees included in each database according to the cortical region from where the neurons were sampled

These experiments were run on an Intel Core2 Quad CPU at 2.49Ghz with 6GB RAM. The algorithms were implemented in Matlab under Windows Vista. Table 5 shows runtime intervals for each algorithm, i.e., the minimum and maximum runtime in the 12 runs (4 Bayesian networks for each one of the three cortical areas).

Analysis of Bayesian Networks

The Bayesian network structure learned from the data encodes the conditional (in)dependence relationships between the variables in the problem. The model can be validated by verifying if those relationships conform to current biological knowledge. On the other hand, a thorough analysis of the relationships could help to discover new factors involved in neuron development and dendritic morphology. Figure 6 shows the structure of the four Bayesian networks built from the M2 database, corresponding to the root segments, first-order segments, second-order segments and the other segments with higher centrifugal order. As described earlier (see "Bayesian Network Learning and Model Construction"), the ordering established during Bayesian network learning ensured that the construction variables (shaded) were always influenced by either evidence variables or other construction variables. On the other

Table 5 Runtimes of the algorithms

Algorithm	Runtime (seconds)
MWST algorithm	3.02–4.52
K2 search algorithm	4.29–9.76
Simulation algorithm (100 virtual dendritic trees)	1.53–3.86

Runtimes of the different algorithms implemented in the methods section

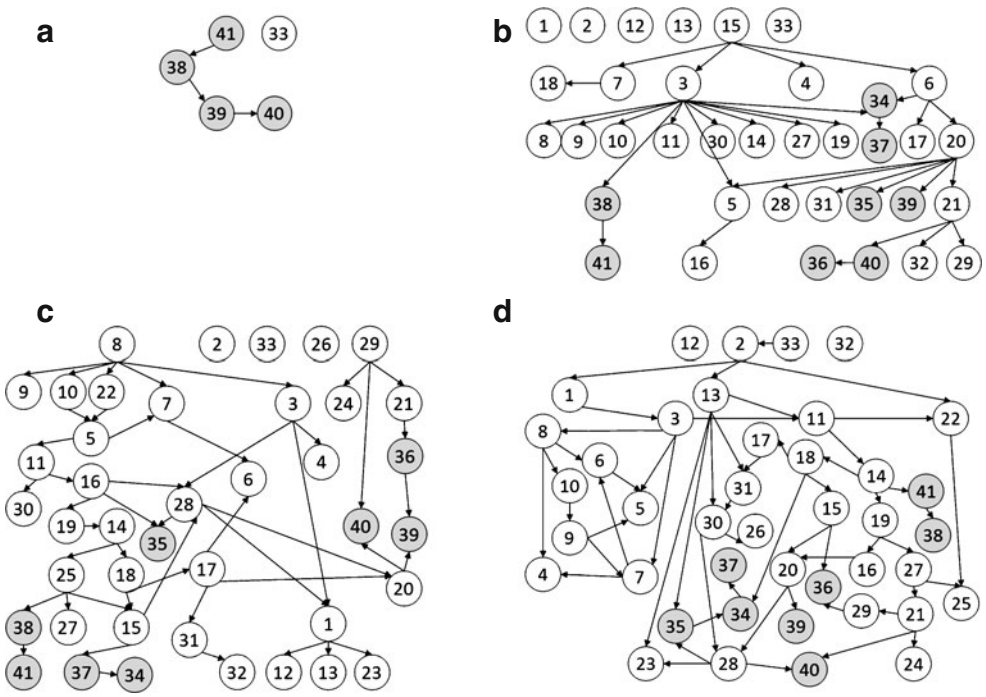

Fig. 6 Structure of the four Bayesian networks learned from the M2 database. The numbers in the *nodes* refer to the variables in Table 1. *Shaded nodes* represent construction variables and evidence variables are shown on a *white background*. The four Bayesian networks correspond to root segments (**a**), first-order segments (**b**), second-order segments (**c**) and segments with a higher order (**d**)

hand, evidence variables (with a white background) could only be influenced by other evidence variables.

In the first Bayesian network (Fig. 6a) only four construction variables and the centrifugal order evidence variable are shown. This Bayesian network was used to generate the root segment of the dendrite. There were no evidence variables to measure because no dendritic segment had yet been simulated. At this level, only one segment (the root segment) has to be sampled, and it is not possible to distinguish between left and right segments. Therefore, we decided to use variables 38–41 to encode the root segment morphology. Variables 38–41 refer to the right segment in the other Bayesian network structures (Fig. 6b–d). In the Bayesian network for first-order segments (Fig. 6b) only variables 22–26 were unavailable. Those variables could not be measured at this level because the neighboring segment could not be defined, since the root segment was the only segment that had so far been simulated. On the other hand, the other two structures (Fig. 6c and d) contained all the variables in Table 1 because the simulated part of the dendritic tree was complex enough to measure all the evidence features.

At first sight, it is clear that network complexity grew as we considered segments of a higher centrifugal order. For example, the number of relationships (line plot in Fig. 7) steadily increased from three relationships in the first to 55 relationships in the last Bayesian network. When we counted the number of variables with 0–3 parents (bars in Fig. 7), we found that variables with a higher number of parents were also more frequent in the higher-order networks. Since 0 and first-order Bayesian networks did not include all the variables, fewer relationships were expected. However, the complexity growth was higher than the number of variables added, and it also increased in the last two networks, including all the variables. The subtree from where the evidence attributes were measured grows and gets more complex as we consider high-order segments. Thus, a more widely connected network was necessary to model the relationships in the dendritic tree structure. Besides, the last Bayesian network was learned for segments with different centrifugal order and a higher variability in the data was observed. The

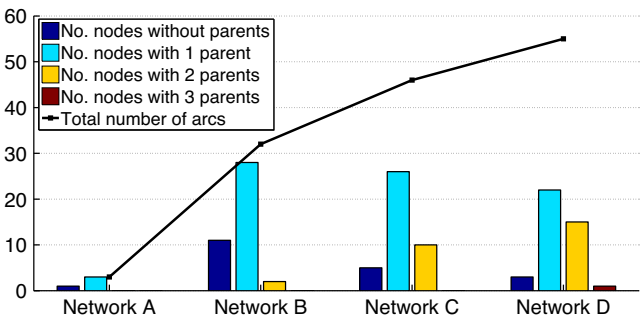

Fig. 7 Bayesian networks complexity analysis. The *line plot* shows the number of arcs in each Bayesian network from Fig. 6. The *bars* represent the number of variables with different numbers of parents in each network

Bayesian network needs more information to model this variability. This is a possible explanation for why variables had more parents.

One way to distinguish between global and centrifugal order-specific relationships is to count the number of times two variables are connected in the different networks, without taking into account the direction of the relationship. Frequent edges could represent order-independent global relationships and scarce edges could encode context-specific interactions that only appear at a certain level. Additionally, the conditional probability distributions in each variable (not shown) could be analyzed to check how the variable values change depending on the parent values. The only edge that appeared in all Bayesian networks (Fig. 6) was the one that related segment length and bifurcation occurrence for both the left (variables 34 and 37) and the right segments (variables 38 and 41) in a bifurcation. This relationship encoded the knowledge that terminal segments in basal dendrites were longer than intermediate segments. In fact, dendrites usually branch when they are close to the soma, producing short segments; whereas the segments that do not branch spread away from the soma. Figure 8 confirms this relationship, as there is a clear difference between terminal and intermediate segments at each branch order.

Segment angles also exhibited consistent relationships in the different Bayesian networks. Segment azimuth (variables 36 and 40) was related in different ways to the subdendrite azimuth (variable 21) and the parent segment azimuth (variable 29). Subdendrite inclination (variable 20) and parent segment inclination (variable 28) also influenced segment inclination (variables 35 and 39). Parent and subdendrite angles (variables 28–29 and 20–21) were frequently interrelated too. Both parent and subdendrite vectors were used by Samsonovich and Ascoli (2003) to successfully model dendritic tropism. They hypothesize that dendritic guidance might be controlled by the host cell rather than by exclusively external factors, so our model found relationships that had already been shown to be relevant. We show that segment orientation is mainly controlled by the orientation of the previous segments, reflecting how the dendrites extend away from the soma without making any sharp change of direction. We believe that the fact that our model simultaneously considers and makes a distinction between the two sibling segments at a bifurcation is possibly a key strength. Segment angles in a bifurcation are assumed to be related since they do not grow in the same direction. Thus, it could be important to consider these interactions when modeling dendritic orientation, e.g., the azimuth and inclination of different segments were related (arc from 36–9 in Fig. 6c).

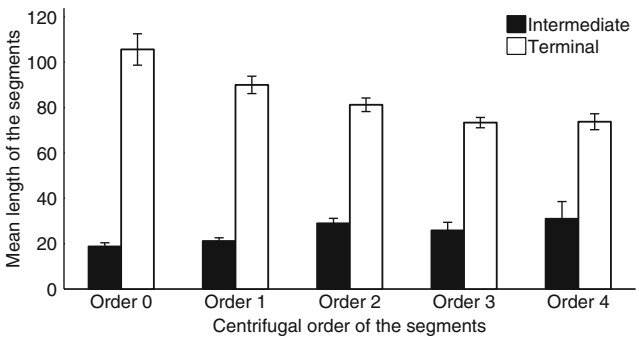

Fig. 8 Mean length of intermediate and terminal segments of real dendrites. The figure shows the mean length of the intermediate (*black*) and terminal (*white*) segments of each centrifugal order in real basal dendrites of pyramidal neurons from the M2 area. Differences between intermediate and terminal segments can be identified

Other subdendrite measurements can also directly influence some construction variables. In Fig. 6c, the width component of the subdendrite (variable 15) influenced segment bifurcation (variable 37), preventing it from splitting when this parameter showed high values. This helped to constrain tree size and could be related to resource division and competition between branches. In the above Bayesian network again, inclination angles (variable 35) were more likely to be small when subdendrite height (variable 16) took higher values. This makes the dendrite grow straight in one direction instead of the branches spreading when the segments are far from the soma and helps to model dendritic tropism. On the other hand, the subdendrite width was used to regulate segment azimuth (variable 36) and control the dendritic tree spread in the next network structure (Fig. 6d). Therefore, the dendritic trees tend to first spread rapidly when they are close to the soma and then, once they have reached a minimum size, grow straight away from the soma. Figure 9 shows two interesting facts that support these explanations. First, when considering bifurcations of the same order, complex dendritic trees with a high maximum centrifugal order (fourth and fifth order trees) have wider angles than less complex trees (first, second and third order trees). Second, when considering dendritic trees of the same complexity, the mean angle between sibling segments steadily decreases as we consider higher order bifurcations. Therefore, sibling segments grow to fill up the area defined by the angle of the first-order segment.

Subdendrite length also bounded dendritic size (arcs 14 → 41 in Fig. 6d and 3 → 38 in Fig. 6b). Note that variable 3 equaled variable 14 in Fig. 6b since only the

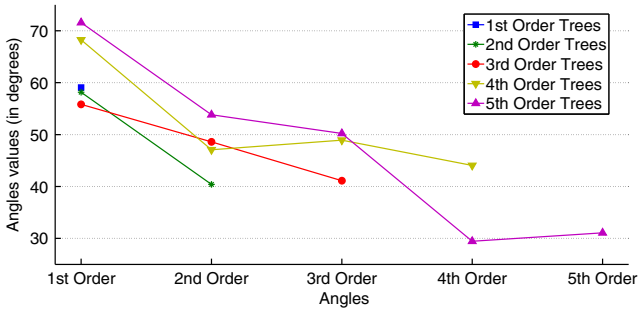

Fig. 9 Angles formed between sibling segments. Mean values of the angles formed between the two sibling segments in a bifurcation for real M2 basal dendrites. The dendritic trees are grouped by their maximum centrifugal order. Each data point shows the mean value of the angles (*in degrees*) between segments at each bifurcation node

root segment had been simulated. When the subdendrite is short, the simulation is still near the soma and more bifurcations are supposed to occur. On the other hand, a long subdendrite tends to stop the simulation by ending the dendrite with a long terminal segment.

In general, information related to the subdendrite appears to be more important for the simulation process than the whole subtree measurements because construction variables were almost always related to subdendrite variables. Summing up, segment orientation was defined by the orientation of previous segments (subdendrite) to ensure that the dendrites showed no angles causing a rough change of direction. On the other hand, the length of the path to the soma was used to control segment size and whether or not it bifurcated, bounding the total length of the dendrites. Although this could lead to the understanding that dendrites grow independently from each other, influences from other branches could be important for modeling competition and resource availability factors.

Evaluation of Features Used in the Model

We compared the original and simulated values of the features included in the Bayesian networks to check if the model accurately reproduces those variables. Figure 10 shows the results in the three areas for the datasets compared using the Kullback-Leibler multivariate test designed in the Evaluation Methodology section. At least 61 rejections were considered to be significant, as this corresponded to the rejection region limit of a sign test with 100 observations. In the three cortical areas, a significant number of rejections was found when comparing the real and simulated values in the last dataset. This could be caused by its higher data variability. This dataset was the only one that contained segments with different centrifugal order (from three to five) so the subtrees and subdendrites where evidence features were measured are also more diverse.

The univariate comparison of features could help to analyze which variables were causing such a result. The number of rejections in the three univariate tests are summarized in Fig. 11a for the M2 database. Only 14 variables out of the 123 variables (11.38%) used in the four Bayesian networks had a significant number of rejections in at least one of the tests: the rejected variables correspond to variables 5, 6, 7, 8, 9, 10, 11, 14, 16, 17, 18, 19, 26 and 30 in Table 1. If we only count the variables where at least two tests gave a significant number of rejections, nine variables are selected: 9, 10, 11, 14, 16, 18, 19, 26 and 30. All those variables belong to the fourth dataset, so they might have decreased the performance of the whole dataset when multivariate comparison was applied (Fig. 10).

The variables with rejections were evidence features, mostly from the subtree and the subdendrite. As previously mentioned, this dataset includes information from different orders, and the higher variability could be an obstacle to their simulation. We tried to decrease this variability by further dividing this dataset according to the centrifugal order of the segments. We created a dataset for each order, and learned as many Bayesian networks as the maximum order value. However, this was not a feasible solution due to the low number of higher order segments, as there has to be enough data to accurately estimate the probabilities included in the Bayesian networks.

Despite the rejections in evidence variables, the number of rejections for construction variables was never significant. In fact, only two of the rejected variables, i.e., subdendrite length (14) and subdendrite box

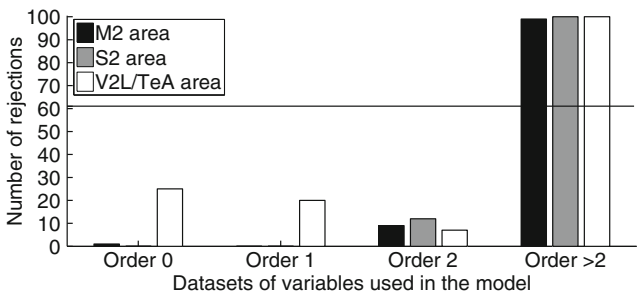

Fig. 10 Multivariate comparison of datasets used in the models. The number of rejections in the Kullback–Leibler multivariate test is shown for each of the four datasets according to the centrifugal order of the segments. The *horizontal line* represents the threshold (61 rejections) for rejections to be considered significant

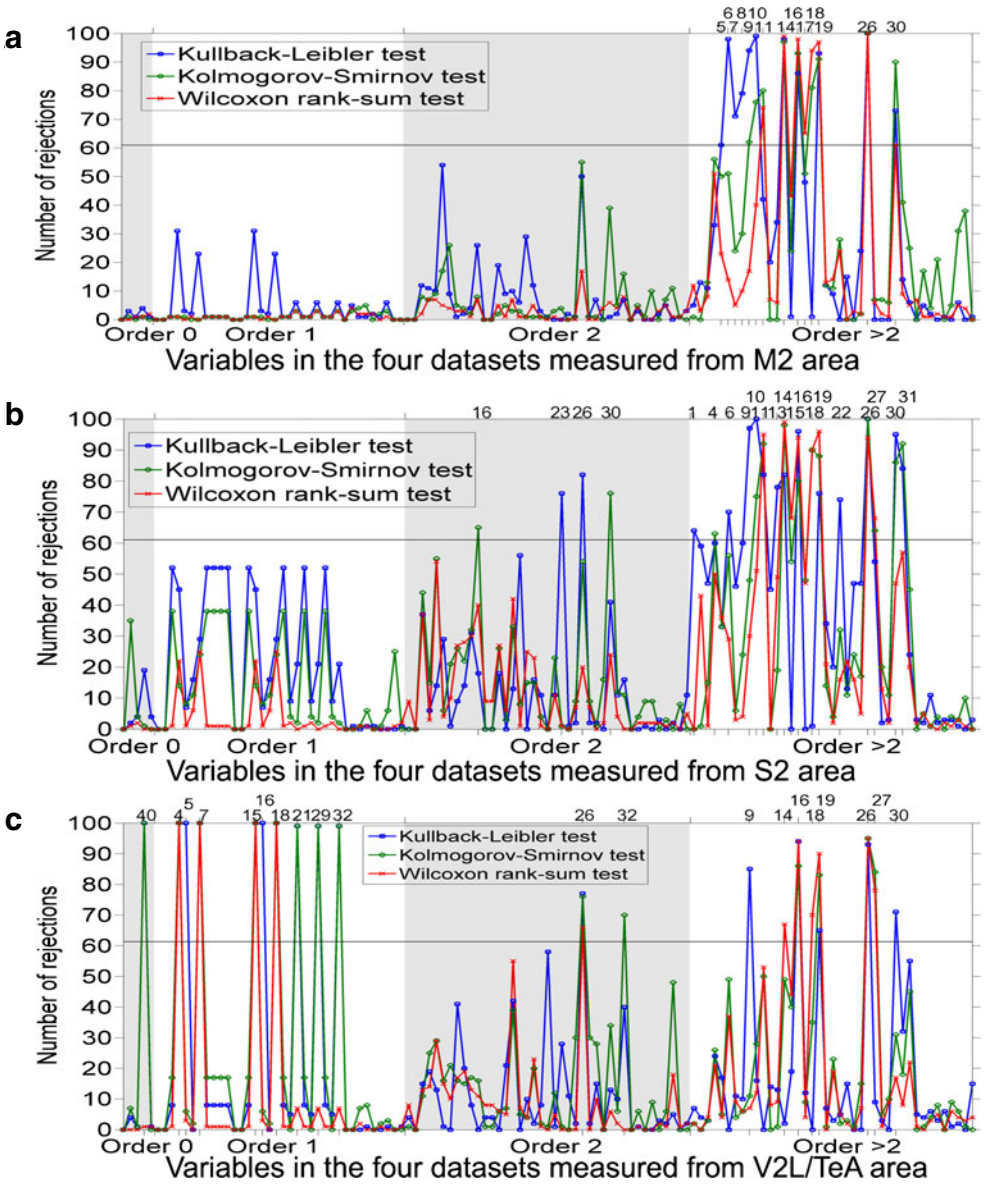

Fig. 11 Univariate comparison of variables used in the model. The number of rejections for the Kullback–Leibler test (*blue squares*), Kolmogorov–Smirnov test (*green circles*) and Wilcoxon rank-sum test (*red crosses*) are shown. The *horizontal axis* represents the variables in each Bayesian network ordered according to their number in Table 1. The numbers at the *top of the graph* indicate the rejected variables. At least 61 rejections are considered to be significant as indicated by the *horizontal line*. The figures correspond to areas M2 (**a**), S2 (**b**) and V2L/TeA (**c**)

volume (18), were directly related to any construction variable in Fig. 6d. Moreover, the simulated dendritic trees with a maximum order value greater than two were indistinguishable from real ones (see the next section). This means that the model was robust against false or heterogeneous evidence and was still able to simulate correct values for the variables used to build the virtual dendrites.

The results for the S2 area were quite similar (Fig. 11b), where 21 of 123 variables (17.07%) had a significant amount of rejections. Most of these rejections appeared in the last dataset, like M2 results. On the other hand, the tests in the V2L/TeA area (Fig. 11c) gave 20 variables with significant rejections (16.26%). Note, however, that in the first-order dataset, the subdendrite, subbranch, parent segment and root segment refer to the same information, since only one segment has been simulated. Thus, rejections for variables 21, 29 and 32 in that dataset were interrelated and also related to the azimuth rejections in the order 0 dataset (variable 40). Similarly, pairs of variables 4–15, 5–16 and 7–18 represent the same values. Moreover, when we analyzed the meaning of these variables, we could see that they were related to the azimuth value that was rejected in the root segments dataset, i.e. width, height and box volume closely depend on the azimuth value. Azimuth angle is defined in a circular domain so it is a difficult feature to model, specially when considering discrete values, and could be contributing to the rejections. However, it was the only case in which

a construction variable had a significant number of rejections.

Comparison of Emergent Parameters Not Used in the Model

A set of 12 new variables was measured from the complete real and simulated dendritic trees (see Table 6). Ascoli et al. (2001) proposed the term "emergent parameter" to refer to these global variables not included in the model that can be used to describe and compare the dendritic morphology. Model variables and "predicted variables" were previously compared by Van Pelt and Uylings (1999).

In this work, the emergent parameters are variables that can be measured at the level of the whole tree, as opposed to subdendrite or subtree variables, which are measured from a part of the tree. First, these emergent parameters are measured from the real dendritic trees. Then, the simulation step generates a set of virtual dendritic trees. Finally, the emergent parameters are calculated for the virtual dendritic trees, and both the multivariate and univariate statistical analyses are conducted again.

The number of rejections when all the real and virtual dendritic trees were tested was very high in the three areas. We hypothesized that it might be wrong to consider all the dendrites together since we could find a high diversity of dendritic morphologies. For example, comparing a dendritic tree that extends away from the soma without branching (has only a root segment) with a complex dendritic tree that branches extensively is expected to return a high number of mismatches. To check that statement we repeated the analyses comparing only real and virtual trees with the same maximum centrifugal order. The bars in Fig. 12 show a low number of rejections when we compared these subgroups

Fig. 12 Multivariate comparison of emergent parameters not included in the model. The number of rejections in the Kullback–Leibler multivariate test is shown when only trees with the same maximum centrifugal order are compared. The *horizontal line* represents the threshold as of which the rejections are considered significant

of dendritic trees, suggesting that simulated dendrites were in fact similar to real ones.

The horizontal line in Fig. 12 shows the rejection threshold (61 in a sign test with 100 observations). However, there were some runs in the 100 repetitions where no dendritic trees with a given centrifugal order value were simulated. In the M2 area, only 94 repetitions generated dendritic trees with order 5 and the rejection threshold was decreased to 57. In the S2 area, dendritic trees with a maximum centrifugal order value of 0 were simulated in 81 repetitions and trees with order 5 were generated in 90 repetitions. Therefore, the rejection threshold was set to 50 and 55 accordingly. The breaks in the horizontal line in Fig. 12 (and Fig. 13) show the above changes in the rejection threshold.

The analysis of each feature independently (Fig. 13) showed that the number of rejections also decreased when grouping the dendritic trees by their centrifugal order. The most rejected variable in the three areas and in the different groups was the dendritic tree depth (variable 7 in Table 6). Dendritic depth is highly dependent on the resolution of the *z-dimension* when tracing the neurons, which is lower than the *x* and *y* dimensions. Thus, the *z-dimension* measurements are a common source of uncertainty and errors (Steuber et al. 2004) that could also be limiting the model's ability to accurately capture dendritic depth.

In the M2 database, dendritic depth (7) and maximum Euclidean distance from the soma to the terminal tips (10) were the only variables with a significant number of rejections (Fig. 13a). Although there were more rejections for these two features than for the other features in the three tests, only the Kullback-Leibler test yielded a significant number of rejections. Figure 14 shows that, while the median value in real and

Table 6 Emergent parameters measured from the whole dendritic tree

No.	Variable
1	Degree (no. endings)
2	No. of bifurcations
3	Total length
4	Mean asymmetry index
5	Dendritic tree width
6	Dendritic tree height
7	Dendritic tree depth
8	Box volume
9	Max distance between nodes
10	Max distance to soma
11	Max centrifugal order
12	Min centrifugal order

Emergent parameters not included in the model. The values of these features are used to describe and compare the morphology of the whole dendritic trees

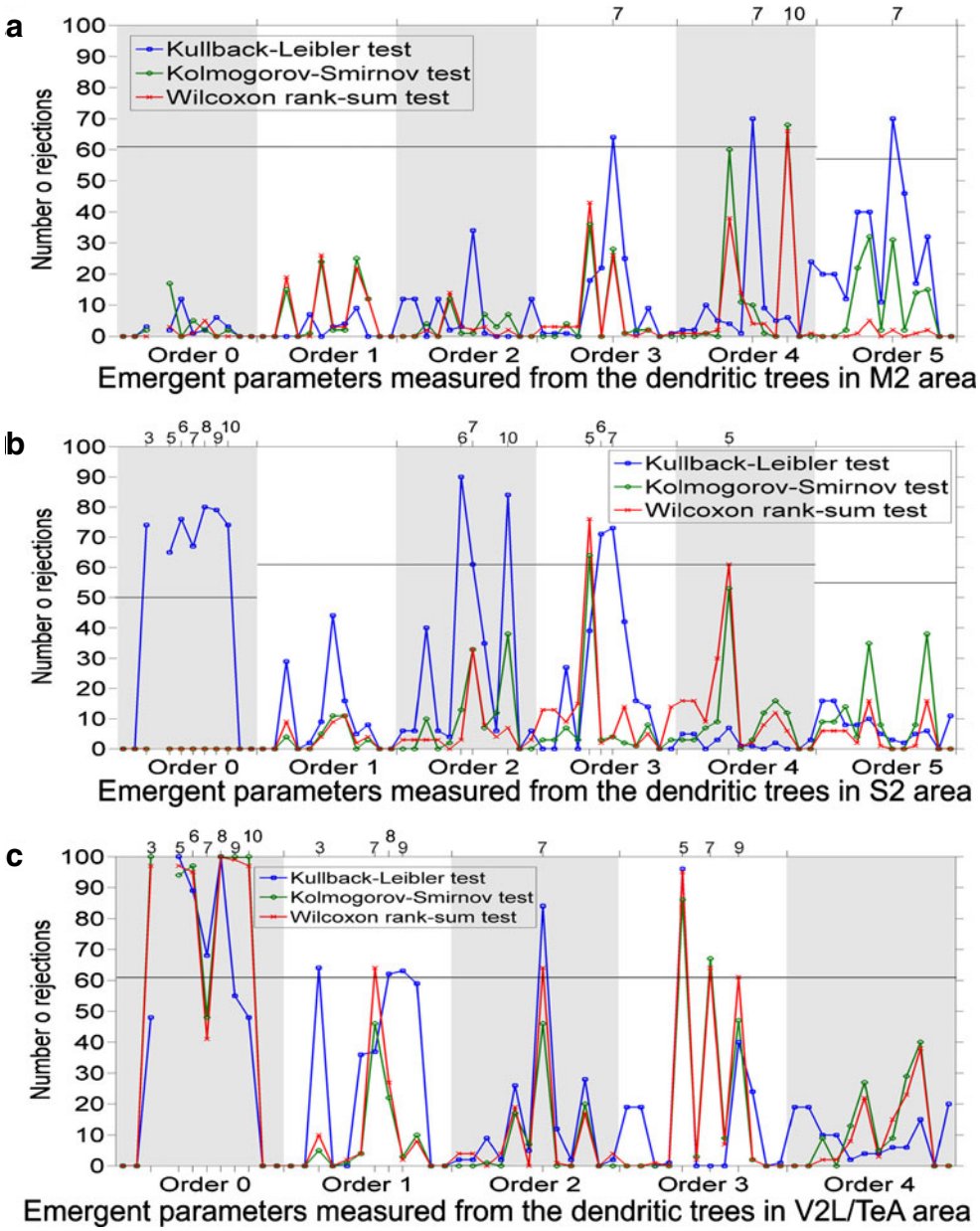

Fig. 13 Univariate comparison of emergent parameters not used in the model. Results for 12 emergent parameters (Table 6) using univariate tests: Kullback–Leibler (*blue squares*), Kolmogorov–Smirnov (*green circles*) and Wilcoxon rank-sum (*red crosses*). This figure shows the number of rejections when the dendritic trees are grouped according to their maximum centrifugal order. The *horizontal line* indicates the number as of which rejections are considered significant. The figures correspond to areas M2 (**a**), S2 (**b**) and V2L/TeA (**c**)

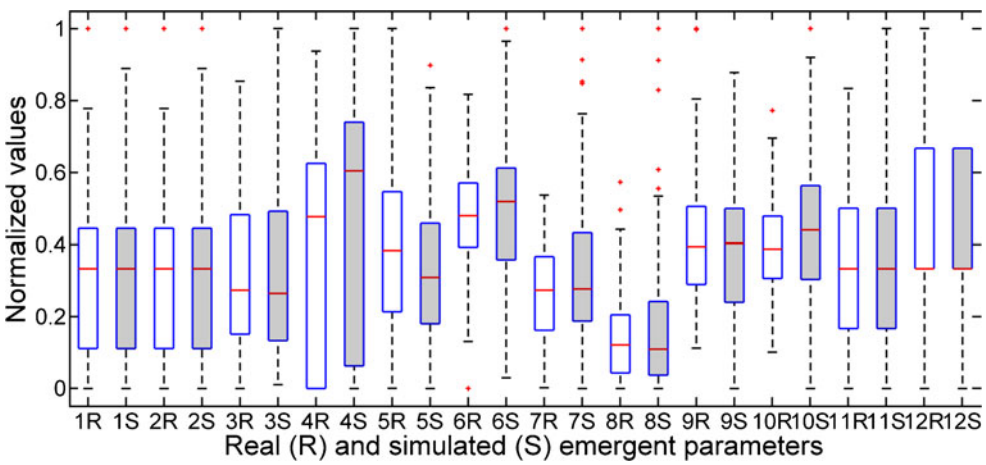

Fig. 14 Boxplots for the real and simulated values of the emergent parameters. The *boxes* show the values when all the dendritic trees from the M2 area are compared and no subgroups are considered. The numbers refer to the variables in Table 6, whereas R and S stand for real (with *white background*) and simulated (*shaded*) data, respectively

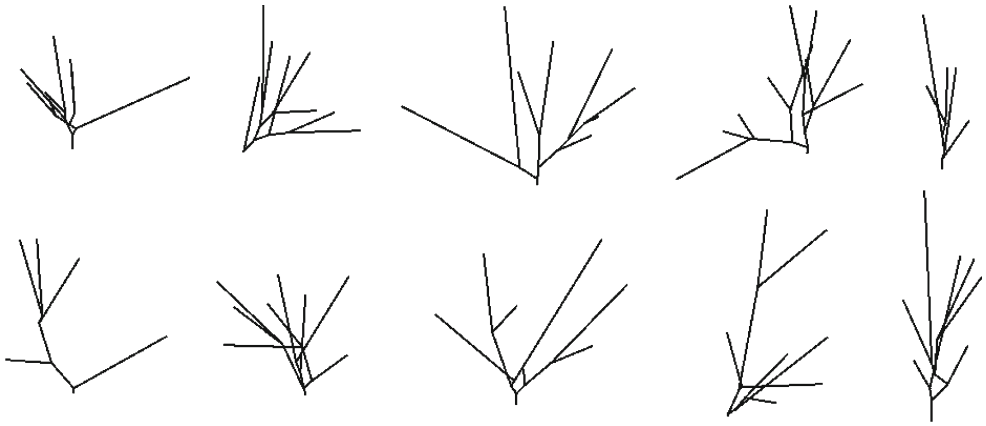

Fig. 15 Examples of real and virtual dendritic trees. The figure shows 2D projections of real (*top*) and simulated (*bottom*) main trunks of basal dendrites from M2 pyramidal neurons. Virtual dendrites are similar to the real ones and show some distinctive traits of basal dendritic trees from pyramidal cells

simulated data was quite similar, the whiskers in the simulated data were wider, and the number of outliers was also higher. These mismatches could be due to the discretization process, since it might attach a higher probability to some infrequent data.

In the S2 and V2L/TeA areas (Fig. 13b and c), a high number of rejections were identified in zero-order dendrites. There were very few such dendrites in the S2 area, where only 3 out of 103 dendritic trees were selected (2.91%), so the tests might be influenced by this shortage of data. On the other hand, zero-order dendrites were more frequent in the V2L/TeA area (27 out of 156, 17.31%). As previously noted in the evaluation of the model variables, the difficulty of modeling root segment azimuth in this area might be causing these rejections.

Visual Comparison

A visual comparison between real and simulated dendrites (Fig. 15) confirms that the model is able to generate virtual dendrites that show some of the defining features of pyramidal basal dendritic trees, e.g., long terminal segments or more frequent bifurcations at closer distances from the soma. 3D morphologies can be accessed (see "Information Sharing Statement") to further check this correspondence and allow for a qualitative validation of the simulated dendritic trees.

Discussion

We have presented a new reconstruction approach for the simulation of 3D dendritic morphology. The methodology uses Bayesian networks, a probabilistic graphical model, to capture the interactions between the variables in the problem domain. A complete set of evidence and construction variables is measured from the dendrites, and a learning algorithm is applied to find the structure and estimate the probability distributions included in the Bayesian networks. Then, a simulation algorithm is used to build the virtual dendrites sampling values from the Bayesian networks and a thorough evaluation is performed to show the model's ability to generate realistic dendrites.

Bayesian networks encode a factorization of the joint probability distribution over the variables, and can model an extensive set of variables by exploiting the local properties of the probability distributions. Discrete Bayesian networks have a solid mathematical background, their theoretical properties have been widely studied and applications in different fields have been successful (Pourret et al. 2008). In this work, the analyses of the robustness of the network structures (Friedman et al. 1999) and the minimum sample size that guarantees that there is a high probability of the learned and the true distributions being close to each other (Friedman and Yakhini 1996) confirmed the stability of the models (see the "Information Sharing Statement").

This proposal has an important advantage over previous research because the relationships between the variables in the model are found by directly mining the data. Therefore, it is not necessary to previously specify all the interactions between the morphological features. Instead, we have a computer algorithm find the model that best fits the data. Applying machine learning algorithms to discover dependencies between parameters and simulate virtual morphologies is useful when complete information on the structure of the problem is not available. Since the model is learned from the data, the methodology can be applied to any neuronal class without the need for any modification. We should note that the model presented here is implicitly conditioned by the centrifugal order of the segments. The datasets are split during the data acquisition and preparation step.

This means that different Bayesian networks (different relationships and probability distributions) are learned for different orders. This is a commonly used feature in the literature (e.g., Devaud et al. 2000; Donohue and Ascoli 2008; Anwar et al. 2009). However, once the datasets have been created, the Bayesian network learning process is completely data-driven.

Other reconstruction models either do not explicitly include relationships between variables, which is an overly simplistic approach, or include predefined relationships (e.g., Lindsay et al. 2007; Donohue and Ascoli 2008; Koene et al. 2009; Anwar et al. 2009). However, it is very difficult to find common relationships that apply to every neuronal class, and, if this is possible, their validity would have to be checked every time a new class is modeled. On the other hand, if a model has explicit relationships to accurately represent a neuronal class, changes will have to be made to fit the model to other classes that do not comply with those hypotheses. The methodology proposed here is generally applicable, whereas the models are data specific. Common interactions among neuronal classes can be identified by looking for similarities between the different models, whereas rare relationships could capture unique features in a given class. This approach addresses knowledge extraction from the models in a more straightforward way.

Furthermore, although reconstructions of neurons from diverse classes are available online in several public databases (Ascoli et al. 2007), the high tracing process heterogeneity between labs and researchers can negatively affect the model's behavior (Ascoli 2007). In our study, we have used data from basal dendritic arbors of pyramidal cells from the same laboratory. Additionally, they are fully reconstructed basal dendritic arbors, so they are particularly valuable for validating simulated virtual neurons. On the other hand, segment diameter is commonly used in the literature to determine segment length or branching probability, as it is highly correlated with structural factors like the microtubule density (Hillman 1979). Since measuring segment diameter is prone to errors and noise (Steuber et al. 2004), diameter values were not considered in the reconstruction of the real neurons; and, consequently, have not been included in our simulation model. All these factors will have to be taken into account when we study dendrites from other neuronal classes traced in different labs.

Our model comprises four Bayesian networks that encode the relationships between the variables at different levels of the dendritic tree. A number of stages have been described during neuron development, and different parts might have different morphologies and be regulated by different factors (Donohue and Ascoli 2005a). Making a distinction between segments at different centrifugal orders is useful for identifying context-specific interactions. In fact, the Bayesian networks found for the different levels are quite different (see the "Analysis of Bayesian Networks"). Bayesian multinets (Geiger and Heckerman 1996), a generalization of Bayesian networks where different relationships can be considered depending on the values of some variables are also worth investigating for this problem. These models can identify the changes in the problem structure for different parts of the dendrites without the need for more than one Bayesian network.

The analysis of the Bayesian networks showed some interesting results. Segment length is always related to the branching probability, indicating a statistical difference between intermediate and terminal segments (Uylings et al. 1986; Larkman 1991; Van Veen and Van Pelt 1993). In fact, some of the models in previous works separate root, intermediate and terminal segments when modeling or evaluating segment length (e.g., Lindsay et al. 2007; Torben-Nielsen et al. 2008b; Anwar et al. 2009). In our case, that distinction was inferred automatically from the data by the learning algorithm and not predefined on the model. Segment orientation is mainly controlled by parent segment and subbranch azimuth and inclination angles to successfully capture dendritic tropism. Other relationships with subbranch measurements were found, supporting the idea that host cell influence might be more important in neuron development and neurite outgrowth than expected. It is important to keep in mind that the pattern of dendritic arborization is the result of a complex interaction between intrinsic genetic programs and external modulators, e.g., neurotransmitters or patterns of activity (Kaufmann and Moser 2000; Cline 2001; Benavides-Piccione et al. 2004; Wen et al. 2009; Ballesteros-Yáñez et al. 2010). Thus, adding variables to model some environmental factors and checking the relationships that are established with them will help to clarify this issue (Torben-Nielsen et al. 2008b). If relationships between construction variables and subdendrite or subtree evidence variables still occur in the model when environmental features are included, it will suggest that intrinsic factors are relevant and cannot be ignored.

The results were fully evaluated to verify that the virtual dendrites were accurate. Univariate and multivariate analyses were conducted to compare model variables and emergent features. A statistical test using Kullback-Leibler divergence was designed based on a bootstrapping method to compare both univariate and joint probability distributions over the variables.

Applying several statistical tests proved to be useful because some of the mismatches were only detected by one method. The univariate KL test was applied along with Kolmogorov-Smirnov test and Wilcoxon rank-sum test to check if the values of the real and virtual dendrites came from the same distribution. On the other hand, the multivariate KL test allows for a comparison of the joint probability distribution over all the features at the same time. As far as we know, this is the first time that such a multivariate test has been used to evaluate the joint probability distribution over a set of variables that describe dendritic morphology. Usually, univariate statistical tests that compare each variable independently are used (Lindsay et al. 2007; Torben-Nielsen et al. 2008b), or plots are visually inspected to evaluate bivariate or conditional densities (Donohue and Ascoli 2005a; Koene et al. 2009). More complex evaluation methods are still needed in order to accurately compare dendritic morphology. Measures for quantifying dendritic branching patterns and tree structures are rather simple and not very informative, although some efforts are being made to propose such advanced measures, as in Heumann and Wittum (2009).

Future work will focus on using Bayesian networks that can directly manage continuous features without the need for discretization. Current state-of-the-art algorithms for hybrid Bayesian networks, which include both discrete and continuous variables, usually impose some assumptions on the shape of the data: Gaussian distributions, mixtures of truncated exponentials (Romero et al. 2006), etc. However, our data may not fit any of these distributions, so non-parametric alternatives will also be considered. Learning and simulating from this kind of models are difficult and interesting tasks, as illustrated by the amount of research in the area.

The Bayesian network computational model may be useful for identifying new aspects of the morphological specializations of pyramidal cells related to cortical functions or evolution. For example, a remarkable characteristic of pyramidal cells is the huge variations in the pattern of their dendritic arborization in different cortical layers, areas and species (Jacobs and Scheibel 2002; Elston 2007). Future work will address the generation of the whole basal arborization of pyramidal cells. This implies studying the distribution of the dendritic trees, i.e., number of trees, orientation, complexity, interactions between trees, etc. This work will include a complete analysis and comparison of the models presented here against the models for the complete basal arborizations. This will help to identify whether the differences between basal dendrites in the cortical areas (Benavides-Piccione et al. 2006) are due to significant differences in the individual trees or in their distributions, i.e., how the dendritic arbor is built from the dendritic trees. Additionally, we would like to extend this research using neurons grouped by species, cortical area, layer, age, etc.

Also, subgroups of dendrites can be formed using other features apart from maximum centrifugal order to evaluate real and simulated data, e.g., number of terminal segments, total dendritic length, etc. Since there is no commonly accepted way to make this division, the results might be different when other features are used.

Finally, our model may also help to develop algorithmic methods for repairing or recovering the missing parts of the neuron morphology (Markram 2006; Anwar et al. 2009) using different sampling methods. Such a repair method would make it possible to examine pyramidal neurons in coronal sections and cover both the apical and basal dendritic systems, helping to generate a complete picture of pyramidal cell morphology.

We conclude that probabilistic graphical models are useful techniques for simulating 3D dendrites, as illustrated by this proposal based on Bayesian networks. We hope that further research in this area and the implementation of more complex models (Bayesian multinets, hybrid Bayesian networks, causal networks, etc.) will provide an excellent tool for examining dendritic structure.

Information Sharing Statement

A website has been set up containing all the information about the experiments reported in this paper. The Bayesian networks learned for the different cortical areas (M2, S2 and V2L/TeA) and the results of the statistical evaluation of the models are available. An analysis of the stability of the network structures obtained by the learning algorithm and the study of the sample size are included. The results of the tests performed to assess the quality of the real and simulated data and to evaluate the multivariate KL test can also be accessed. Examples of the 3D dendritic morphologies of real and virtual dendrites from the three areas can be explored online using an applet based on CVAPP software (Cannon et al. 1998). The code has mainly been developed using Matlab. The website is available at http://www.dia.fi.upm.es/~concha/dendriticsimulation.

Acknowledgements This work has been supported by Spanish Science and Innovation Ministry, Cajal Blue Brain Project (C080020-09), TIN2010-20900-C04-04 Project and Consolider Ingenio 2010-CSD2007-00018. PL-C is supported by a FPU Fellowship from the Spanish Education Ministry.

References

Anwar, H., Riachi, I., Hill, S., Schürmann, F., & Markram, H. (2009). An approach to capturing neuron morphological diversity. In E. De Schutter (Ed.), *Computational modeling methods for neuroscientists* (pp. 211–232). The MIT Press.

Ascoli, G. A. (2007). Successes and rewards in sharing digital reconstructions of neuronal morphology. *Neuroinformatics, 5*, 154–160.

Ascoli, G. A., & Krichmar, J. L. (2000). L-neuron: A modeling tool for the efficient generation and parsimonious description of dendritic morphology. *Neurocomputing, 32–33*, 1003–1011.

Ascoli, G. A., Krichmar, J. L., Nasuto, S., & Senft, S. (2001). Generation, description and storage of dendritic morphology data. *Philosophical transactions of the Royal Society of London. Series B, Biological Sciences, 356*, 1131–1145.

Ascoli, G. A., Donohue, D. E., & Halavi, M. (2007). Neuromorpho.org: A central resource for neuronal morphologies. *Journal of Neuroscience, 27*(35), 9247–9251.

Ascoli, G. A., Alonso-Nanclares, L., Anderson, S., Barrionuevo, G., Benavides-Piccione, R., Burkhalter, A., et al. (2008). Petilla terminology: Nomenclature of features of gabaergic interneurons of the cerebral cortex. *Nature Reviews. Neuroscience, 9*(7), 557–568.

Ballesteros-Yáñez, I., Benavides-Piccione, R., Bourgeois, J., Changeux, J., & DeFelipe, J. (2010). Alterations of cortical pyramidal neurons in mice lacking high-affinity nicotinic receptors. *Proceedings of the National Academy of Sciences of the United States of America, 107*(25), 11,567–11,572.

Benavides-Piccione, R., Ballesteros-Yáñez, I., Martínez de Legrán, M., Elston, G., Estivill, X., Fillat, C., et al. (2004). On dendrites in Down syndrome and DS murine models: A spiny way to learn. *Progress in Neurobiology, 74*, 111–126.

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

Brown, K. M., Gillette, T. A., & Ascoli, G. A. (2008). Quantifying neuronal size: Summing up trees and splitting the branch difference. *Seminars in Cell & Developmental Biology, 19*, 485–493.

Cannon, R., Turner, D., Pyapali, G., & Wheal, H. (1998). An online archive of reconstructed hippocampal neurons. *Journal of Neuroscience Methods, 84*, 49–54.

Chen, J. Y. (2009). A simulation study investigating the impact of dendritic morphology and synaptic topology on neuronal firing patterns. *Neural Computation, 22*(4), 1086–1111.

Cline, H. (2001). Dendritic arbor development and synaptogenesis. *Current Opinion in Neurobiology, 11*(1), 118–126.

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9*, 309–347.

DeFelipe, J. (2008). The neuroanatomist's dream, the problems and solutions, and the ultimate aim. *Frontiers in Neuroscience, 2*, 10–12.

DeFelipe, J., & Fariñas, I. (1992). The pyramidal neuron of the cerebral cortex: Morphological and chemical characteristics of the synaptic inputs. *Progress in Neurobiology, 39*, 563–607.

Devaud J. M., Quenet, B., Gascuel, J., & Masson, C. (2000). Statistical analysis and parsimonious modelling of dendrograms of in vitro neurones. *Bulletin of Mathematical Biology, 62*, 657–674.

Ding, B., Gentleman, R., & Carey, V. (2010). *bioDist: Different distance measures*. R package version 1.18.0.

Donohue, D. E., & Ascoli, G. A. (2005a). Local diameter fully constrains dendritic size in basal but not apical trees of CA1 pyramidal neurons. *Journal of Computational Neuroscience, 19*(2), 223–238.

Donohue, D. E., & Ascoli, G. A. (2005b). Models of neuronal outgrowth. In S. Koslow, & S. Subramaniam (Eds.), *Databasing the brain: From data to knowledge* (pp. 303–326). New York: Wiley.

Donohue, D. E., & Ascoli, G. A. (2008). A comparative computer simulation of dendritic morphology. *PLoS Computational Biology, 4*(6). doi:10.1371/journal.pcbi.1000089.

Efron, B., & Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science, 1*(1), 54–75.

Elston, G. (2007). Specializations in pyramidal cell structure during primate evolution. In J. Kaas, & T. Preuss (Eds.), *Evolution of nervous systems* (pp. 191–242). Academic: Oxford.

Elston, G., & Rosa, M. (1997). The occipito-parietal pathway of the macaque monkey: Comparison of pyramidal cell morphology in layer III of functionally related cortical visual areas. *Cerebral Cortex, 7*(5), 432–452.

Fay, M. P., & Proschan, M. A. (2010). Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys, 4*, 1–39.

Feldman, M. (1984). Morphology of the neocortical pyramidal neuron. In A. Peters, & E. Jones (Eds.), *Cerebral cortex. Cellular components of the cerebral cortex* (Vol. 1, pp. 201–253). New York: Plenum Press.

Friedman, N., & Yakhini, Z. (1996). On the sample complexity of learning Bayesian networks. In *Proceedings of the twelfth conference on uncertainty in artificial intelligence (UAI 96)* (pp. 274–282).

Friedman, N., Goldszmith, M., & Wyner, A. (1999). Data analysis with Bayesian networks: A bootstrap approach. In *Proceedings of the fifteenth conference on uncertainty in artificial intelligence (UAI 99)* (pp. 196–205).

Geiger, D., & Heckerman, D. (1996). Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence, 82*, 45–74.

Glaser, J., & Glaser, E. (1990). Neuron imaging with neurolucida—a PC-based system for image combining microscopy. *Computerized Medical Imaging and Graphics, 14*(5), 307–317.

Hamilton, P. (1993). A language to describe the growth of neurites. *Biological Cybernetics, 68*(6), 559–565.

Häusser, M., & Mel, B. (2003). Dendrites: Bug or feature? *Current Opinion in Neurobiology, 13*(3), 372–383.

Heckerman, D. (1996). A tutorial on learning with Bayesian networks. Tech. Rep. MSR-TR-95-06, Microsoft Corporation.

Hentschel, H. G., & van Ooyen, A. (1999). Models of axon guidance and bundling during development. *Proceedings of the Royal Society of London. Series B, Biological Sciences, 266*, 2231–2238.

Heumann, H., & Wittum, G. (2009). The tree-edit-distance, a measure for quantifying neuronal morphology. *Neuroinformatics, 7*(3), 179–190.

Hillman, D. (1979). Neuronal shape parameters and substructures as a basis of neuronal form. In F Schmitt (Ed.), *The neurosciences, 4th study program* (pp. 477–498). MIT Press.

Jacobs, B., & Scheibel, A. (2002). Regional dendritic variation in primate cortical pyramidal cells. In A. Schüz, & R. Miller (Eds.), *Cortical areas: Unity and diversity* (pp. 111–131). CRC Press.

Kaufmann, W. E., & Moser, H. W. (2000). Dendritic anomalies in disorders associated with mental retardation. *Cerebral Cortex, 10*(10), 981–991.

Koch, C., & Segev, I. (2000). The role of single neurons in information processing. *Nature Neuroscience, 3*, 1171–1177.

Koch, C., Poggio, T., & Torres, V. (1982). Retinal ganglion cells: A functional interpretation of dendritic morphology. *Proceedings of the Royal Society of London. Series B, Biological Sciences, 298*(1090), 227–263.

Koene, R. A., Tijms, B., van Hees, P., Postma, F., de Ridder, A., Ramakers, G. J., et al. (2009). Netmorph: A framework for the stochastic generation of large scale neuronal networks with realistic neuron morphologies. *Neuroinformatics, 7*(3), 195–210.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models. Principles and techniques*. The MIT Press.

Krause, P. J. (1998). Learning probabilistic networks. *Knowledge Engineering Review, 13*(4), 321–351.

Kullback, S., & Leibler, R. (1951). On information and sufficiency. *Annals of Mathematical Statistics, 22*(1), 79–86.

Larkman, A. (1991). Dendritic morphology of pyramidal neurones of the visual cortex of the rat: I. Branching patterns. *Journal of Comparative Neurology, 306*(2), 307–319.

Leray, P., & Francois, O. (2006). *BNT structure learning package: Documentation and experiments*. Tech. Rep. FRE CNRS 2645, Laboratoire PSI—INSA Rouen.

Li, G. H., & Qin, C. D. (1996). A model for neurite growth and neuronal morphogenesis. *Mathematical Biosciences, 132*(1), 97–110.

Lindsay, K. A., Maxwell, D. J., Rosenberg, J. R., & Tucker, G. (2007). A new approach to reconstruction models of dendritic branching patterns. *Mathematical Biosciences, 205*(2), 271–296.

Luczak, A. (2006). Spatial embedding of neuronal trees modeled by diffusive growth. *Journal of Neuroscience Methods, 157*(1), 132–141.

Mainen, Z. F., & Sejnowski, T. J. (1996). Influence of dendritic structure on firing pattern in model neocortical neurons. *Nature, 382*, 363–366.

Markram, H. (2006). The blue brain project. *Nature Reviews. Neuroscience, 7*(2), 153–160.

McAllister, A. K. (2000). Cellular and molecular mechanisms of dendrite growth. *Cerebral Cortex, 10*(10), 963–973.

Miina, J., & Pukkala, T. (2002). Application of ecological field theory in distance-dependent growth modelling. *Forest Ecology and Management, 161*, 101–107.

Murphy, K. (2001). The Bayes net toolbox for Matlab. In E. Wegman, A. Braverman, A. Goodman, & P Smyth (Eds.), *Computing science and statistics. Proceedings of the 33rd symposium on the interface* (Vol. 33, pp. 331–350).

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.

Pourret, O., Naïm, P., & Marcot, B. (2008). *Bayesian networks: A practical guide to applications*. Wiley.

R Development Core Team (2009). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria.

Robert, M. E., & Sweeney, J. D. (1997). Computer model: Investigating the role of filopodia-based steering in experimental neurite galvanotropism. *Journal of Theoretical Biology, 188*(3), 277–288.

Romero, V., Rumí, R., & Salmerón, A. (2006). Learning hybrid Bayesian networks using mixtures of truncated exponentials. *International Journal of Approximate Reasoning, 42*, 54–68.

Rozenberg, G., & Salomaa, A. (1980). *The mathematical theory of L-systems*. New York: Academic Press.

Samsonovich, A. V., & Ascoli, G. A. (2003). Statistical morphological analysis of hippocampal principal neurons indicates cell-specific repulsion of dendrites from their own cell. *Journal of Neuroscience Research, 71*(2), 173–187.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*(2), 461–464.

Scott, E. K., & Luo, L. (2001). How do dendrites take their shape? *Nature Neuroscience, 4*(4), 359–365.

Shepherd, G. M. (ed) (2004). *The synaptic organization of the brain* (5th edn). Oxford University Press.

Spruston, N. (2008). Pyramidal neurons: Dendritic structure and synaptic integration. *Nature Reviews. Neuroscience, 9*(3), 206–221.

Steuber, V., De Schutter, E., & Jaeger, D. (2004). Passive models of neurons in the deep cerebellar nuclei: The effect of reconstruction errors. *Neurocomputing, 58–60*, 563–568.

Sumida, A., Terazawa, I., Togashi, A., & Komiyama, A. (2002). Spatial arrangement of branches in relation to slope and neighbourhood competition. *Annals of Botany, 82*, 301–310.

Torben-Nielsen, B., Tuyls, K., & Postma, E. O. (2006). Shaping realistic neuronal morphologies: An evolutionary computation method. In *International joint conference on neural networks (IJCNN2006)* (pp. 573–580). Vancouver, Canada.

Torben-Nielsen, B., Tuyls, K., & Postma, E. O. (2007). On the neuronal morphology-function relationship: A synthetic approach. In *Knowledge discovery and emergent complexity in bioinformatics, LNBI*. (Vol. 4366, pp. 135–149). Springer.

Torben-Nielsen, B., Tuyls, K., & Postma, E. O. (2008a). Evolneuron: Neuronal morphology generation. *Neurocomputing, 71*, 963–972.

Torben-Nielsen, B., Vanderlooy, S., & Postma, E. O. (2008b). Non-parametric algorithmic generation of neuronal morphologies. *Neuroinformatics, 6*, 257–277.

Uylings, H. B., & van Pelt, J. (2002). Measures for quantifying dendritic arborizations. *Network: Computation in Neural Systems, 13*, 397–414.

Uylings, H. B., Ruiz-Marcos, A., & Van Pelt, J. (1986). The metric analysis of three-dimensional dendritic tree patterns: A methodological review. *Journal of Neuroscience Methods, 18*, 127–151.

Van Pelt, J., & Uylings, H. B. (1999). Modeling the natural variability in the shape of dendritic trees: Application to basal dendrites of small rat cortical layer 5 pyramidal neurons. *Neurocomputing, 26–27*, 305–311.

Van Pelt, J., & Uylings, H. B. (2005). Natural variability in the geometry of dendritic branching patterns. In G. Reeke, R. Poznanski, K. Lindsay, J. Rosenberg, & O. Sporns (Eds.), *Modeling in the neurosciences: From biological systems to neuromimetic robotics* (pp. 89–116). CRC Press.

Van Pelt, J., van Ooyen, A., & Uylings, H. B. (2001). Modeling dendritic geometry and the development of nerve connections. In E. De Schutter (Ed.), *Computational neuroscience: Realistic modeling for experimentalists* (pp. 179–208). CRC Press.

Van Veen, M. P., & Van Pelt, J. (1993). Terminal and intermediate segment lengths in neuronal trees with finite length. *Bulletin of Mathematical Biology, 55*, 277–294.

Verwer, R., van Pelt, J., & Uylings, H. B. (1992). An introduction to topological analysis of neurones. In M Stewart (Ed.), *Quantitative methods in neuroanatomy* (pp. 292–323). John Wiley and Sons.

Vetter, P., Roth, A., & Häusser, M. (2001). Propagation of action potentials in dendrites depends on dendritic morphology. *Journal of Neurophysiology, 85*(2), 926–937.

Wang, Q., Kulkarni, S. R., & Verdú, S. (2006). A nearest-neighbor approach to estimating divergence between contin-

uous random vectors. In *IEEE international symposium on information theory (ISIT 2006)* (pp. 242–246).

Wen, Q., Stepanyants, A., Elston, G., Grosberg, A., & Chklovskii, D. (2009). Maximization of the connectivity repertoire as a statistical principle governing the shapes of dendritic arbors. *Proceedings of the National Academy of Sciences of the United States of America, 106*(30), 12,536–12,541.

White, E. (1989). *Cortical circuits: Synaptic organization of the cerebral cortex. Structure, function and theory*. Boston: Birkhauser.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin 1*(6), 80–83.

Yuste, R., & Bonhoeffer, T. (2004). Genesis of dendritic spines: Insights from ultrastructural and imaging studies. *Nature Reviews. Neuroscience, 5*, 24–34.