International Journal of Approximate Reasoning 55 (2014) 3–22

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Bayesian network modeling of the consensus between experts: An application to neuron classification

Pedro L. López-Cruz [a,*], Pedro Larrañaga [a], Javier DeFelipe [b], Concha Bielza [a]

[a] Computational Intelligence Group, Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid, Spain

[b] Instituto Cajal (CSIC) and Laboratorio de Circuitos Corticales (Centro de Tecnología Biomédica), Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Article history:
Available online 2 April 2013

Keywords:
Bayesian networks
Bayesian multinets
Expert consensus
Neuron classification

ABSTRACT

Neuronal morphology is hugely variable across brain regions and species, and their classification strategies are a matter of intense debate in neuroscience. GABAergic cortical interneurons have been a challenge because it is difficult to find a set of morphological properties which clearly define neuronal types. A group of 48 neuroscience experts around the world were asked to classify a set of 320 cortical GABAergic interneurons according to the main features of their three-dimensional morphological reconstructions. A methodology for building a model which captures the opinions of all the experts was proposed. First, one Bayesian network was learned for each expert, and we proposed an algorithm for clustering Bayesian networks corresponding to experts with similar behaviors. Then, a Bayesian network which represents the opinions of each group of experts was induced. Finally, a consensus Bayesian multinet which models the opinions of the whole group of experts was built. A thorough analysis of the consensus model identified different behaviors between the experts when classifying the interneurons in the experiment. A set of characterizing morphological traits for the neuronal types was defined by performing inference in the Bayesian multinet. These findings were used to validate the model and to gain some insights into neuron morphology.

1. Introduction

The morphologies, molecular features and electrophysiological properties of neuronal cells are extremely variable [1–4]. Neuronal morphology is a key feature in the study of brain circuits, as it is highly related to information processing and functional identification. Except for some special cases, this variability makes it hard to find a set of features that unambiguously define a neuronal type [3]. In addition, there are distinct types of neurons in particular regions of the brain. Indeed, neurons in the cerebral cortex can be classified into two main categories based on their morphology: pyramidal neurons and interneurons (Fig. 1). In general, pyramidal neurons are excitatory (glutamatergic) cells which display spines in their dendrites and have an axon which projects out of the white matter. Their name refers to the pyramidal shape of their soma. Interneurons are cells with short axons that do not leave the white matter and their dendrites show few or no spines. These interneurons appear to be mostly GABAergic (inhibitory) and constitute ∼15–30% of the total neuron population, but they display chemical, physiological and synaptic heterogeneity [3]. Thus, the identification of classes and subclasses of interneurons is clearly critical for gaining a better understanding of how these cell shapes relate to cortical functions in both health and disease. This paper focuses on GABAergic interneurons, which also show a remarkable morphological variability between species, layers and areas [5]. The Internet has made it possible for researchers to share digital three-dimensional reconstructions of neuronal morphology in publicly accessible databases [6,7]. With such amount of available data, a com-

* Corresponding author.
E-mail addresses: pedro.lcruz@upm.es (P.L. López-Cruz), pedro.larranaga@fi.upm.es (P. Larrañaga), defelipe@cajal.csic.es (J. DeFelipe), mcbielza@fi.upm.es (C. Bielza).

0888-613X/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2013.03.011

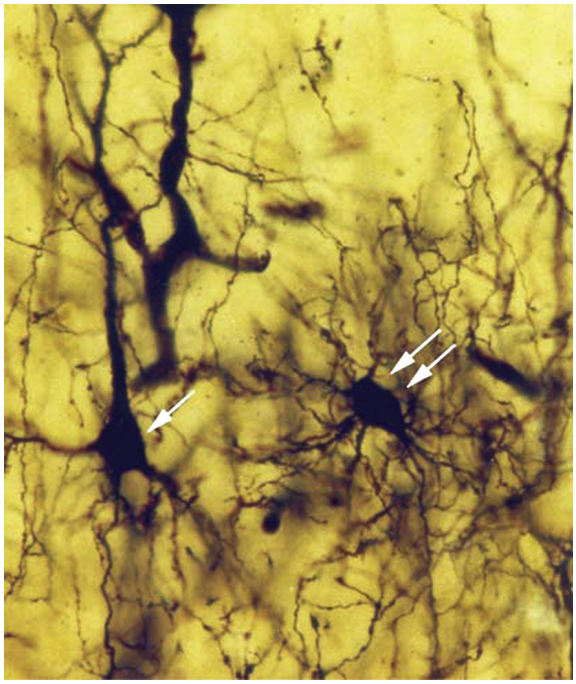


Fig. 1. Photomicrograph from Cajal's preparation of the occipital pole of a cat stained with the Golgi method, showing a pyramidal cell (one arrow) and an interneuron (neurogliaform cell) (two arrows). From DeFelipe and Jones (Cajal on the Cerebral Cortex, Oxford University Press, New York, 1988).

mon nomenclature for naming cortical neurons is a crucial prerequisite for advancing in our knowledge of neuronal structure [3,8].

Bayesian networks [9,10] are a kind of probabilistic graphical model that provides a natural way of modeling uncertainty in artificial intelligence. Therefore, they have been successfully applied across a large number of problems from very different domains [11]. Bayesian networks are specially well suited for modeling and incorporating expert's knowledge, although this kind of analysis has not been applied to its full potential for neuron classification. There are two approaches for integrating this information into a Bayesian network. First, we can elicit both the structure [12] and the parameters [13] of the Bayesian network. Second, we can build a dataset which reflects the behavior of the expert and learn a Bayesian network from the data. This paper focuses on the second approach, i.e., a consensus Bayesian network is built based on data which reflects expert opinions.

We present a methodology for building a Bayesian network that models the opinions of a group of experts. First, a Bayesian network was learned for each expert, representing his/her behavior in the classification task. Second, a clustering algorithm was run on the Bayesian networks to find groups of experts with similar behaviors, and a representative Bayesian network was induced for each cluster of experts. Expert behavior when classifying the set of interneurons was extremely variable. Therefore, experts with similar behaviors have to first be clustered and then combined. Otherwise, combining all experts behaviors into a single consensus model would presumably hide some of these differing behaviors [14,15]. In this way, we can explicitly model each group of similar experts as a representative Bayesian network for the cluster. Third, the final consensus model was a Bayesian multinet [16] encoding a mixture of Bayesian networks [17,18], where each component was the Bayesian network which represented the opinions of a cluster of experts. A similar idea has been proposed for case-based Bayesian networks [19,20], where the authors cluster the observations before learning a Bayesian network which captures the different properties of each cluster. Bayesian multinets are a kind of asymmetric Bayesian network which allows to model different statistical (in)dependences between the variables for different values of a distinguished variable. Bayesian multinets can capture local differences between variables and model the problem domain more closely, allowing for sparser models and more robust parameter estimation. For instance, they have been shown to outperform other Bayesian network models in supervised classification problems [21].

The model was studied at length to validate the proposed methodology and to gather useful knowledge for neuroscience research. The resulting consensus Bayesian multinet can be used to analyze the behavior of a set of experts and to reason about the underlying classification task. The representative Bayesian networks for each cluster can be compared to find similarities and differences between groups of experts and to identify different behaviors or currents of opinion. Also, we can use the consensus model to reason about the task the experts were asked to perform. For instance, we can introduce some evidence into the consensus Bayesian multinet and infer "agreed" answers to those queries. These "agreed" answers could be compared to those obtained by each representative Bayesian networks to find clusters of experts with outlying behaviors against experts with moderate opinions.

We apply the proposed methodology to the problem of the morphological classification of GABAergic interneurons from the cerebral cortex. The research is based on a previous study [22], where we selected and asked a group of 48 experts to classify a set of 320 interneurons according to their most prominent morphological features. However, the methodology presented in this study can be applied to a wide range of scientific fields. For instance, in a medical setting, it may be interesting to model and analyze the different opinions of a group of physicians regarding the diagnosis, prognosis or the most appropriate treatment for a given disease. Another example can be found in a risk assessment scenario, where different people could have different opinions on a given matter depending on their personal preferences, risk perception, etc. The process of obtaining the opinions of different experts on a given task (here, the morphological classification of interneurons) is challenging because it can be difficult, costly and time-consuming. However, new Internet tools and crowdsourcing techniques have alleviated some of these problems, and obtaining classification data from different experts is now affordable for a lot of problems [23].

The paper is organized as follows. Section 2 explains the data acquisition process for gathering the experts' morphological classification of the set of interneurons. Section 3 details the proposed methodology for building a consensus Bayesian multinet which models experts' opinions. Section 4 includes the evaluation of the model and the biological interpretation of the results. Finally, Section 5 ends with conclusions and suggestions for future work.

2. Interneuron classification by a set of experts

We selected $N = 320$ cortical GABAergic interneurons from different species: cat, human, monkey, mouse, rabbit and rat used in a previous study [22]. Briefly, three-dimensional reconstructions of 241 of those interneurons were retrieved from NeuroMorpho.org [7], whereas the rest were scanned from relatively old papers with no data on the three-dimensional distribution of their dendrites and axons. A set of 48 experts were asked to classify each one of the neurons according to their most prominent morphological features. A web application [1] was built to display the neuronal morphologies for the participants and to retrieve their classifications. Two-dimensional projections of all the neurons were available. Additionally, a three-dimensional visualization applet based on Cvapp software [24] was provided for the neurons taken from NeuroMorpho.org, which experts could use to navigate, rotate and zoom the neuronal morphologies. Fig. 2 shows a screenshot of the web application. Additional data about the location of the neuron, such as the cortical area, the layer and the thickness of the layer were included when available. Other web application features included a help page with instructions and definitions of the neuronal types, and a search engine which showed other neurons previously classified by the expert as a given neuronal type. These data were obtained and analyzed in [22]. The goal of this research was to achieve a common nomenclature for the cortical GABAergic interneurons with a utilitarian purpose. The agreement between experts when classifying the interneurons was studied at length. We found that agreement was reasonably high for the attributes describing the general neuronal morphology. Looking at the low-level classification into ten different neuronal types, however, we found remarkable disagreements between the experts for some neuronal types. Here, the goal is to build a consensus Bayesian multinet which models the opinions of all the experts and to use this model to further investigate their agreements and disagreements.

The experts who participated in the experiment were asked to classify the neurons according to four attributes describing the main morphological features of the neurons:

1. The first attribute described the horizontal distribution of the axon relative to the cortical layer. Here, the experts had to separate neurons with an axonal arborization in the same layer as the soma (`Intralaminar`) from neurons with axons distributed in different layers (`Translaminar`).
2. The second attribute referred to the vertical distribution of the axon relative to a reference cortical column (width = 300 μ m). The experts had to classify each neuron according to whether the axonal arborization is distributed primarily in the same cortical column (`Intracolumnar`) or in different cortical columns (`Transcolumnar`).
3. The third attribute represented the relative position of the axon and the dendrites. Neurons with dendritic arbors placed in the center of the axonal arborization were classified as `Centered`, whereas neurons with dendrites shifted with respect to the axon were classified as `Displaced`. When a neuron was classified as both `Translaminar` and `Displaced`, the experts were asked to further characterize the neurons according to whether the axon was directed towards the cortical surface (`Ascending`), the white surface (`Descending`) or both (`Both`).
4. The fourth attribute included a low-level classification of the neurons into nine neuronal types which are frequently used in the literature [25]: `Arcade`, `Cajal-Retzius`, `Chandelier`, `Common basket`, `Horse-tail`, `Large basket`, `Martinotti`, `Neurogliaform` and `Common type`. Additionally, the experts could classify a neuron as `Other` and provide an alternative name for that neuron if they felt that it did not fit any of the proposed neuronal types.

A neuron was classed as `Uncharacterized` when the reconstructed part of the morphology was not clear enough (due to incomplete labeling, reconstruction noise, etc.) for it to be worthwhile having a go at classification. When a neuron was classified as `Uncharacterized`, no value could be given for the other attributes.

[1] Available at http://cajalbbp.cesvima.upm.es/gardenerclassification/. Username: ijar. Password: ijar.

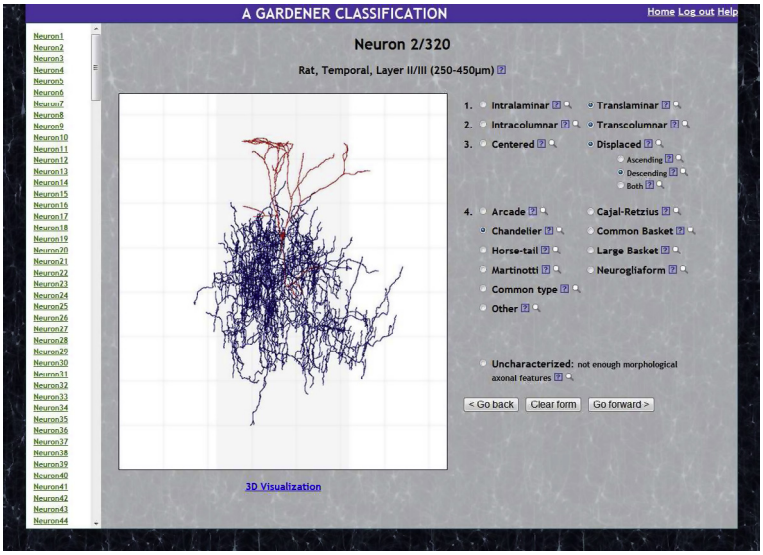


Fig. 2. Web application showing one of the 320 neurons to be classified by each expert.

Each expert was administered the form in Fig. 2 for each neuron. When the experiment finished, 42 out of the 48 experts had classified all 320 neurons. We only used the information about the 42 experts who completed the experiment. The goal in this paper was to build a model which encoded the opinions of the experts when classifying the interneurons in the experiment.

3. A methodology for inducing a consensus Bayesian multinet from a set of expert opinions

In this section, we detail the process for obtaining a Bayesian multinet representing the consensus among the experts who completed the experiment. Fig. 3 visually represents the whole methodology, which can be summarized in three main steps:

1. Learn one Bayesian network for each expert using the classifications provided in the experiment.
2. Cluster the Bayesian networks into groups and induce a new representative Bayesian network for each cluster, which models the opinions of the experts in the cluster.
3. Combine the representative Bayesian networks of each cluster into one consensus Bayesian multinet.

The following sections describe each step in the previous methodology. Section 3.1 introduces Bayesian networks theory and details how to use the classification provided by each expert to learn a Bayesian network representing his/her behavior in the experiment. Section 3.2 explains how to discover groups of similar Bayesian networks by applying clustering algorithms and how to induce a representative Bayesian network for each group. In Section 3.3, the final consensus Bayesian multinet model is built from the representative Bayesian networks of each cluster.

3.1. Bayesian network modeling of each expert's behavior

Bayesian networks [9,10] are a class of probabilistic graphical models, defined as a pair $B = (G(\mathbf{X}, \mathbf{A}), \mathbf{P})$, where:

- $G(\mathbf{X}, \mathbf{A})$ is the graphical component of the model, i.e., a directed acyclic graph (DAG) where the nodes ($\mathbf{X}$) represent the variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ in the problem domain and the arcs ($\mathbf{A}$) encode the probabilistic conditional (in)dependence relationships between the variables.
- $\mathbf{P}$ is the probabilistic component of the model. $\mathbf{P}$ includes a conditional probability table $P(X_i|\mathbf{Pa}(X_i))$ for each variable X_i, $i = 1, \ldots, n$ in the problem, where $\mathbf{Pa}(X_i)$ is the set of parents of X_i in G: $\mathbf{Pa}(X_i) = \{Y \in \mathbf{X} | (Y, X_i) \in \mathbf{A}\}$. Therefore, $\mathbf{P} = \{P(X_i|\mathbf{Pa}(X_i), i = 1, \ldots, n\}$.

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

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

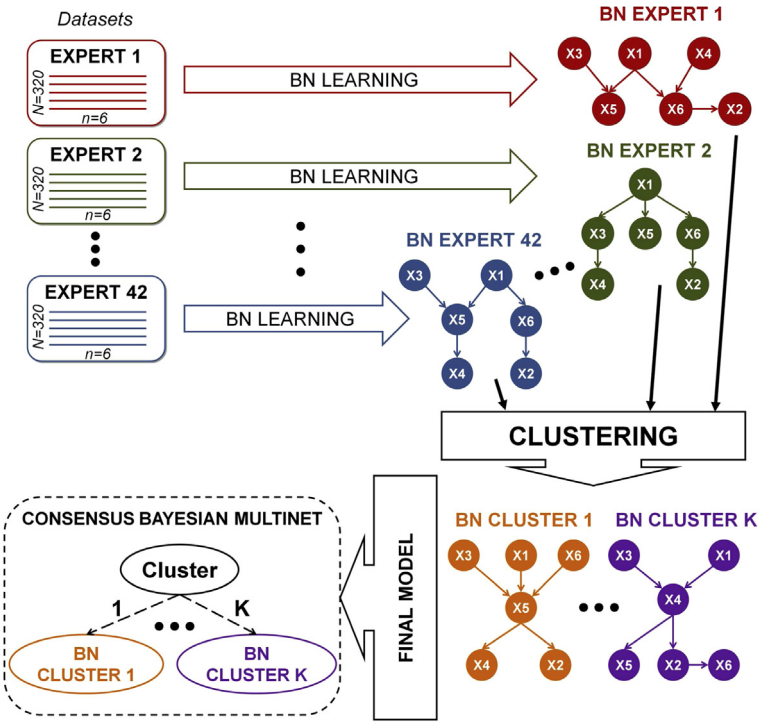


Fig. 3. General methodology for building a consensus Bayesian multinet which represents the behavior of a set of experts.

Bayesian networks are both interpretable and efficient. The graphical component of a Bayesian network is a compact representation of the problem domain, while the factorization of the JPD reduces the computational workload of using high-dimensional probability distributions.

Bayesian network learning from data is a two-step procedure [26–28]: structural search and parameter fitting. There are two main methods for learning the structure G of a Bayesian network: constraint-based methods and score+search methods. Constraint-based methods rely on performing statistical tests to find conditional independence relationships between groups of variables in the network. Then, an undirected independence graph is built, and edge orientation discovers a Bayesian network structure which encodes those conditional independence relationships. Score+search approaches use a heuristic search algorithm to explore the space of DAGs, and a score function to evaluate the candidate network structures and direct the search procedure. Once the network structure has been found, the parameters in the conditional probability tables ($\mathbf{P}$) are estimated from the counts in the dataset.

We focused on score+search methods and learned the Bayesian network structure using the greedy thick thinning (GTT) algorithm [29] implemented in the GeNIe free modeling environment.[2] K2 scoring function [30] was used to evaluate each candidate structure, by measuring the joint probability of the Bayesian network structure G and a dataset $\mathcal{D}$:

$$P(G,\mathcal{D}) = P(G)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}!, \tag{2}$$

where $P(G)$ is the prior probability of the network structure G, r_i is the number of distinct values of X_i, q_i is the number of possible configurations of $\mathbf{Pa}(X_i)$, N_{ij} is the number of instances in the dataset $\mathcal{D}$ where the set of parents $\mathbf{Pa}(X_i)$ takes their j-th configuration, and N_{ijk} is the number of instances where the variable X_i takes the k-th value x_{ik} and $\mathbf{Pa}(X_i)$ takes their j-th configuration $\left(N_{ij}=\sum_{k=1}^{r_i}N_{ijk}\right)$.

The GTT algorithm implements a two-step procedure for discovering a Bayesian network structure (see Algorithm 1). Given an initial (empty) graph G, it iteratively adds the arc which maximizes the increase in the likelihood (thicking step). When no further increment is possible by adding arcs, the algorithm iteratively removes arcs until no arc deletion yields a positive increase in the likelihood (thinning step). Then, the algorithm stops and the resulting Bayesian network structure is returned. The GTT algorithm has a number of advantages, e.g., unlike other methods [30–32] it does not require an ordering of the variables. Also, it is simple, computationally efficient and avoids overfitting by removing arcs in the thinning step.

[2] Developed by the Decision Systems Laboratory of the University of Pittsburgh: http://dsl.sis.pitt.edu.

Algorithm 1 (*Greedy thick thinning algorithm*).
Given an initial graph $G(\mathbf{X}, \mathbf{A})$ and a dataset $\mathcal{D}$

1. Thicking step: While the K2 score function (2) increases:
   (a) Find the arc (X_i, X_j) which maximizes (2) when included in $G'(\mathbf{X}, \mathbf{A}')$ with $\mathbf{A}' = \mathbf{A} \cup \{(X_i, X_j)\}$.
   (b) Set $G \leftarrow G'$.
2. Thinning step: While the K2 score function (2) increases:
   (a) Find the arc (X_i, X_j) which maximizes (2) when deleted in $G'(\mathbf{X}, \mathbf{A}')$ with $\mathbf{A}' = \mathbf{A} \setminus \{(X_i, X_j)\}$.
   (b) Set $G \leftarrow G'$.
3. Return G.

A Bayesian network was learned for each one of the $N_e = 42$ experts who completed the experiment. The goal was to build a model which captures how each expert understands the values of the morphological attributes and their relationships. The graphical representation of the Bayesian networks structures offers a compact and easy way for the experts in the domain to interpret their models. The Bayesian networks were learned independently for each expert, so they do not capture whether or not the experts classified the same neurons in the same way. However, since the experts classified the same set of interneurons, we can use the Bayesian networks to systematically analyze their opinions and behaviors. One would expect that if two experts differed in their opinions (as encoded in their Bayesian networks), then they would also classify the neurons differently. Also, having an individual Bayesian network for each expert makes it easier to analyze and validate the representative Bayesian networks for each cluster and the final consensus Bayesian multinet, because the inputs (Bayesian networks) and the output (Bayesian multinet) share the same representation.

Therefore, one dataset for each expert was generated with the classifications provided in the experiment. The resulting dataset had $N = 320$ observations (the number of interneurons in the experiment) and $n = 6$ variables, which corresponded to the features that the experts were asked to classify. Some restrictions on different combinations of feature values were imposed in the experiment design (see Section 2). For instance, selecting `Uncharacterized` in the first feature disabled all the other variables. Therefore, when a neuron was classified as `Uncharacterized`, the values for the other variables were empty. Similarly, the `Ascending/Descending/Both` feature was only available when `Translaminar` and `Displaced` were selected for the corresponding features. To build the dataset for each expert, we filled in incomplete observations with a new category named `Dummy`. Therefore, for each expert, we had a dataset with $n = 6$ categorical variables with values:

- X_1 ($r_1 = 2$): `Characterized`, `Uncharacterized`.
- X_2 ($r_2 = 3$): `Intralaminar`, `Translaminar`, `Dummy`.
- X_3 ($r_3 = 3$): `Intracolumnar`, `Transcolumnar`, `Dummy`.
- X_4 ($r_4 = 3$): `Centered`, `Displaced`, `Dummy`.
- X_5 ($r_5 = 4$): `Ascending`, `Descending`, `Both`, `Dummy`.
- X_6 ($r_6 = 11$): `Common type`, `Horse-tail`, `Chandelier`, `Martinotti`, `Common basket`, `Arcade`, `Large basket`, `Cajal-Retzius`, `Neurogliaform`, `Other`, `Dummy`.

We used the data provided by each expert in the experiment to learn a Bayesian network which encoded the conditional independence relationships between the variables for that expert. The GTT algorithm was used to find the Bayesian network structure, and the parameters were fitted using maximum likelihood estimators with Laplace correction. We did not allow any variable to be a parent of variable X_1, corresponding to the `Characterized/Uncharacterized` feature. This restriction encoded the knowledge that the decision of classifying a neuron as `Characterized` or `Uncharacterized` should be taken before classifying all the other features (modeled with variables X_2 to X_6). We limited the complexity of the Bayesian networks by imposing a maximum of three parents for each variable. This allowed us to control the size of the conditional probability distributions and to compute robust estimators of their parameters. However, this was not a very restrictive constraint since only 5 out of $6 \times 42 = 252$ variables in all the Bayesian networks had three parents.

In [22], a remarkable variability among experts' opinions when classifying the interneurons was found. We performed a preliminary analysis of the Bayesian networks induced for each expert to check whether or not this variability was reflected in the network structures. Fig. 4 shows how many of the 42 Bayesian networks contained each possible edge between every pair of variables. Disagreements were highlighted in dark grey, showing relationships that appeared in half of the Bayesian networks but omitted in the other half. The Bayesian network structures showed an important variability, e.g., arcs containing relationships between $X_1 - X_2$ or $X_1 - X_3$ were found in approximately half of the Bayesian networks but were absent in the rest. Additionally, some relationships appeared in almost all the Bayesian networks ($X_4 - X_5$ and $X_3 - X_6$), whereas other relationships were not found in any network structure ($X_3 - X_4$ and $X_3 - X_5$). We can conclude that the disagreements between experts were also reflected in their induced Bayesian network structures. These disagreements between the experts prevented us from building a single Bayesian network which represented them all, because a single common structure could obscure the differences in the experts' behavior [14,15]. Therefore, we first performed a clustering step to find groups of Bayesian networks encoding similar expert opinions and built the final consensus Bayesian multinet reflecting all the groups of experts.

	X_1	X_2	X_3	X_4	X_5	X_6
X_1		17	17	14	2	10
X_2			4	32	30	30
X_3				0	0	38
X_4					41	14
X_5						29
X_6						

Fig. 4. Number of Bayesian networks including the edge for each pair of variables. The matrix is symmetric so only the upper triangle is shown. Light-shaded cells show agreements in the experts' Bayesian network structures, i.e., edges which appear in most or none of the Bayesian networks, whereas dark-shaded cells show disagreements in the Bayesian network structures.

3.2. Clustering of Bayesian networks

The experiment was designed to find groups of Bayesian networks corresponding to experts with similar behaviors. In this section, we detail the process of finding groups of Bayesian networks which define similar JPDs and inducing a representative Bayesian network for each cluster. To the best of our knowledge, the problem of clustering Bayesian networks had not been studied before. Note that this is not the same problem as using Bayesian networks to cluster data [33,34] or clustering variables in Bayesian network learning for high-dimensional problems [35,36]. Bayesian networks have two main components (see Section 3.1): the graphical part and the probabilistic part. Therefore, we could consider clustering at the two levels:

- *Clustering of Bayesian network structures*: The graphical component $G(\mathbf{X}, \mathbf{A})$ of a Bayesian network is a DAG which encodes the conditional (in)dependence relationships between the variables in the problem domain. Therefore, we could use existing approaches for clustering graphs [37,38] and, in particular, clustering DAGs [39] to find groups of structurally similar Bayesian networks. Another approach could be to list the conditional independence relationships encoded in a Bayesian network and then apply a clustering algorithm to group Bayesian networks which share the same set of conditional independences.
- *Clustering of Bayesian network probabilities*: The probabilistic component $\mathbf{P}$ in a Bayesian network contains the conditional probability distributions of each variable X_i given its parents $\mathbf{Pa}(X_i)$. Clustering of probability distributions has not received much attention in the statistics and machine learning fields. The approaches in [40,41] cannot be directly applied to our problem because $\mathbf{P}$ includes several (conditional) probability distributions: one probability distribution for each variable given its parents' values. Comparing the conditional probability distributions of the same variable in two different Bayesian networks is challenging because each variable can have a different number of parents, and the set of parents may be different. Therefore, the conditional probability distributions cannot be directly compared. A simple approach, which could also be useful in problem domains with a lot of variables, would be to compute the marginal probability distribution for each variable in each Bayesian network and to cluster the Bayesian networks based on these marginal distributions.

Here, we propose clustering the Bayesian networks based on the joint probability distributions that they encode. Therefore, our approach is included in the second group of techniques. Fig. 5 outlines the proposed methodology, which can be summarized in three steps. First, the JPD encoded by each Bayesian network is computed. These JPDs also model the experts' behavior in the experiment. Second, groups of similar experts/Bayesian networks are found by clustering their corresponding JPDs. Third, a representative Bayesian network is induced for each cluster, which represents the common behavior of the experts in the cluster. The following sections detail each one of these three steps.

3.2.1. Computation and preprocessing of the joint probability distributions

For each expert, we computed the JPD over the six variables encoded by the Bayesian network learned in the previous step. Not all the experts selected all the possible values when completing the experiment, e.g., some experts did not classify any neuron as `Arcade`, `Cajal-Retzius` or `Other` in variable X_6. Therefore, not all the Bayesian networks contained all the values for all the variables. However, we wanted all the JPDs to have the same number of values for the purposes of comparison. Therefore, we completed the conditional probability tables in the Bayesian networks learned with GeNIe using maximum likelihood estimators with Laplace correction, so that all the Bayesian networks had all the values for all the variables. Then, the JPD over all the variables encoded by each Bayesian network was computed by multiplying the conditional probability distributions in $\mathbf{P}$, as in Eq. (1). The resulting JPD had $2 \times 3 \times 3 \times 3 \times 4 \times 11 = 2376$ values. However, most of these values corresponded to inadmissible combinations of the values of the variables. For example, when `Uncharacterized` was selected, all the other variables should have the value `Dummy` and any other combination of values was not valid. Similarly, variable X_5 could only take a value different from `Dummy` when X_2 = `Translaminar` and X_4 = `Displaced`. We erased the values in the JPDs corresponding to these forbidden combinations. The resulting JPDs had 121 values each.

3.2.2. Clustering of joint probability distributions

We approached the problem of finding groups of similar Bayesian networks by clustering the JPDs obtained in the previous step (Section 3.2.1). We generated a dataset with $N_e = 42$ observations and $r = 121$ variables, where each observation (row)

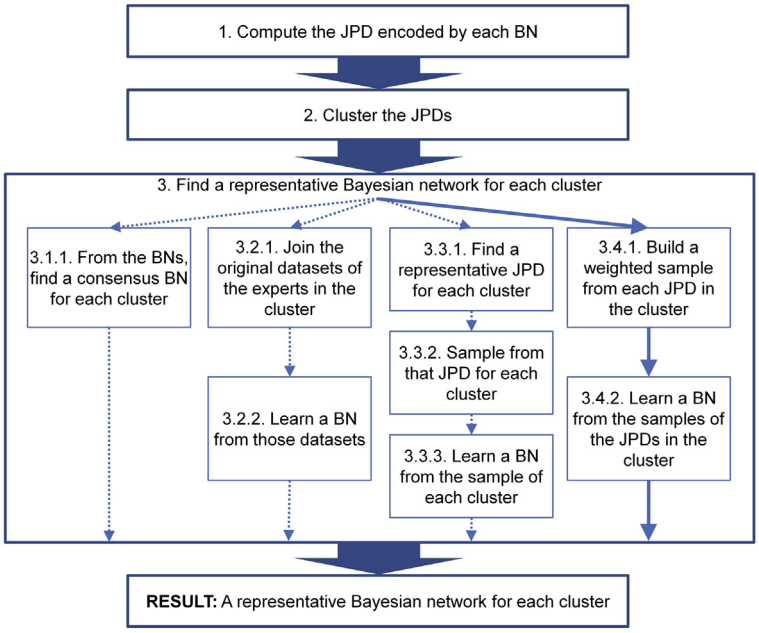


Fig. 5. Procedure for clustering Bayesian networks. In step 3, the solid line represents the proposed workflow for inducing a representative Bayesian network for each cluster, whereas the dashed lines show alternative ways of achieving this goal.

was a JPD corresponding to the Bayesian network of each expert and each variable (column) was a value of the JPD. There are three main paradigms which can be used for clustering [42]: probabilistic, hierarchical and partitional clustering.

Hierarchical and partitional paradigms are the classical approaches to clustering. In general, both paradigms rely on the definition of a distance or dissimilarity measure between the observations. A classical agglomerative (bottom-up) hierarchical clustering algorithm starts with one cluster per observation and iteratively merges the two most similar clusters according to some criterion, called linkage function, which depends on the distances of the observations in the clusters. Therefore, hierarchical clustering techniques do not generate a single partition but a hierarchy of clusters. On the contrary, partitional clustering techniques generate a single partition of the objects into clusters by applying an optimization process which maximizes/minimizes an objective function. This objective function usually measures the distances between the objects in the same cluster (minimization) and/or the distance between objects in different clusters (maximization). In both hierarchical and partitional approaches, the number of clusters to be generated is a free parameter that has to be set by the expert. Also, an appropriate distance measure has to be chosen depending on the nature of the data.

Probabilistic clustering deals with the problem of fitting a finite mixture of distributions [43], where each component is the probability distribution which models the observations belonging to the cluster. Probabilistic clustering offers a number of advantages. First, it generates a probabilistic model which describes the data. Using that model, one can compute the (posterior) probability of a given observation belonging to each cluster. Also, it is able to formally address the problem of model selection (finding an appropriate number of clusters). Since each of our observations is a JPD, the Dirichlet distribution [44] could be a suitable choice of a probability density function for each component. However, the low number of observations ($N_e = 42$) over the number of variables ($r = 121$) ruled out the use of this approach, because it is difficult to obtain accurate estimators of a finite mixture model with so few data.

Here, we chose to adapt the classical K-means algorithm [45] to characterize properties of our data. Algorithm 2 shows a general outline of the algorithm. The algorithm alternates two steps. First, the observations are assigned to the cluster with the closest center. Second, the cluster centers are recomputed taking into account only the observations in the clusters.

Algorithm 2 (*K-means algorithm*)**.**
Input: the number of clusters K and a dataset of r-dimensional observations $\mathcal{P} = \{\mathbf{o}_i, \ldots, \mathbf{o}_{N_e}\}$. Steps:

1. Initialize the cluster centers $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ to K random observations in $\mathcal{P}$ without replacement.
2. While the cluster centers $\mathbf{C}$ change
   (a) For each observation $\mathbf{o}_i$, compute the dissimilarity between $\mathbf{o}_i$ and each cluster center $\mathbf{c}_k$: $d(\mathbf{o}_i, \mathbf{c}_k)$.
   (b) Assign each observation $\mathbf{o}_i$ to cluster $k_i^* \in \{1, \ldots, K\}$ with the closest center: $k_i^* \leftarrow \arg\min_{k=1,\ldots,K} d(\mathbf{o}_i, \mathbf{c}_k)$.
   (c) Recompute the cluster centers from the observations in each cluster: $\mathbf{c}_k \leftarrow \texttt{Combine}(\{\mathbf{o}_i \in \mathcal{P} | k_i^* = k\})$.
3. Return the cluster centers $\mathbf{C}$.

The K-means algorithm iteratively minimizes the sum of the distances of each observation to its cluster center: $J(\mathcal{P}, \mathbf{C}) = \sum_{i=1}^{N_e} d\left(\mathbf{o}_i, \mathbf{c}_{k_i^*}\right)$. K-means is guaranteed to find a local minimum of $J(\mathcal{P}, \mathbf{C})$. Therefore, Algorithm 2 is usually restarted several times with different initialization values for step 1. A similar approach was used in [14,15] in the context of decision making in influence diagrams. In order to apply the K-means algorithm to the problem of clustering JPDs we have to choose a suitable dissimilarity measure $d(\mathbf{o}_i, \mathbf{c}_k)$ and a method for computing the cluster centers from the observations in the cluster (`Combine` function in step 2(c) of Algorithm 2).

Dissimilarity measures for probability distributions. In general, our choice of a dissimilarity measure $d(\mathbf{o}_i, \mathbf{c}_k)$ should be, at least, symmetric. Therefore, one could consider using the symmetric Kullback–Leibler divergence,

$$d_{KL}(\mathbf{p}_1, \mathbf{p}_2) = KL(\mathbf{p}_1||\mathbf{p}_2) + KL(\mathbf{p}_2||\mathbf{p}_1),$$

where $KL(\mathbf{p}_1||\mathbf{p}_2)$ is the Kullback–Leibler divergence [46] from an empirical probability distribution $\mathbf{p}_1$ to the true distribution $\mathbf{p}_2$

$$KL(\mathbf{p}_1||\mathbf{p}_2) = \sum_{j=1}^{r} p_{1j} \log \frac{p_{1j}}{p_{2j}},$$

where r is the number of values of the probability distribution $\mathbf{p}_i$, and p_{ij} is the probability of the jth value in the probability distribution $\mathbf{p}_i$. One disadvantage of the Kullback–Leibler divergence is that it is not upper bounded. However, other measures can be considered, such as the Jensen–Shanon divergence,

$$d_{JS}(\mathbf{p}_1, \mathbf{p}_2) = \frac{1}{2} KL(\mathbf{p}_1||\mathbf{m}) + \frac{1}{2} KL(\mathbf{p}_2||\mathbf{m}), \tag{3}$$

where $\mathbf{m}$ is the mean probability distribution $\mathbf{m} = 0.5(\mathbf{p}_1 + \mathbf{p}_2)$. The Jensen–Shanon divergence has a number of interesting properties [47]: it is symmetric, its square root is a metric and it is bounded $0 \leq d_{JS} \leq 1$. Therefore, we chose d_{JS} as the dissimilarity measure for the K-means algorithm. Additionally, the fact that d_{JS} is a bounded measure was also useful when computing the representative Bayesian network for each cluster (Section 3.2.3).

Combination of probability distributions. Two main methods can be found in the literature to compute an average probability distribution $\overline{\mathbf{p}}$ from a set of probability distributions [48]: the linear combination pool (LinOp) and the logarithmic combination pool (LogOp). If we have N_k probability distributions $\{\mathbf{p}_1, \ldots, \mathbf{p}_{N_k}\}$ in a cluster, the linear combination pool is defined as the weighted arithmetic mean

$$\overline{\mathbf{p}}_{LinOp} = \sum_{i=1}^{N_k} \omega_i \mathbf{p}_i, \tag{4}$$

where $\sum_{i=1}^{N_k} \omega_i = 1$ and $\omega_i > 0$ is the weight for the probability distribution $\mathbf{p}_i$. The logarithmic combination pool is defined as the weighted geometric mean

$$\overline{p}_{jLogOp} = \frac{\prod_{i=1}^{N_k} p_{ij}^{\omega_i}}{\sum_{v=1}^{r} \prod_{i=1}^{N_k} p_{iv}^{\omega_i}}. \tag{5}$$

Genest and Zideck [48] give a number of reasons for choosing LogOp over LinOp, the most compelling being that it is externally Bayesian, i.e., it can be derived from joint probabilities [49]. Also, it is known that LinOp does not preserve independences [50], i.e., combining probability distributions which share a common independence does not guarantee that the resulting distribution will be equally independent. Heskes [51] showed that using LogOp is equivalent to finding the probability distribution $\mathbf{p}$ which minimizes the weighted sum of the Kullback–Leibler divergences to each probability distribution $\mathbf{p}_i$

$$\overline{\mathbf{p}}_{LogOp} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N_k} \omega_i KL(\mathbf{p}||\mathbf{p}_i).$$

Therefore, we chose LogOp as a combination method for computing the cluster centers in the K-means algorithm (step 2(c) of Algorithm 2). All the experts were considered as equals, so the weights ω_i were all set to $1/N_k$ for each cluster.

3.2.3. Finding a representative Bayesian network for each cluster

Once the JPDs have been clustered and K cluster centers (JPDs) have been obtained, the next step is to induce a Bayesian network which represents the common features of the corresponding Bayesian networks (and experts) in the cluster. Step 3 in Fig. 5 shows four possible approaches for finding a representative Bayesian network for each cluster. In the following, we discuss the four approaches for performing this task, we review the works related to each one and analyze their advantages and disadvantages for modeling experts' opinions on the problem of the morphological classification of GABAergic interneurons.

The first approach consists of directly combining a set of Bayesian networks into a single representative one (Fig. 5, step 3.1.1). Learning Bayesian networks from a set of expert opinions has been a recurrent interest in the field. However, [52] showed that even when the Bayesian networks share the same structure, there is no way of combining the parameters

to preserve that structure. They proposed a methodology for combining both the Bayesian network structures and the parameters. The algorithm finds a common network structure by transforming the DAGs into moral graphs, performing the union of the edges and transforming the resulting moral graph back into a DAG. The conditional probability tables are combined by applying the LogOp combination pool of Eq. (5). This approach is expected to yield highly connected Bayesian networks because of the union of the edges of the moral graphs. Therefore, the conditional probability distributions will have a lot of parameters and their estimates will not very robust when there are few training instances (in our scenario, 320 neurons). Sagrado and Moral [53] studied the theoretical properties of Bayesian networks obtained by performing either the intersection or the union of the arcs of the network structures, and proposed ways for finding the consensus Bayesian network structure. However, they left the combination of the conditional probability tables as a matter for future research. Zhang et al. [54] built on the work by Sagrado and Moral [53] and proposed a score+search method for fusing the Bayesian network structures. However, they applied Bayesian inference not data to combine the parameters of the Bayesian networks and to compute the scores of the network structures. Peña [55] derived a correction of the algorithms proposed by Matzkevich and Abramson [56,57] for finding the consensus Bayesian network structure with a minimum number of parameters. It represents only the common independences appearing in all the Bayesian network structures. He outlined some ideas for combining the parameters of the Bayesian networks, but this issue was mainly left for future research. Finally, other methods for Bayesian network aggregation have been proposed in the context of model averaging (for a review, see Section 4.13 in [28]). These methods combine the probabilities inferred with a set of Bayesian networks but they do not obtain a single representative Bayesian network which models the opinions of a set of experts. In the neuron classification problem, obtaining the representative Bayesian network explicitly was important because the experts would like to analyze and interpret these models and not only their outputs.

The second approach deals with the problem of learning a consensus Bayesian network from data. Maynard-Reich and Chajewska [58] assumed that the differences between experts are the result of observing different subsets of data. This is related to the problem of learning Bayesian networks from distributed datasets, see e.g. [59]. In our experiment, however, all the experts classified the same 320 interneurons, so this assumption did not apply. Steps 3.2.1 and 3.2.2 show another possibility which conformed to our problem: joining the original datasets for each expert in the cluster and learning a Bayesian network from this cluster's dataset. We could consider different degrees of membership of each expert to his cluster by only including a subset of interneurons from his dataset in the cluster's dataset. However, there were some neuronal morphologies which did not appear frequently in the data. Therefore, this approach could erase some important information about the experts.

The third approach is based on sampling the JPDs and learning a Bayesian network from the generated data as explained in Section 3.1 (Fig. 5, steps 3.3.1 to 3.3.3). First, we compute a representative JPD for each cluster, then we sample the JPD to obtain a dataset and, finally, we learn a Bayesian network from that dataset. Again, one could consider using the LinOp (Eq. (4)) or the LogOp (Eq. (5)) combination pools for computing the representative JPD and different weights could be applied to each expert's JPD. However, if the cluster center JPD does not accurately represent all the experts in the cluster, the resulting representative Bayesian network for the cluster would not model all the experts' opinions either.

Here, we implemented another approach based on proportional sampling of the individual JPDs of each expert (Fig. 5, steps 3.4.1 and 3.4.2). The goal was to obtain a sample of data for each cluster k, taking into account the dissimilarity between each JPD and the cluster center $\mathbf{c}_k$ to decide the number of samples to draw from each JPD. The fact that $d_{JS}(\mathbf{p}_i, \mathbf{c}_k)$ (Eq. (3)) is upper bounded facilitates the computation of these expert degrees of membership. For a given cluster k, we found the JPDs included in the cluster and computed a degree of membership μ_i for each one as

$$\mu_i = \frac{1 - d_{JS}(\mathbf{p}_i, \mathbf{c}_k)}{\sum_{j=1}^{N_k} \left(1 - d_{JS}(\mathbf{p}_j, \mathbf{c}_k)\right)}.$$

Then, to obtain a sample with size M for cluster k, $\mu_i \times M$ observations were drawn from each JPD $\mathbf{p}_i$ in cluster k. Finally, both the structure and the parameters of the representative Bayesian network were learned (Section 3.1) from that sample of size M obtained for each cluster.

This approach tries to avoid some of the disadvantages of the other three approaches. The learning algorithm allows to fully specify the Bayesian networks as opposed to the methods in the first approach (step 3.1.1), which can have problems when computing the parameters of the conditional probability distributions. An advantage of this method over the second approach (steps 3.2.1 and 3.2.2) is that our approach uses the Bayesian networks themselves (through their JPDs) to compute the representative Bayesian network for the cluster. The second approach, on the other hand, assumes that the Bayesian networks were learned from data and that experts' data is still available. This may not be the case in some scenarios where Bayesian networks are elicited from experts' knowledge and not induced from data. Finally, as opposed to the third approach, we consider each Bayesian network in the cluster individually through its JPD while taking into account different degrees of membership to the cluster.

3.3. Building the consensus Bayesian network

The final step in the methodology (see Fig. 3) deals with the problem of building a probabilistic graphical model that represents all the experts who participated in the experiment and also takes into account their differing behaviors. We

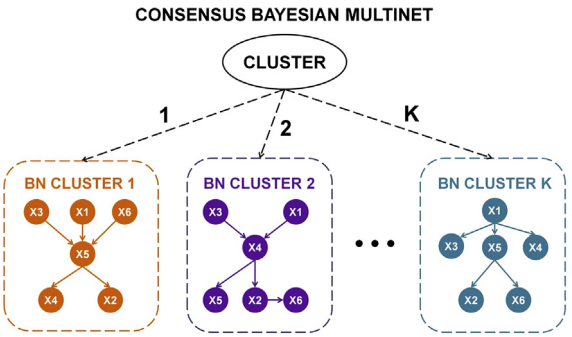


Fig. 6. Finite mixture of Bayesian networks represented as a Bayesian multinet with the cluster variable as the distinguished variable.

modeled the whole problem as a finite mixture of Bayesian networks [17]

$$P(\mathbf{X} = \mathbf{x}) = \sum_{k=1}^{K} \pi_k P(\mathbf{X} = \mathbf{x} | C = k, G_k, \mathbf{P}_k), \tag{6}$$

where π_k was set to the proportion of experts in the kth cluster (N_k/N_e), and each component $P(\mathbf{X} = \mathbf{x}|C = k, G_k, \mathbf{P}_k)$ was the representative Bayesian network for the kth cluster with structural component G_k and probabilistic component $\mathbf{P}_k$. Finite mixtures of Bayesian networks form a kind of Bayesian multinet [16] with a distinguished variable C which represents the cluster variable. In principle, the cluster variable C is hidden but we found it previously by clustering the Bayesian networks (Section 3.2). Fig. 6 is a diagram of the final consensus Bayesian multinet.

4. Results

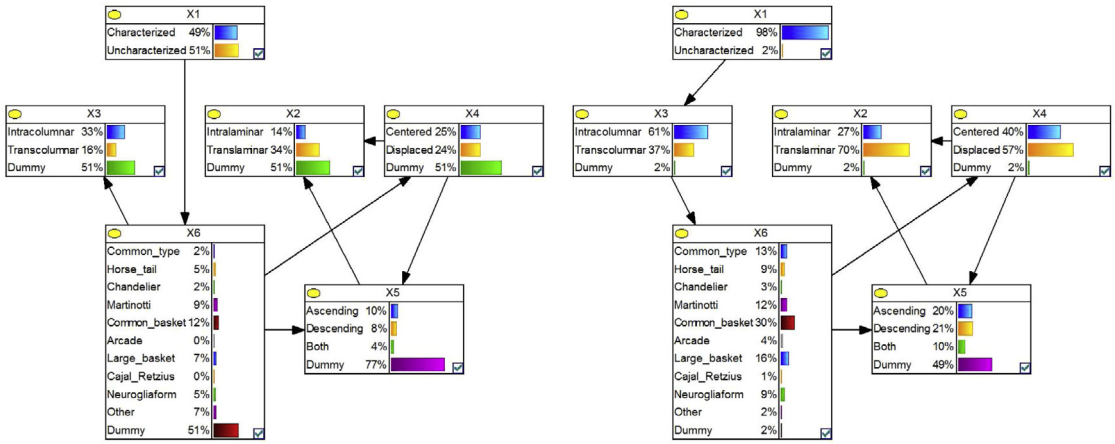
This section includes the results corresponding to one run of the whole process as described in Section 3 (see Fig. 3). First, one Bayesian network was learned for each one of the 42 experts who completed the experiment (Section 3.1). Then we clustered the Bayesian networks following the procedure described in Section 3.2. We started the process by computing the JPD encoded by each Bayesian network and generating a data matrix with dimensions 42×121, where each row was a JPD corresponding to an expert and each column corresponded to a value of the JPD, i.e., a combination of possible values of the variables in the experiment. We used the K-means algorithm with Jensen–Shanon distance (Eq. (3)) and the LogOp combination pool (Eq. (5)) to cluster the JPDs. We used $K = 6$ clusters because we were thus able to find distinguishable clusters with characterizing properties. We used proportional sampling to get a dataset for each cluster, and a representative Bayesian network was learned from that sample using GeNIe. Finally, a consensus probabilistic graphical model was built as a finite mixture of Bayesian networks represented with a Bayesian multinet (Section 3.3). In the consensus Bayesian multinet, the cluster variable was the distinguished variable and each component of the mixture was the representative Bayesian network for a cluster (see Fig. 6).

In the following sections, we analyze the results by studying the consensus Bayesian multinet at different levels. Fig. 7 shows the representative Bayesian networks learned for each cluster of experts. These Bayesian networks can be downloaded in GeNIe format from the supplementary material website.[3] First, the Bayesian networks for each expert learned with the GTT algorithm were compared with other algorithms for learning Bayesian network structures from data (Section 4.1). Then, we tried to characterize each one of the clusters by studying the marginal probabilities of their representative Bayesian networks (Section 4.2). Also, a structural analysis of the Bayesian networks was performed to validate the results and to find agreements and differences between clusters (Section 4.3). We extracted agreed definitions of the different neuronal types proposed in the experiment by performing inferences in both the consensus Bayesian multinet and the representative Bayesian networks for each cluster (Section 4.4). A principal component analysis was performed to visually inspect a low-dimensional representation of the clusters (Section 4.5). Finally, we looked for possible currents of opinion by studying correlations between the clusters and the geographical location of the experts' workplace (Section 4.6).

4.1. Validation of the Bayesian network structure learning algorithm

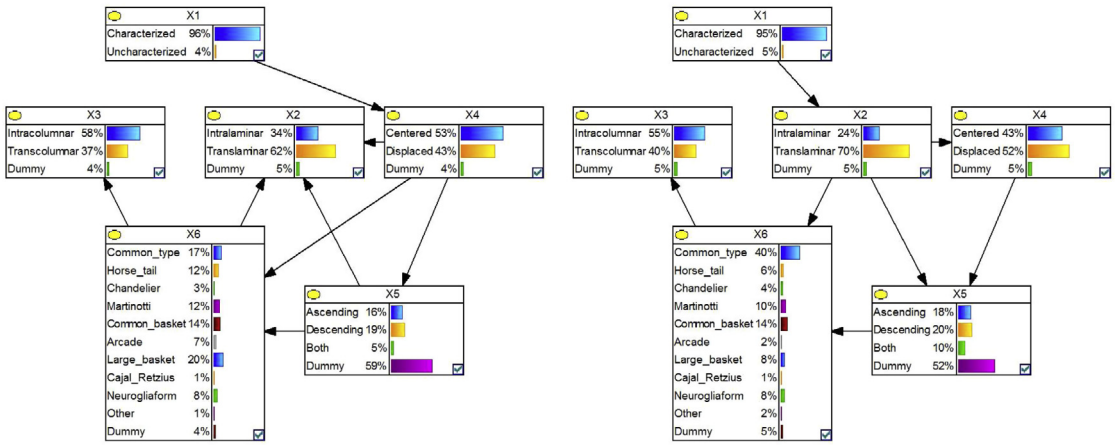
We studied the influence of the structure learning algorithm when finding the Bayesian networks for each expert (see Section 3.1). We compared the Bayesian networks learned with the greedy thick thinning algorithm (Algorithm 1) with other four algorithms for learning Bayesian network structures available in the `bnlearn` package [60] for R statistical software [61]: a hill-climbing algorithm (HC), a tabu search algorithm (TA), a max-min algorithm (MM) and the 2-phase restricted search max-min algorithm (RS). HC and TA are score+search algorithms, whereas MM and RS are hybrid algorithms combining score+search with constraint-based approaches. 100 restarts were computed for the hill-climbing algorithm and the best

[3] Available at http://cig.fi.upm.es/index.php/members/138-supplementary-material.



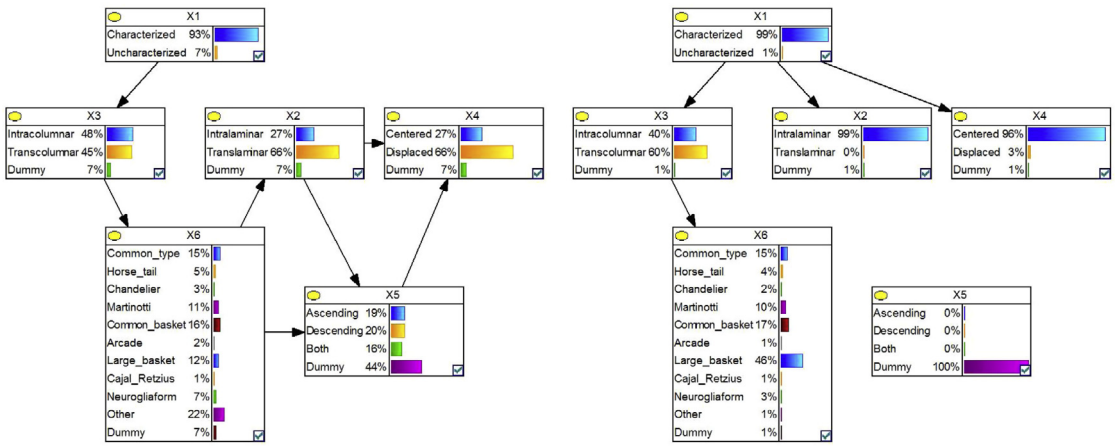
(a) Bayesian network for cluster 1 (3 experts)

(b) Bayesian network for cluster 2 (15 experts)



(c) Bayesian network for cluster 3 (4 experts)

(d) Bayesian network for cluster 4 (12 experts)



(e) Bayesian network for cluster 5 (7 experts)

(f) Bayesian network for cluster 6 (1 expert)

Fig. 7. Network structures and marginal probabilities of the representative Bayesian networks for each cluster. Each one of the Bayesian networks corresponds to a component in the finite mixture of Bayesian networks that builds up the consensus Bayesian multinet.

scoring network structure was returned. Additionally, we considered two scoring functions: K2 [30] and BIC [62]. Thus, for each expert, we learned eight Bayesian network structures using the four algorithms and the two scoring functions. Maximum likelihood estimators of the parameters with Laplace correction were computed for filling in the conditional probability tables. The JPD encoded by each Bayesian network was computed and simplified to a 121-dimensional JPD as

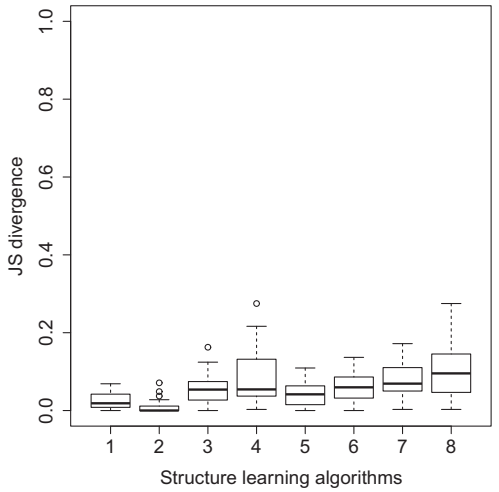


Fig. 8. Comparison between the greedy thick thinning (GTT) algorithm and eight algorithms for learning the Bayesian network structures: (1) HC-K2, (2) TA-K2, (3) MM-K2, (4) RS-K2, (5) HC-BIC, (6) TA-BIC, (7) MM-BIC and (8) RS-BIC. Each boxplot summarizes the 41 Jensen–Shanon divergence values (42 experts minus expert #33) between the JPDs of the Bayesian networks obtained with the GTT algorithm and the JPDs obtained with each one of the eight alternative methods.

explained in Section 3.2.1. The Jensen–Shanon divergence (Eq. (3)) between the JPD corresponding to the Bayesian network learned with the GTT algorithm and the eight alternative structure learning methods was computed. The structure learning algorithms could not be applied for expert #33 because he/she classified all the neurons as $X_2 =$ `Intralaminar` and the algorithms could not handle variables with only one value.

Fig. 8 shows boxplots of the Jensen–Shanon divergence values (Y axis) between the GTT algorithm and the other eight algorithms (X axis) obtained for the 41 experts (42 minus expert #33). Note that the JS divergence is both lower and upper bounded: $0 \leq d_{JS} \leq 1$. We can see that the JS divergence yielded very low values, being almost all of them below 0.2. On the one hand, the TA-K2 algorithm (second boxplot in Fig. 8) yielded the lowest JS divergence values. On the other hand, the RS algorithm (fourth and eighth boxplots in Fig. 8) learned Bayesian networks which yielded JPDs differing the most compared to those obtained with the GTT algorithm. As expected, we can see that algorithms using K2 scoring function yielded lower JS divergences than those using BIC, because the GTT algorithm also used the K2 scoring function. We concluded that the algorithm used for learning the Bayesian network structures did not have an important influence in the proposed methodology because we used the JPDs for clustering the Bayesian networks and they were similar regardless of the applied algorithm.

4.2. Cluster labeling and analysis of the probability distributions

We identified differences between the groups of experts by studying the marginal (or prior) probabilities in the representative Bayesian networks for each cluster (see Fig. 7). We used these marginal probabilities to characterize each group of experts and we interpreted these differences as different approaches when classifying the neurons:

- Cluster 1 (including three experts) represented experts who considered that half of the neurons in the experiment did not have enough reconstructed axonal processes for it to be feasible to actually try to classify them. Thus, they assigned the neurons to the `Uncharacterized` category in X_1 (probability 0.51). The probability of `Uncharacterized` was much lower in all the other Bayesian networks ($\leq$0.07). In fact, the combination of values of the variables with higher probability (mode) corresponded to X_1 = `Uncharacterized`, X_2 = `Dummy`, X_3 = `Dummy`, X_4 = `Dummy`, X_5 = `Dummy` and X_6 = `Dummy`.
- Cluster 2 included 15 experts with a coarse classification scheme. In this Bayesian network, most of the neurons were classified as `Common basket` (0.30). The mode of the JPD encoded in the representative Bayesian network was X_1 = `Characterized`, X_2 = `Intralaminar`, X_3 = `Intracolumnar`, X_4 = `Centered`, X_5 = `Common basket` and X_6 = `Dummy`.
- Cluster 3 (including four experts) represented experts who stuck to the fine-grained classification scheme proposed in the experiment and tried to distinguish between the different neuronal types, including the difficult ones such as `Common basket`, `Common type`, `Large basket` and `Arcade` cells. Experts in this cluster found more `Arcade` cells (0.07) than the experts in the other clusters. In this cluster, `Common type` (0.17), `Common basket` (0.14) and `Large basket` (0.20) cells had similar probabilities. The mode of the JPD encoded by the representative Bayesian network was the same as in cluster 2.
- Similarly to cluster 2, experts in cluster 4 (including 12 experts) showed a less detailed classification scheme than those in clusters 3 or 5. However, experts in cluster 4 assigned a high probability to the `Common type` class (0.40), whereas the most likely neuronal type in cluster 2 was `Common basket`. Accordingly, the mode of the JPD of the representative

Bayesian network was X_1 = Characterized, X_2 = Translaminar, X_3 = Intracolumnar, X_4 = Centered, X_5 = Common type and X_6 = Dummy.

- Cluster 5 represented a group of seven experts with a detailed classification scheme, since they distinguished between Common type, Common basket and Large basket cells. However, the experts did not seem to agree with the nomenclature included in the experiment or found it incomplete. This was observed in the high probability of the category Other (0.22) in X_6, where they could propose an alternative name for that class of neurons. Interestingly, the mode of the JPD of the representative Bayesian network for this cluster was X_1 = Uncharacterized, X_2 = Dummy, X_3 = Dummy, X_4 = Dummy, X_5 = Dummy and X_6 = Dummy. In fact, we can see that this cluster assigned the second highest probability to Uncharacterized in all the clusters.
- Cluster 6 included only one expert with a remarkably different behavior than the other experts. This expert did not classify any neuron as Translaminar in X_2, so the probability of that value in the representative Bayesian network is almost 0. Also, this expert assigned a very high probability to Centered in X_4 (0.96). Therefore, X_5 was disabled for all the neurons (recall that X_5 was only available when Translaminar and Displaced were set as values in X_2 and X_4, respectively). Therefore, X_5 had a constant Dummy value in Fig. 7(f). The conclusions of the analysis of the mode of the JPD were the same, as the combination of values of the variables with highest probability was X_1 = Characterized, X_2 = Intralaminar, X_3 = Transcolumnar, X_4 = Centered, X_5 = Large basket and X_6 = Dummy.

4.3. Analysis of the Bayesian network structures

Similarities in the behaviors of all the group of experts were identified by analyzing the representative Bayesian network structures. Variables X_3 and X_6 were the only two variables which were directly related in all the Bayesian networks. Variable X_3 describes the neuronal morphology in the horizontal dimension. This feature encodes whether or not the axonal arborization of the neuron extends more than 300 μm from the soma. This means that the interneuron contacts with neurons inside and outside its cortical column, so we could conclude that some neuronal types mainly connect with other neurons from the same cortical column, whereas other neuronal types connect additionally with neurons from different cortical columns.

Additionally, variables X_2, X_4 and X_5 were related in all but one Bayesian network, the one corresponding to cluster 6. Also, there was an edge between X_5 and X_6 in all the Bayesian networks but the one for cluster 6. Note that cluster 6 contained only the outlying expert 33. Variables X_2, X_4 and X_5 are mainly related to the neuronal morphology in the vertical dimension. These relationships could determine whether a given neuronal type sends the information to other neurons in the same cortical layer or in different (either upper or lower) layers. We also analyzed the Markov properties of the representative Bayesian network structures to identify conditional independence relationships between the variables. X_3 was conditionally independent of variables (X_2, X_4, X_5) given the value of X_6 and X_1. Therefore, the morphological properties of GABAergic interneurons in the horizontal and vertical dimensions seemed to be independent given the neuronal type.

4.4. Finding agreed definitions for neuronal types using inference in Bayesian networks

The representative Bayesian networks were used to infer the main properties of the different neuronal types in X_6 by setting evidence in some variables and updating the probabilities in the unobserved variables. We studied the propagated probabilities and identified differences and similarities between clusters. Cluster 6 corresponded to an outlier expert which has already been analyzed, so we focused on the other five clusters. First, the main morphological properties of the neuronal types were found by setting every value in X_6 as evidence and propagating the probabilities using the clustering algorithm [63,64] in GeNIe:

- Martinotti cells were defined as Translaminar ($\geq$0.94), Displaced ($\geq$0.83) and Ascending ($\geq$0.57) cells. Experts in cluster 5 classified these neurons as mostly Transcolumnar (0.73), whereas they were classified in clusters 1, 2, 3 and 4 as either Intracolumnar or Transcolumnar with similar probabilities.
- Horse-tail cells seem to have a common and easily recognizable morphology, since the most likely values achieved high probabilities in all the clusters: Translaminar ($\geq$0.92), Intracolumnar ($\geq$0.80), Displaced ($\geq$0.88) and Descending ($\geq$0.50).
- Chandelier cells seemed to be mainly Intracolumnar ($\geq$0.72). However, they were classified as either Intralaminar or Translaminar and Centered or Displaced in different clusters. Clusters 2 and 4 assigned a higher probability to Translaminar, cluster 3 assigned a higher probability to Intralaminar and the probabilities were almost uniform in the X_3 variable in clusters 1 and 5. Centered received a higher probability in cluster 3, whereas the probabilities were more uniform in the other clusters.
- Neurogliaform cells were defined as mainly Intracolumnar ($\geq$0.83). Experts in clusters 3, 4 and 5 classified them as Intralaminar ($\geq$0.76), whereas experts in clusters 1 and 2 assigned more uniform probabilities in variable X_2. For experts in cluster 5, Neurogliaform cells could be either Centered or Displaced, whereas Centered was more likely in all the other clusters ($\geq$0.75).

- Common type cells were characterized as Translaminar ($\geq$0.62) cells. Experts in clusters 4 and 5 classified them as either Intracolumnar or Transcolumnar, whereas experts in clusters 1, 2 and 3 selected Intracolumnar as the most likely value ($\geq$0.66).
- The properties for Common basket cells could not be easily identified. Experts in cluster 2 and 4 classified most of them as Translaminar ($\geq$0.63), cluster 3 assigned the highest probability to Intralaminar (0.82), whereas in the other clusters they were classified as either Translaminar or Intralaminar. Intracolumnar was always more likely than Transcolumnar, although the differences in the probability values greatly varied in the clusters. We also found major disagreements in X_4: Clusters 1 and 3 assigned Centered with a high probability ($\geq$0.86), whereas the probabilities of Centered and Displaced were similar in the other clusters.
- Large basket cells were characterized as Translaminar ($\geq$0.58) and Transcolumnar ($\geq$0.63) cells. Clusters 1 and 3 defined them as mainly Centered ($\geq$0.74), cluster 5 assigned a higher probability to Displaced (0.6), whereas in the other clusters Centered and Displaced had more uniform probabilities.
- Arcade cells were frequently classified as Translaminar ($\geq$0.65), Intracolumnar ($\geq$0.55) and, when Translaminar and Displaced were selected, as Descending cells.
- Most of the neurons classified as Other were characterized as Translaminar ($\geq$0.62). Intracolumnar was more likely than Transcolumnar in all the clusters. Also, Displaced had a higher probability than Centered in all the clusters, except for cluster 6. However, the differences between the probabilities of these values greatly varied from cluster to cluster. Cluster 3 yielded a high probability for Both category in X_5 (0.50), whereas cluster 1 assigned a greater probability to Descending (0.38). The probabilities in X_5 were more uniform in the other clusters.

Setting evidence in the other variables also highlighted some differences between groups of experts. For example, setting Intralaminar as evidence in X_2 yielded Common basket as the most likely value for X_6 in all the Bayesian networks, except for the one corresponding to cluster 4, where Common type and Neurogliaform got higher probabilities. Setting Translaminar as evidence in X_2 yielded very different propagated probabilities in the clusters. When setting Intracolumnar as evidence in X_3, the most likely values in X_6 were Common basket (clusters 1 and 2), Common type (clusters 3 and 4) and Other (cluster 5).

The consensus Bayesian multinet was used to perform inferences taking into account all the representative Bayesian networks at the same time. The probability of a given query was computed using the finite mixture of Bayesian networks expression (Eq. (6)). Table 1 shows the conditional probabilities of each variable given the neuronal type in X_6. We used these conditional probabilities to infer a set of agreed definitions for some neuronal types:

- Martinotti cells were usually classified as Translaminar, Displaced and Ascending.
- Horse-tail cells were commonly defined as Translaminar, Intracolumnar, Displaced and Descending neurons.
- A common feature of Chandelier neurons was that they were Intracolumnar.
- Neurogliaform cells were mainly Intralaminar, Intracolumnar and Centered cells.
- Common type cells were primarily Translaminar.
- Large basket neurons were characterized as Translaminar and Transcolumnar.
- Arcade neurons were usually classified as Translaminar.
- Neurons classified as Other were commonly classified as Translaminar and Intracolumnar cells.

4.5. Clustering visualization with PCA

The clusters obtained with K-means were visually inspected using a representation in a lower dimensional space. The goal was to obtain a three-dimensional representation that approximates the 121-dimensional JPDs and check whether or not the clusters were visually distinguishable. A principal component analysis (PCA) was performed, and the three principal components which account for the highest proportion of variance (67.14%) were studied [65]. Fig. 9 plots the values of the JPDs for each expert in the transformed three-dimensional space. Different symbols and colors were used to show the cluster assigned by the K-means algorithm to each expert. Two-dimensional projections were also included for ease of interpretation. Also, we studied the weights associated with each JPD value in each one of the principal components (PCs):

- The first PC, which accounted for 47.32% of the variance, distinguished the experts in cluster 1 from the other clusters. In this PC, the value of the JPD with highest (absolute) weight was X_1 = Uncharacterized, X_2 = Dummy, X_3 = Dummy, X_4 = Dummy, X_5 = Dummy, X_6 = Dummy (weight = 0.9828). The second weight with the largest absolute value had a value equal to -0.06119. This PC primarily separated experts with different behaviors when classifying the neurons as either Characterized or Uncharacterized in variable X_1. Therefore, the three experts in cluster 1 (Fig. 7(a)), which classified a lot of neurons as Uncharacterized, were easily distinguished using this PC.
- The second PC distinguished the outlying expert in cluster 6 and accounted for 10.74% of the variance. This PC yielded the largest weight (in absolute terms) for the value of the JPD corresponding to X_1 = Characterized, X_2 = Intralaminar,

Table 1
Conditional probabilities of each variable given the neuronal type (X_6), computed with the consensus Bayesian multinet. The largest value for each conditional probability distribution is highlighted in boldface.

	Common type	Horse-tail	Chandelier	Martinotti	Common basket	Arcade	Large basket	Cajal-Retzius	Neurogliaform	Other	Dummy
Conditional probabilities $P(X_1\|X_6)$											
Characterized	**0.9989**	**0.9981**	**0.9962**	**0.9988**	**0.9992**	**0.9937**	**0.9988**	**0.9835**	**0.9983**	**0.9950**	0.0115
Uncharacterized	0.0011	0.0019	0.0038	0.0012	0.0008	0.0063	0.0012	0.0165	0.0017	0.0050	**0.9885**
Conditional probabilities $P(X_2\|X_6)$											
Intralaminar	0.2847	0.0720	0.4270	0.0632	0.4477	0.2863	0.2642	0.2947	**0.6806**	0.2423	0.0072
Translaminar	**0.7136**	**0.9254**	**0.5671**	**0.9350**	**0.5511**	**0.7057**	**0.7342**	**0.6817**	0.3170	**0.7491**	0.0081
Dummy	0.0017	0.0026	0.0059	0.0018	0.0012	0.0080	0.0016	0.0236	0.0024	0.0086	**0.9847**
Conditional probabilities $P(X_3\|X_6)$											
Intracolumnar	**0.6190**	**0.8639**	**0.7903**	0.4001	**0.6862**	**0.6365**	0.1874	0.4687	**0.8589**	**0.7242**	0.0030
Transcolumnar	0.3802	0.1346	0.2065	**0.5990**	0.3132	0.3579	**0.8117**	**0.5165**	0.1398	0.2716	0.0030
Dummy	0.0008	0.0015	0.0032	0.0009	0.0006	0.0056	0.0009	0.0148	0.0013	0.0042	**0.9940**
Conditional probabilities $P(X_4\|X_6)$											
Centered	0.4293	0.1088	**0.5292**	0.1151	**0.6075**	0.4078	**0.5126**	0.3833	**0.7524**	0.3410	0.0052
Displaced	**0.5696**	**0.8893**	0.4668	**0.8837**	0.3917	**0.5856**	0.4862	**0.5997**	0.2459	**0.6540**	0.0055
Dummy	0.0011	0.0019	0.0040	0.0012	0.0008	0.0066	0.0012	0.0170	0.0017	0.0050	**0.9893**
Conditional probabilities $P(X_5\|X_6)$											
Ascending	0.1623	0.1244	0.0950	**0.6479**	0.1008	0.1290	0.1630	0.1852	0.0400	0.1859	0.0036
Descending	0.2169	**0.6439**	0.1961	0.1103	0.1270	0.2606	0.1762	0.2259	0.0369	0.2106	0.0036
Both	0.1296	0.1119	0.0754	0.1187	0.0790	0.1311	0.1096	0.1553	0.0302	0.2252	0.0036
Dummy	**0.4912**	0.1198	**0.6335**	0.1231	**0.6932**	**0.4793**	**0.5512**	**0.4336**	**0.8929**	**0.3783**	**0.9892**

X_3 = Transcolumnar, X_4 = Centered, X_5 = Large basket, X_6 = Dummy (weight = -0.7385). Fig. 7(f) shows that the representative Bayesian network of the outlying expert in cluster 6 had a very high probability (0.46) for Large Basket cells. Therefore, this PC separated the expert in cluster 6 from the rest of the clusters.

- The third PC accounted for 9.08% of the variance and could not easily separate the rest of the clusters. However, this PC seemed to be able to distinguish between experts in cluster 2 and experts in cluster 4. These two clusters contained experts with two different behaviors. Fig. 7(d) shows that the experts in cluster 4 classified most of the neurons as Common type (0.40), whereas the experts in cluster 2 (Fig. 7(b)) classified most of the neurons as Common basket (0.30). Clusters 3 and 5 were less distinguishable because the probability was more uniformly distributed across the values in X_6 (Fig. 7(c) and (e)). The weights in the third PC were also harder to interpret. However, cluster 4 and cluster 2 could be distinguished. All the values of the JPD with X_6 = Common type had weights smaller or equal than -0.02859, whereas all the values with X_6 = Common basket had weights greater or equal than -0.0102 (see Fig. 10). Therefore, the set of values with X_6 = Common type (cluster 4) and X_6 = Common basket (cluster 2) were disjoint according to the third PC.

We concluded that the behavior of experts in clusters 1 and 6 was remarkably different from the behavior of the rest of the experts. The K-means algorithm was able to identify those characterizing behaviors and generated two different clusters for them. Additionally, differences between the experts in clusters 2 and 4 were also correctly identified. The differences between clusters 3 and 5 were more subtle and it was difficult to find a three-dimensional representation of the JPDs which separated these experts.

4.6. Geographical identification of the clusters

We studied possible correlations between the experts' workplace and the cluster they were assigned to. The goal was to try to identify different approaches or currents of opinion regarding interneuron classification in different regions, cities or laboratories in the world. We studied the statistical significance of some of the groups of experts according to the country or the city where they worked. A bootstrapping approach was used, where a sample of experts was selected without replacement and we estimated the probability of some of them belonging to the same clusters. The sampling procedure was repeated 100,000 times for different sample sizes. We could not find any statistically significant result using a significance level of $\alpha = 0.05$. Therefore, we concluded that there is no geographical correlation between the experts and the cluster they were assigned to.

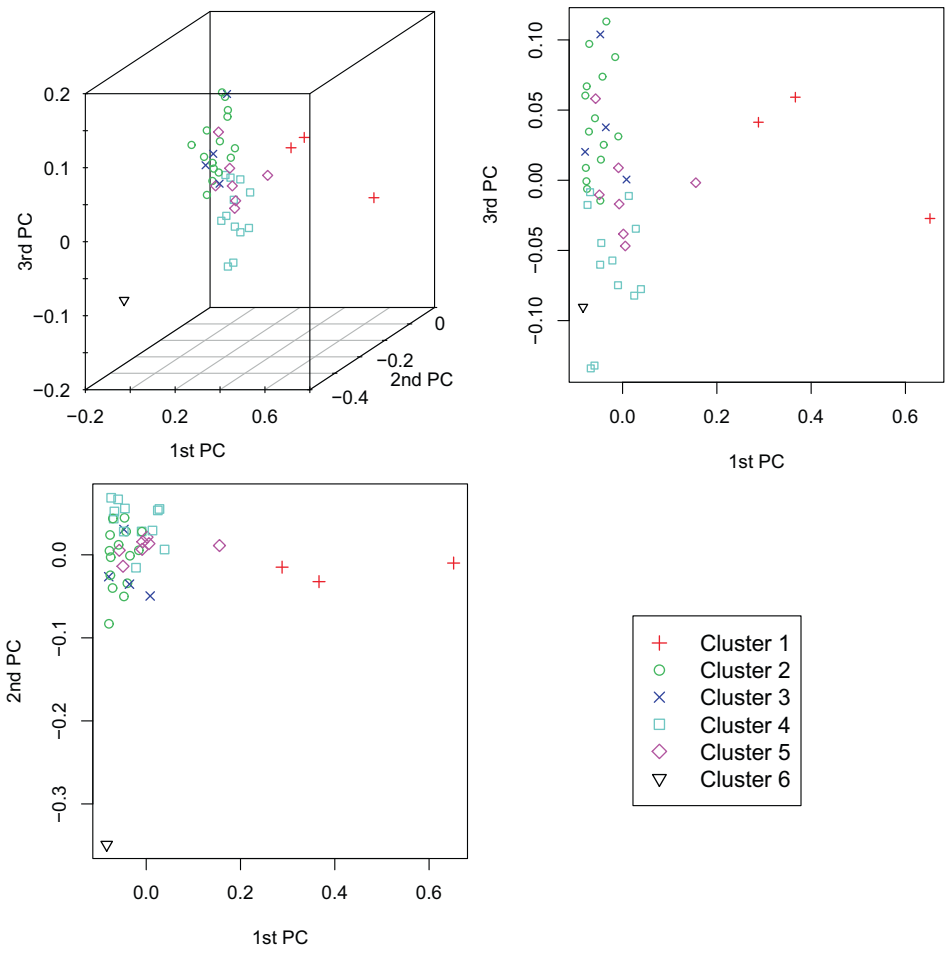


Fig. 9. Visualization of the clusters computed with K-means ($K = 6$) in three and two-dimensional spaces obtained with principal component analysis. The three-dimensional coordinates of the experts correspond to the values of the three principal components with highest proportion of variance.

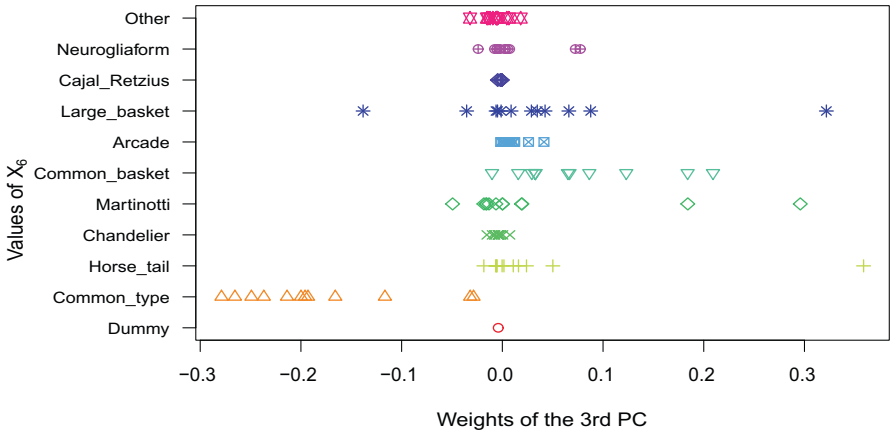


Fig. 10. Weights of the third principal component according to the value of the variable X_6 in the JPD.

5. Conclusions

Bayesian networks have been successfully applied to a wide range of problems in very different domains. In this paper, we have presented a methodology for building a consensus Bayesian multinet that represents the opinions of a set of experts. The methodology can be summarized in three steps. First, a Bayesian network is learned for each expert. Second, the Bayesian networks corresponding to experts with similar behaviors are clustered. Third, a consensus Bayesian multinet is built which

models the behavior of all experts. To the best of our knowledge, the problem of clustering Bayesian networks had not been studied in the literature before. Therefore, this work also addresses an interesting problem in Bayesian network research. Our proposal consists of computing the JPD encoded by each Bayesian network and applying a partitional clustering approach to find groups of similar JPDs. The K-means algorithm with logarithmic combination pool and Jensen–Shanon divergence was used to cluster the JPDs. Then, a representative Bayesian network was induced for each cluster by proportional sampling of the JPDs in the cluster and applying a Bayesian network learning algorithm on the generated dataset. The final model is a consensus Bayesian multinet which encodes a finite mixture of Bayesian networks, where each component is the representative Bayesian network for a cluster of experts.

We applied the proposed methodology to a problem of modeling experts' opinions when classifying a set of cortical GABAergic interneurons based on the morphological features of their reconstructions [22]. This is a difficult task because neuroscientists do not have a set of commonly agreed definitions which clearly distinguish the different neuronal types [3]. The consensus Bayesian multinet built in this work was analyzed to gain some insights into the problem of classifying GABAergic interneurons. We managed to find some easily distinguishable clusters of experts, which behaved differently from the rest of the clusters. By studying the marginal probabilities in the representative Bayesian networks, we were able to identify different approaches to neuron classification. This highlighted the importance of clustering the experts before building the consensus model. Directly combining experts with such differing behaviors would presumably hide some of these opinions, and the final model would not represent all the experts thoughts. Additionally, we performed inference with the model to provide agreed definitions of the neuronal types. We analyzed the representative Bayesian network structures to identify common conditional independence relationships between the groups of experts, which had a direct biological interpretation.

We discussed the different approaches that could be considered for performing each one of the tasks and motivated our decisions in each step. One advantage of the proposed methodology is that it is data independent, i.e., we only used experts' data for learning the initial Bayesian networks, and the clustering algorithm and the construction of the consensus Bayesian multinet are only based on these Bayesian networks. Therefore, the methodology can still be applied when experts' data is not available, e.g., when Bayesian networks are elicited from experts' knowledge.

Future work includes the application of different techniques for clustering JPDs. In particular, model-based clustering using finite mixtures of distributions was discarded because of the low number of observations (42 experts) and the high dimensionality of the JPDs (121 values). We discussed the possible use of finite mixtures of Dirichlet distributions [44] as the most straightforward model for clustering probability distributions. However, the Dirichlet distribution has some constraints, e.g., its covariance matrix is strictly negative so it cannot model positive correlations between variables. Finite mixtures of generalized Dirichlet distributions [66] overcome some of these constraints. However, the generalized Dirichlet distribution has more parameters than the Dirichlet distribution, so the problem of high-dimensionality combined with few data is even more challenging. L_1-regularization approaches in finite mixture modeling [67,68] could be used to perform feature subset selection and reduce data dimensionality. Another point for future research is the use of different techniques for finding a representative Bayesian network for each cluster. Some of the possibilities for achieving this goal were discussed in Section 3.2.3. Finding a consensus Bayesian network from a set of Bayesian networks which represent different experts' opinions has been a recurring interest in the field [52–56]. Combining these methods with the proposed clustering approach and studying the differences and similarities in the representative Bayesian networks obtained for each cluster in a real problem could give some insights into the relative merits of each technique.

This paper serves as an example of the kind of synergies stemming from multidisciplinary research when real problems drive research in artificial intelligence. The strategy developed in this study could be useful for identifying and correlating morphological parameters that are difficult, if not impossible, to obtain by visual inspection alone. Furthermore, including other neuronal features, such as their molecular and physiological features, would result in a powerful tool for classifying neurons with a more functional significance. Indeed, we expect these models to play a major role in solving some of the most challenging problems in biology and medicine, such as the study of low-level interactions between neurons in the brain and their relationships to perception, learning or brain diseases.

Acknowledgements

This work has been partially supported by the Spanish Economy and Competitiveness Ministry through Cajal Blue Brain (C080020-09), TIN2010-20900-C04-04 and Consolider Ingenio 2010-CSD2007-00018 projects. PLLC is supported by an University Lecturer Training Fellowship (FPU AP2009-1772) from the Spanish Education, Culture and Sports Ministry. The authors thank Rubén Armañanzas and Diego Vidaurre for useful comments, and the set of experts for participating in the experiment.

References

[1] R.B. Stein, E.R. Gossen, K.E. Jones, Neuronal variability: noise or part of the signal?, Nature Reviews Neuroscience 6 (2005) 389–397.
[2] S.G. Sadeghi, M.J. Chacron, M.C. Taylor, K.E. Cullen, Neural variability, detection thresholds, and information transmission in the vestibular system, Journal of Neuroscience 27 (2007) 771–781.
[3] Petilla Interneuron Nomenclature Group, Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex, Nature Reviews Neuroscience 9 (2008) 557–568.

[4] M. Ding, D. Glanzman, The Dynamic Brain: An Exploration of Neuronal Variability and its Functional Significance, Oxford University Press, 2011.
[5] J. DeFelipe, Cortical interneurons: from Cajal to 2001, Progress in Brain Research 136 (2002) 215–238.
[6] G.A. Ascoli, Successes and rewards in sharing digital reconstructions of neuronal morphology, Neuroinformatics 5 (2007) 154–160.
[7] G.A. Ascoli, D.E. Donohue, M. Halavi, Neuromorpho.org: a central resource for neuronal morphologies, Journal of Neuroscience 27 (2007) 9247–9251.
[8] M. Bota, L.W. Swanson, The neuron classification problem, Brain Research Reviews 56 (2007) 79–88.
[9] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufman, 1988.
[10] D. Koller, N. Friedman, Probabilistic Graphical Models, Principles and Techniques, The MIT Press, 2009.
[11] O. Pourret, P. Naïm, B. Marcot, Bayesian Networks: A Practical Guide to Applications, Wiley, 2008.
[12] M.J. Flores, A.E. Nicholson, A. Brunskill, K.B. Korb, S. Mascaro, Incorporating expert knowledge when learning Bayesian network structure: a medical case study, Artificial Intelligence in Medicine 53 (2011) 181–204.
[13] P.H. Garthwaite, J.B. Kadane, A. O'Hagan, Statistical methods for eliciting probability distributions, Journal of the American Statistical Association 100 (2005) 680–701.
[14] K.A. Greene, J.M. Kniss, G.F. Luger, C.R. Stern, Satisficing the masses: applying game theory to large-scale, democratic decision problems, in: International Conference on Computational Science and Engineering, IEEE Computer Society, 2009, pp. 1156–1162.
[15] K.A. Greene, J.M. Kniss, G.F. Luger, Representing diversity in communities of Bayesian decision-makers, in: IEEE Second International Conference on Social Computing, IEEE Computer Society, 2010, pp. 315–322.
[16] D. Geiger, D. Heckerman, Knowledge representation and inference in similarity networks and Bayesian multinets, Artificial Intelligence 82 (1996) 45–74.
[17] B. Thiesson, C. Meek, D.M. Chickering, D. Heckerman, Learning mixtures of DAG models, in: G.F. Cooper, S. Moral (Eds.), Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998, pp. 504–513.
[18] M. Meilă, M.I. Jordan, Learning with mixtures of trees, Journal of Machine Learning Research 1 (2000) 1–48.
[19] E. Santos, A. Hussein, Case-based Bayesian network classifiers, in: V. Barr, Z. Markov (Eds.), Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2004, pp. 538–543.
[20] A. Hussein, E. Santos, Exploring case-based Bayesian networks and Bayesian multi-nets for classification, in: A.Y. Tawfik, S.D. Goodwin (Eds.), Proceedings of the Seventeenth Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI-2004), Lecture Notes in Computer Science, vol. 3060, Springer, 2004, pp. 485–492.
[21] J. Cheng, R. Greiner, Learning Bayesian belief network classifiers: algorithms and system, in: E. Stroulia, S. Matwin (Eds.), Proceedings of the Fourteenth Biennial Conference of the Canadian Society for Computational Studies of Intelligence, Lecture Notes in Computer Science, vol. 2056, Springer, 2001, pp. 141–151.
[22] J. DeFelipe, P.L. López-Cruz, R. Benavides-Piccione, C. Bielza, P. Larrañaga, S. Anderson, A. Burkhalter, B. Cauli, A. Fairén, D. Feldmeyer, G. Fishell, D. Fitzpatrick, T.F. Freund, G. González-Burgos, S. Hestrin, S. Hill, P.R. Hof, J. Huang, E.G. Jones, Y. Kawaguchi, Z. Kisvárday, Y. Kubota, D.A. Lewis, O. Marín, H. Markram, C.J. McBain, H.S. Meyer, H. Monyer, S.B. Nelson, K. Rockland, J. Rossier, J.L.R. Rubenstein, B. Rudy, M. Scanziani, G.M. Shepherd, C.C. Sherwood, J.F. Staiger, G. Tamás, A. Thomson, Y. Weng, R. Yuste, G.A. Ascoli, New insights into the classification and nomenclature of cortical GABAergic interneurons, Nature Reviews Neuroscience 14 (2013) 202–216.
[23] A. Doan, R. Ramakrishnan, A.Y. Halevy, Crowdsourcing systems on the World-Wide Web, Communications of the ACM 54 (2011) 86–96.
[24] R. Cannon, D. Turner, G. Pyapali, H. Wheal, An on-line archive of reconstructed hippocampal neurons, Journal of Neuroscience Methods 84 (1998) 49–54.
[25] A. Peters, E.G. Jones, Cerebral Cortex, in: Cellular Components of the Cerebral Cortex, vol. 1, Plenum Press, 1984
[26] W. Buntine, A guide to the literature on learning probabilistic networks from data, IEEE Transactions on Knowledge and Data Engineering 8 (1996) 195–210.
[27] D. Heckerman, A tutorial on learning with Bayesian networks, Technical Report MSR-TR-95-06, Microsoft Corporation, 1996.
[28] R. Daly, Q. Shen, S. Aitken, Learning Bayesian networks: approaches and issues, The Knowledge Engineering Review 26 (2011) 99–157.
[29] D. Dash, G. Cooper, Model averaging for prediction with discrete Bayesian networks, Journal of Machine Learning Research 5 (2004) 1177–1203.
[30] G. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (1992) 309–347.
[31] P. Larrañaga, C.M.H. Kuijpers, R.H. Murga, Y. Yurramendi, Learning Bayesian network structures by searching for the best ordering with genetic algorithms, IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans 26 (1996) 487–493.
[32] N. Friedman, D. Koller, Being Bayesian about network structure: a Bayesian approach to structure discovery in Bayesian networks, Machine Learning 50 (2003) 95–125.
[33] J.M. Peña, J.A. Lozano, P. Larrañaga, I. Inza, Dimensionality reduction in unsupervised learning of conditional Gaussian networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 590–603.
[34] D.T. Pham, G.A. Ruz, Unsupervised training of Bayesian networks for data clustering, Proceedings of the Royal Society A (Mathematical, Physical & Engineering Sciences) 465 (2009) 2927–2948.
[35] S. Jung, K.H. Lee, D. Lee, Enabling large-scale Bayesian network learning by preserving intercluster directionality, IEICE Transactions on Information and Systems E90-D (2007) 1018–1027.
[36] B. Chen, Q. Liao, Z. Tang, A clustering based Bayesian network classifier, in: Proceedings of the IEEE Fourth International Conference on Fuzzy Systems and Knowledge Discovery, IEEE Computer Society, 2007, pp. 444–448.
[37] A. Robles-Kelly, E.R. Hancock, Graph edit distance from spectral seriation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 365–378.
[38] X. Gao, B. Xiao, D. Tao, X. Li, A survey of graph edit distance, Pattern Analysis & Applications 13 (2010) 113–129.
[39] A. Gerasoulis, T. Yang, A comparison of clustering heuristics for scheduling directed acyclic graphs on multiprocessors, Journal of Parallel and Distributed Computing 16 (1992) 276–291.
[40] T.V. Van, T. Pham-Gia, Clustering probability distributions, Journal of Applied Statistics 37 (2010) 1891–1910.
[41] A. Goh, R. Vidal, Unsupervised Riemannian clustering of probability density functions, in: W. Daelemans, B. Goethals, K. Morik (Eds.), Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Springer, 2008, pp. 377–392.
[42] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, 2001.
[43] G.J. McLachlan, D. Peel, Finite Mixture Models, Wiley, 2000.
[44] N. Bouguila, D. Ziou, J. Vaillancourt, Unsupervised learning of a finite mixture model based on the Dirichlet distribution and its application, IEEE Transactions on Image Processing 13 (2004) 1533–1543.
[45] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1967, pp. 281–297.
[46] S. Kullback, R.A. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (1951) 79–86.
[47] J. Lin, Divergence measures based on the Shannon entropy, IEEE Transactions on Information Theory 37 (1991) 145–151.
[48] C. Genest, J.V. Zideck, Combining probability distributions: a critique and an annotated bibliography, Statistical Science 1 (1986) 114–135.
[49] R. Bordley, A multiplicative formula for aggregating probability assessments, Management Science 28 (1982) 1137–1148.
[50] G.U. Yule, Notes on the theory of association of attributes in statistics, Biometrika 2 (1903) 121–134.
[51] T. Heskes, Selecting weighting factors in logarithmic opinion pools, in: Advances in Neural Information Processing Systems, The MIT Press, 1998, pp. 266–272.
[52] D.M. Pennock, M.P. Wellman, Graphical representations of consensus belief, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman, 1999, pp. 531–540.
[53] J. Sagrado, S. Moral, Qualitative combination of Bayesian networks, International Journal of Intelligent Systems 18 (2003) 237–249.
[54] Y. Zhang, K. Yue, M. Yue, W. Liu, An approach for fusing Bayesian networks, Journal of Information & Computational Science 8 (2011) 194–201.
[55] J.M. Peña, Finding consensus Bayesian network structures, Journal of Artificial Intelligence Research 42 (2011) 661–687.
[56] I. Matzkevich, B. Abramson, The topological fusion of Bayes nets, in: D. Dubois, M.P. Wellman (Eds.), Proceedings of the Eight Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman, 1992, pp. 191–198.

[57] I. Matzkevich, B. Abramson, Deriving a minimal I-map of a belief network relative to a target ordering of its nodes, in: D. Heckerman, E.H. Mamdani (Eds.), Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman, 1993, pp. 159–165.
[58] P. Maynard-Reich II, U. Chajewska, Aggregating learned probabilistic beliefs, in: J.S. Breese, D. Koller (Eds.), Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman, 2001, pp. 354–361.
[59] R. Chen, K. Sivakumar, H. Kargupta, Collective mining of Bayesian networks from distributed heterogeneous data, Knowledge and Information Systems 6 (2004) 164–187.
[60] M. Scutari, Learning Bayesian networks with the bnlearn R package, Journal of Statistical Software 35 (2010) 1–22.
[61] R Development Core Team, R: a language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.
[62] G.E. Schwarz, Estimating the dimension of a model, Annals of Statistics 6 (1978) 461–464.
[63] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems (with discussion), Journal of the Royal Statistical Society, Series B (Methological) 50 (1988) 157–224.
[64] F.V. Jensen, S.L. Lauritzen, K.G. Olsen, Bayesian updating in causal probabilistic networks by local computations, Computational Statistics Quarterly 4 (1990) 269–282.
[65] K. Pearson, On lines and planes of closest fit to systems of points in space, Philosophical Magazine 2 (1901) 559–572.
[66] N. Bouguila, D. Ziou, High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 1716–1731.
[67] W. Pan, X. Shen, Penalized model-based clustering with application to variable selection, Journal of Machine Learning Research 8 (2007) 1145–1164.
[68] B. Xie, W. Pan, X. Shen, Variable selection in penalized model-based clustering via regularization on grouped parameters, Biometrics 64 (2008) 921–930.