## Supplementary Online Information S1

A set of experts on cortical interneurons from different laboratories were asked to classify morphological reconstructions of interneurons (http://cajalbbp.cesvima.upm.es/gardenerclassification). The main characteristics of the experiment were:

- Number of experts who started the experiment: 48. The number of experts who completed the experiment was 42. We used only data from these completed experiments for our analyses.
- Number of neurons: 320.
    - 241 are 3D reconstructions from NeuroMorpho.Org (Ascoli et al., 2007).
    - 79 are 2D images scanned from older papers.
- Number of features for each neuron: 6. Total number of categories (each possible value of every feature): 21.
    1. Categories for Feature 1 (F1): Intralaminar vs Translaminar
    2. Categories for Feature 2 (F2): Intracolumnar vs Transcolumnar
    3. Categories for Feature 3 (F3): Centered vs Displaced
    4. Categories for Feature 4 (F4): Ascending vs Descending vs Both
    5. Categories for Feature 5 (F5) (interneuron type): Common type, Horse-tail, Chandelier, Martinotti, Common basket, Arcade, Large basket, Cajal-Retzius, Neurogliaform or Other
    6. Categories for Feature 6 (F6): Characterized vs Uncharacterized

Feature 6 models whether or not the labeled part of the neuron's morphology allows for its categorization in the rest of the features. When an expert selects Uncharacterized for a given neuron, s/he cannot provide values for any of the other features. Considering the opposite case, assigning a category to any feature from F1 to F5 implies that the neuron is Characterized, i.e., enough morphological detail is available to allow its categorization. Additionally, Feature 4 is only available when Translaminar is selected in Feature 1 and Displaced is selected in Feature 3. Therefore, the number of values available for each feature can differ between neurons, depending on the values selected by the experts in other features.

In the analysis, we first applied statistical techniques on the experts' selections. These analyses were used to study the agreement among the experts at both the feature and category levels. Different machine learning techniques were then used to confirm the agreement results and extract knowledge from the data. Clustering algorithms were applied to find groups of easily distinguishable neurons and to identify their defining properties. We used a Bayesian network approach to model the statistical relationships between the features and to study the underlying reasoning of each expert. Finally, supervised classification algorithms were run to induce models that were able to automatically classify the neurons taking into account an exhaustive set of morphological measurements of their three-dimensional reconstructions.

### Experts' agreement analysis

We analyzed the agreement achieved by the 42 experts who completed the experiment involving the classification of the 320 neurons according to the six features. First, we computed the overall inter-expert agreement, as well as the chance-corrected Fleiss' pi agreement index (Fleiss, 1971) for each feature independently. Fleiss' pi index adjusts the observed agreement by subtracting the agreements between experts that are due to chance alone. Also, we computed the inter-expert agreement and Fleiss' pi index for each category of every feature, taking into account all the experts' ratings (Cicchetti and Feinstein, 1990).

We studied the sensitivity of the agreement to the set of experts, i.e. whether or not removing a small number of experts would significantly modify the agreement values. We wanted to identify whether any experts showed different choice behavior from the rest of the group (outliers). Therefore, we computed Fleiss' pi index when 1, 2 or 3 of the experts were removed from the analysis.

Also, we analyzed possible overlaps between categories of Feature 5. Pairs and triples of interneuron types were merged into one single category, and Fleiss' pi index was computed for all of these scenarios. We identified where combining interneuron types increased or decreased the agreement. If experts frequently confused two or more interneuron types, the agreement value increased when considering those interneurons as the same category. This was for example the case for categories Common type and Common basket (**Supplementary Online Information S2**). On the contrary, merging easily distinguishable pairs or combinations of interneuron types yielded decrements in the agreement. This was for example the case for categories Martinotti and Common basket (**Supplementary Online Information S2**).

Additionally, we studied the agreement between each possible pair of experts for every feature using three different indices: Cohen's kappa (Cohen, 1960), Prevalence-Adjusted Bias-Adjusted (PABA) kappa (Byrt et al., 1993), and the ratio between Cohen's kappa and its maximum value given fixed marginals (Dunn, 1989). These indices can only be applied to binary features, while Features 4 and 5 are non-binary. Thus, for Feature 5, we used these indices to measure the agreement between each category versus all the other categories considered together. This allowed us to study each category independently for each pair of experts. We proceeded similarly with Feature 4.

**Agreement indices**

For each of the six features, we ran a classification experiment, in which a group of $R$ experts classified a set of $M$ items (neurons) into $Q$ categories. The goal was to measure and analyze the degree of agreement between experts when categorizing the items. We denote $r_{iq}$ as the number of experts who assigned the $i^{th}$ neuron to category $q$:

$$r_i = \sum_{q=1}^{Q} r_{iq}.$$

We denote $n_{jq}$ as the number of items that expert $j$ assigned to category $q$.

***Overall observed agreement***

The most straightforward way to assess consensus is by computing the observed agreement:

$$P_o = \frac{\sum_{q=1}^{Q}\sum_{i=1}^{M} r_{iq}(r_{iq}-1)}{\sum_{i=1}^{M} r_i(r_i-1)}.$$

This overall observed agreement has been widely criticized in the literature (e.g., Carletta, 1996). The observed agreement favors experiments with a low number of categories, $Q$. In addition, it does not take into account the different distributions of items among categories. Two solutions have been proposed to this problem: 1) adjusting the observed agreement for chance agreement; and 2) computing category-specific observed agreement.

***Chance-corrected agreement coefficients***

A solution to the problem of analyzing the agreement between experts in a classification study is correcting the observed value to erase the influence of chance agreements. Popping (1988) identified more than 40 different proposals of chance-corrected agreement indices. In general, most of the chance-corrected agreement indices have the following expression:

$$A = \frac{A_o - A_e}{1 - A_e}, \qquad (1)$$

where $A_o$ is the observed agreement and $A_e$ is the expected agreement by chance. The numerator encodes the observed agreement beyond chance, and the denominator encodes the maximum agreement that can be achieved beyond chance. An index value of $A = 1$ means perfect agreement, whereas a value of $A = 0$ shows chance agreement. Negative values of $A$ indicate agreement below chance. In this study we applied the two most studied agreement indices: Cohen's kappa (and some of its variants) and Fleiss' pi indices.

**Cohen's kappa:** Cohen's kappa index (Cohen, 1960) is defined for a two-expert ($R = 2$) and two-category ($Q = 2$) experiment. Only those items classified by both experts are considered. The results of the experiment can be reported as a cross-classification table (**Table S1**):

**Table S1.** Cross-classification table for an experiment with two experts and two categories (+ and –).

| | | Expert 2 | | |
|---|---|---|---|---|
| | | + | - | Frequency |
| Expert 1 | + | *a* | *b* | $n_{1+}$ |
| | - | *c* | *d* | $n_{1-}$ |
| Frequency | | $n_{2+}$ | $n_{2-}$ | *M* |

Cohen's kappa index has the structure of Equation (1), with

$$A_o^\kappa = \frac{a+d}{M}$$

and

$$A_e^\kappa = \frac{n_{1+}n_{2+} + n_{1-}n_{2-}}{M^2}.$$

Cohen's kappa index is developed under three assumptions: 1) the classified items are independent, 2) the categories are independent, exhaustive, and mutually exclusive, and 3) the experts operate independently and have different distributions for the categories. Cohen's kappa index is negatively affected by the different prevalence of the categories (prevalence problem) and by the degree of disagreement between the two experts (bias problem) (e.g., Feinstein and Cicchetti, 1990). Interpreting the magnitude of Cohen's kappa index is challenging because of these effects. Several standards have been proposed for interpreting the strength of agreement (e.g., Landis and Koch, 1977). These approaches are necessarily subjective and arbitrary, since the interpretation of Cohen's index depends on the field of science, the nature of the experiment, and the prevalence and bias effects in the data (Artstein and Poesio, 2008). Some variants of Cohen's kappa index follow.

- **Prevalence-Adjusted Bias-Adjusted kappa index:** Byrt et al. (1993) propose a Prevalence-Adjusted Bias-Adjusted kappa index (PABAκ) to minimize the effects of prevalence and bias. This value can be reported alongside Cohen's kappa to show the effects of prevalence and bias on the index value and to determine the sources of disagreement. To compute PABAκ, the cross-classification table is modified as in **Table S2**. The agreement cells *a* and *d* (main diagonal in **Table S1**) are changed to their mean *(a+d)/2*, removing the prevalence effect. The disagreement cells *c* and *b* (secondary diagonal in **Table S1**) are also changed to their mean value *(b+c)/2*, adjusting for the bias effect. PABAκ is Cohen's kappa index computed with the values in the modified **Table S2**.

**Table S2.** Modified cross-classification table for minimizing prevalence and bias effects and computing PABAκ.

| | | Expert 2 | | |
|---|---|---|---|---|
| | | + | - | Frequency |
| Expert 1 | + | *(a+d)/2* | *(b+c)/2* | $n'_{1+}$ |
| | - | *(b+c)/2* | *(a+d)/2* | $n'_{1-}$ |
| | Frequency | $n'_{2+}$ | $n'_{2-}$ | *M* |

- **Maximum kappa index:** Another approach for interpreting Cohen's kappa value is comparing it to the maximum value $\kappa_{max}$ that can be achieved when the marginal frequencies of each expert are fixed (Sim and Wright, 2005). To compute $\kappa_{max}$, a modified cross-classification table is used, where the agreement cells (main diagonal) are set to the minimum of the marginal frequencies for their corresponding categories, as in **Table S3**. The disagreement cells (secondary diagonal) are adjusted to maintain the marginal frequencies. The value of $\kappa_{max}$ shows the maximum possible agreement taking into account the different prevalence and bias of the experts. The ratio $\kappa/\kappa_{max}$ is usually used to measure the proportion of agreement that was achieved in the experiment taking into account the differences between experts.

**Table S3.** Modified cross-classification table for computing $\kappa_{max}$.

| | | Expert 2 | | |
|---|---|---|---|---|
| | | + | - | Frequency |
| Expert 1 | + | $min(n_{1+}, n_{2+})$ | $n_{1+} - min(n_{1+}, n_{2+})$ | $n_{1+}$ |
| | - | $n_{2+} - min(n_{1+}, n_{2+})$ | $min(n_{1-}, n_{2-})$ | $n_{1-}$ |
| | Frequency | $n_{2+}$ | $n_{2-}$ | *M* |

**Fleiss' pi index:** When more than two experts join the experiment (*R>2*), Fleiss' (1971) generalization of Scott's (1955) pi index is the most commonly used chance-corrected agreement coefficient. When missing values are allowed (not all the experts have to classify all the items), Fleiss' pi can be adapted to give equal weight to each judgment or equal weight to each item (Artstein and Poesio, 2008). Giving an equal weight to each item, Fleiss' pi follows the structure in Equation (1) with

$$A_o^{\pi} = \frac{1}{M}\sum_{i=1}^{M}\sum_{q=1}^{Q}\frac{r_{iq}(r_{iq}-1)}{r_i(r_i-1)},$$

and

$$A_e^{\pi} = \sum_{q=1}^{Q}\left(\frac{1}{M}\sum_{i=1}^{M}\frac{r_{iq}}{r_i}\right)^2.$$

Fleiss' pi assumes that the marginal distribution of the categories is the same for each expert given the assumption that they are operating by chance. This is the main difference with Cohen's kappa index, where it is assumed that the marginal distributions of the categories for each expert are different.

***Category-specific agreement indices***
We can also study inter-expert agreement (observed and chance-corrected Fleiss' pi index) for each category in each feature individually.

**Observed agreement:** We can compute specific observed agreement values for each category *q=1,…,Q*, using a similar approach as with specificity and sensitivity:

$$P_{oq} = \frac{\sum_{i=1}^{M} r_{iq}(r_{iq} - 1)}{\sum_{i=1}^{M} r_{iq}(r_i - 1)}.$$

Several authors advocate the use of these category-specific indices (e.g., Cicchetti and Feinstein, 1990). Reporting these category-specific indices overcomes the problems of the overall observed agreement and avoids the need to correct for chance agreement.

**Chance-corrected Fleiss' pi index**: A different chance-corrected agreement index can be computed for each category using Fleiss' pi index. The chance-corrected agreement for a category $q = 1,\ldots,Q$ is given by Equation (1) with

$$A_o^{\pi_q} = \frac{\sum_{i=1}^{M} r_{iq}(r_{iq} - 1)}{\sum_{i=1}^{M} r_{iq}(r_i - 1)},$$

and

$$A_e^{\pi_q} = \frac{1}{M}\sum_{i=1}^{M} \frac{r_{iq}}{r_i}.$$

***Statistical tests for chance agreement***
We performed a permutation test to check whether or not the values of the agreement indices explained above indicated an agreement above chance. A random experiment was generated by sampling categories for each feature maintaining the relative frequency of the categories in the complete experiment (shown in **Figure 3A** of the main text). For each expert and each neuron, we sampled a category for Feature 6. If the sampled category was *Characterized*, then categories for Features 1-3 and 5 were sampled. When *Translaminar* and *Displaced* categories were sampled for Features 1 and 3, then a category was randomly sampled for Feature 4. Ten thousand random experiments were generated and the observed agreement, Fleiss' pi and category-specific Fleiss' pi indices were computed. Then, the agreement values using the real classification data provided by the experts was compared to the cumulative distribution of the values of the agreement indices obtained with the randomly generated experiments. Statistical significance was established at a significance level $\alpha = 0.05$.

**Neuron clustering**
We applied unsupervised classification (clustering) algorithms to find groups of interneurons with similar characteristics according to the classifications provided by the experts.

***Neuron clustering for each feature***
First, we wanted to generate clusters for the set of *M* neurons considering each feature independently. For a given feature, we used the category assigned for each expert to each neuron (category value $q = 1,\ldots,Q$) as information for the clustering. Therefore, the dataset used for the clustering algorithm had 320 instances (neurons), where each instance is an *N*-dimensional vector ($N = 42$ experts).

We applied the *k*-modes algorithm (Huang, 1998), an extension of the *k*-means algorithm (MacQueen, 1967) that manages categorical data. **Algorithm 1** sketches the main steps of the *k*-means algorithm, which are the same as in the *k*-modes algorithm. The goal of the *k*-means

algorithm is to find the $k$ cluster centers $\boldsymbol{C} = \{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_k\}$ that minimize a measure of dissimilarity, where $k>1$ is a parameter of the algorithm indicating the number of clusters. For Features 1-3 and Feature 6 a number of clusters $k = 2$ was used. For Feature 4, three clusters ($k = 3$) were selected. Different numbers of clusters (six to ten) were analyzed for Feature 5. The clearest results were obtained with $k$=8. A neuron is assigned to the cluster with the closest center. Therefore, the fitness function to minimize is the sum of the distance of each item to the center of its cluster.

For categorical data, $k$-modes uses the Hamming distance to measure the distance between two items (neurons) or between an item and a cluster center. The set of cluster centers $\boldsymbol{C}$ is found by computing the modes of the items belonging to the cluster. Ties when computing the modes or when assigning items to clusters are broken randomly. In our implementation, the algorithm stopped when no change in the cluster centers occurred or when the fitness function had the same value for 100 consecutive iterations.

**Algorithm 1.** $k$-means clustering algorithm.
Input:

- $k$, number of clusters.
- Dataset of $N$-dimensional items $\boldsymbol{x}_i$, $i = 1,\ldots, M$.

Steps:

1. Initialize the $k$ cluster centers $\boldsymbol{C}$ to $k$ random items $\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(k)}$.
2. While cluster centers $\boldsymbol{C}$ change
    a. Assign each item $\boldsymbol{x}_i$ to the corresponding cluster with the closest center.
    b. Recompute $\boldsymbol{C}$ from the items in the cluster.
3. Return $\boldsymbol{C}$.

The $k$-means algorithm (and also $k$-modes) can yield suboptimal solutions if it gets stuck in local minima. To avoid this situation, the algorithm was run 25 times with different initial values for the cluster centers in step 1 of **Algorithm 1**. The best result (minimum fitness function value) is reported in the Results section. A color checkerboard was used to represent the clusters, where each column represents a neuron and a color point identifies the category selected by each rater. The algorithms were implemented and run in Matlab.

***Neuron clustering for all the features***

We wanted to generate clusters of cells taking into account the agreement of the experts in all the features at the same time. For every neuron, we computed the number of experts that assigned the neuron to each category of every feature. Therefore, the dataset used in the clustering algorithm had 320 instances (neurons), and each instance was an $N$-dimensional vector ($N = 21$), corresponding to all the categories of the six features: Intralaminar, Translaminar, Intracolumnar, Transcolumnar, Centered, Displaced, Ascending, Descending, Both, Common type, Horse-tail, Chandelier, Martinotti, Common basket, Arcade, Large basket, Cajal-Retzius, Neurogliaform, Other, Characterized, and Uncharacterized.

We used the $k$-means algorithm to cluster cells according to the number of votes each neuron had in each category. Different numbers of clusters (six to ten) were analyzed. The clearest results were obtained with $k$=6. For continuous data, $k$-means uses Euclidean distance to compute the distance between every two items. Every time step 2b in **Algorithm 1** is performed, $k$-means computes the cluster centers as the centroid of the items in the cluster.

The algorithm was run 25 times with different initial values for the cluster centers to avoid local optima, similarly to $k$-modes, and the best result was shown. The clusters were illustrated using parallel coordinate diagrams (Wegman, 1990). Each line represents one neuron in the cluster and its height shows the number of experts who selected each category for that neuron. A small amount of noise drawn from a normal distribution (mean = 0, standard deviation = 0.75) was added to the values to ensure that all lines were visible.

**Bayesian networks for modeling experts' opinions**

We trained one Bayesian network (Pearl, 1988) on data from each expert, modeling the statistical relationships between the features. A Bayesian network is a kind of probabilistic

graphical model that encodes a factorization of the joint probability distribution of the features (also called variables) in a given domain. Bayesian networks compactly represent the problem domain and can perform any kind of reasoning (causal, diagnostic, abductive, bidirectional, etc.) efficiently because of the local computations allowed by the probability factorization.

Formally, a Bayesian network can be defined as a pair $B = < G(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{P} >$ with two main components:

- The graphical part $G(\boldsymbol{X}, \boldsymbol{A})$ is a directed acyclic graph (DAG) used to capture the structure of the problem. The set of nodes ($\boldsymbol{X}$) represents the variables, $\boldsymbol{X} = (X_1,\ldots, X_n)$, included in the problem domain. The set $\boldsymbol{A}$ contains the directed edges (called arcs) connecting the nodes. In a DAG, the set of arcs cannot include a directed cycle. The probabilistic conditional (in)dependence relationships between the variables in the domain are codified in the set of arcs ($\boldsymbol{A}$).
- The probabilistic component $\boldsymbol{P}$ includes the conditional probability distributions $P(X_i \mid \boldsymbol{Pa}(X_i))$ associated with the variables $X_i$, $i=1,\ldots,n$. For each variable $X_i$, we define the set of its parents as the set of variables with an arc going to $X_i$: $\boldsymbol{Pa}(X_i) = \{Y \in \boldsymbol{X} \mid (Y, X_i) \in \boldsymbol{A}\}$.

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

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

Here, each feature in the experiment was modeled as a discrete variable in the Bayesian network, i.e., each Bayesian network contained six nodes. In the variables representing F1 to F5, we included one discrete value named “Missing”. This value models the scenarios where a category was not provided, either because Uncharacterized was selected, or because Translaminar and Displaced were not selected (for Feature 4).

We trained the Bayesian networks from the data using the GeNIe free modeling environment[1]. The greedy thick thinning algorithm (Dash and Cooper, 2004) with K2 scoring function (Cooper and Herskovits, 1992) was used to train the Bayesian network structure. K2 score function measures the joint probability of the Bayesian network $G$ and a dataset $D$:

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

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

The greedy thick thinning algorithm starts with an empty graph and iteratively adds the arc (without creating a cycle) that yields the maximum increase in the marginal likelihood. When no increment is possible, the algorithm iteratively removes arcs until no arc deletion yields a positive increase in the marginal likelihood. Then, the algorithm stops and returns the resulting Bayesian network structure. We did not allow any feature to be a parent of the variable corresponding to Feature 6. This restriction encodes the knowledge that categorizing a neuron as Uncharacterized disabled the rest of the features for categorization. Once the network structure was found, the maximum likelihood estimators of the parameters in the conditional probability tables of each node were computed from the counts in the data.

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

We analyzed the graphical structures and made inferences with the Bayesian networks to compare the underlying reasoning of different experts. We used the Bayesian networks to study differences in the experts' behavior. A preliminary analysis of the structures of the Bayesian networks consisted of counting the number of network structures where a given edge (arc connecting two nodes without considering its direction) appeared. Very frequent edges highlight common relationships and properties in a large number of experts.

Also, GeNIe software was used to perform inference on the Bayesian networks. We set some categories as selection of choice ("evidence") and used the exact inference algorithm based on message passing (Lauritzen and Spiegelhalter, 1988; Jensen et al., 1990) to update the probabilities of the variables in the Bayesian networks. We then analyzed different scenarios for a subset of Bayesian networks with either the same or different network structures. Similarities and differences between the experts' reasoning were identified by comparing the updated probabilities in the Bayesian networks.

**Supervised classification of neurons**

In order to build models that automatically identify each of the features studied in this work based on a set of quantitative morphological parameters, we selected the 241 neurons whose 3D reconstructions were available at NeuroMorpho.Org. The MicroBrightField Neurolucida package was used to perform the branched structure, convex hull, Sholl, fractal, fan-in diagram, vertex, and branch angle analyses. These analyses were conducted for the complete neuronal morphology as well as separately for the dendritic and axonal arbors. These analyses yielded a set of 2,886 morphological measures of each neuron, including:

- General information about the dendrites and the axons, e.g., the number of endings, the number of nodes (branching points), the total length and the mean length of each dendritic arbor.
- Morphometric measures of the soma such as the area, aspect ratio, compactness, convexity, contour size (maximum and minimum feret), form factor, perimeter, roundness and solidity.
- The total, mean, median and standard deviation of the length of the segments belonging to dendritic arbors and axons independently. Also, we performed these analyses dividing the segments by their centrifugal order from the soma.
- Number of nodes and segments of the complete dendrites and axons, and number of nodes and segments measured by centrifugal order.
- Convex hull analysis. We performed 2D and 3D convex hull analysis of the dendrites and the axon independently to obtain measures of the area, perimeter, volume and surface of the neuronal morphology.
- Sholl analysis. We computed the number of intersections in concentric spheres centered at the soma with increasing radii of 20 μm. We also used the number of endings, nodes and the total length of the segments included in those spheres.
- Fractal analysis. We computed the fractal dimension for the dendrites and the axon independently using the box-counting method (Mandelbrot, 1982). The fractal dimension is a quantity that indicates how completely the neuron fills space.
- Vertex analysis of the connectivity of the nodes in the branches to describe the topological and metric properties of the arbors. We used the number of nodes of each one of the three types: Va (branching points where the two child segments end), Vb (branching points where one of the child segments end) and Vc (branching points where the two child segments bifurcate). We also used the ratio Va/Vb and computed the number of nodes of each type by centrifugal order.
- Branch angle analysis. We used planar, local and spline angles that measure the direction of the branches at different levels. We computed the mean, standard deviation, and median of the three angles for dendrites and axon individually. Additionally, we computed the mean, standard deviation, and median of the angles of the segments grouped by centrifugal order.

Many variables were measured according to the centrifugal order of the segments they belonged to. Since neurons have different maximum centrifugal order, length, etc., each neuron had a different number of computable variables. For example, one neuron might have dendrites with a maximum centrifugal order of 9 and another neuron could have dendrites with a maximum centrifugal order of 5. Variables that measured neuron morphology at orders 6, 7, 8 and 9 were not computable in the second neuron, so we set those values to 0 to be manageable by the algorithms. Variables concerning the complete neuron morphology are not affected by this issue, since they were obtained from the data directly coming from the 3D reconstructions. For each one of the features in the experiment, we had to assign a single "true category" to each neuron. We used the most frequently occurring value in the 42 assignments made by the experts who completed the experiment, i.e., we applied a simple majority vote to assign a "true category" to each neuron for each feature. Using this approach, there were no neurons categorized as Arcade, Cajal-Retzius or Other by the majority of the experts.

The accuracy of the classifiers was estimated using the leave-one-out technique (Mosteller and Tukey, 1968). The following 10 different classification algorithms available in Weka software were applied using their default parameters (Witten and Frank, 2005).

- NB: Naïve Bayes classifier, where the conditional distributions of the continuous variables given the class values are modeled using Gaussian distributions (Pérez et al., 2006).
- NBdis: Discrete naïve Bayes classifier (Minsky, 1961). The continuous variables are discretized using a supervised discretization technique (Fayyad and Irani, 1993).
- RBFN: Neural network for classification tasks with one single hidden layer that uses Gaussian radial basis functions as activation functions (Bishop, 1995).
- SMO: Support vector machine with polynomial kernels implementing the sequential minimal optimization algorithm (Platt, 1998; Keerthi et al., 2001).
- IB1: Nearest neighbor classifier (Aha et al., 1991).
- IB3: Nearest neighbor classifier using 3 neighbors.
- JRip: Rule induction technique using RIPPER algorithm (Cohen, 1995).
- J48: Classification tree using C4.5 algorithm (Quinlan, 1993).
- RForest: Classification technique using a set of random tree classifiers (Breiman, 2001).
- RTree: Classification tree that chooses the variables at each node randomly.

Additionally, two variable selection methods were studied:

- Gain Ratio: A univariate filter algorithm that ranks the predictive variables according to their Gain Ratio with the class label and keeps the best 500 variables.
- CfsSubsetEvaluation: This algorithm tries to find a subset of predictive variables that is highly correlated with the class, but has low intercorrelation between the predictive variables. It starts with an empty subset and iteratively adds the variable that yields a subset with the highest correlation value. The correlation measures the symmetric uncertainty of each variable in the subset with the class (to maximize), and adjusts it to take into account the symmetric uncertainty between the predictive variables (to minimize). The symmetric uncertainty is a measure of correlation based on the marginal entropies and the joint entropies between pairs of variables (Hall, 1999).

These classification algorithms were applied in three different settings:

- **Classifiers for each feature independently:** Each one of the features in the experiment was considered independently. The number of class values was the same as the number of categories in the features, i.e., two class values for Feature 1, Feature 2, Feature 3, and Feature 6; and three class values for Feature 4. There were no neurons classified as Arcade, Cajal-Retzius or Other, so the classifiers for Feature 5 had 7 class values.
- **Binary classifiers for each category in Feature 5:** We induced a binary classifier (with two class values) to identify each category in Feature 5 versus all the other categories merged together. Neurons classified as Chandelier (3 neurons) or Neurogliaform (4 neurons) were very rare. Therefore, we did not induce binary classifiers for these two categories, because the class values would be too unbalanced for the classifiers to find the characterizing properties of these interneuron types.

- **Classifiers merging interneuron types:** Following the agreement results observed in the previous analyses, we decided to check whether the classification algorithms performed better when interneuron types that are difficult to distinguish were merged into one category. Therefore, we trained classifiers after having merged the categories corresponding to Common type, Common basket and Large basket into a single category, as these three interneuron types were frequently confused with each other. The rest of the categories were considered individually.

We performed an exact binomial test to test the hypothesis that the number of correctly classified neurons is greater than that expected with a base classifier always assigning the class with maximum prior probability. To estimate the number of correctly classified neurons, we multiplied the accuracy reported by the leave-one-out technique by 241. The null hypothesis is that the number of correctly classified neurons matches 241 times the maximum prior probability. The alternative hypothesis is that the number of correctly classified neurons is higher than 241 times the maximum prior probability. Statistical significance was established when the p-values were smaller than the significance level $\alpha = 0.05$.

**References**


Aha, D.W., Kibler, D., Albert, M.K. (1991) Instance-based learning algorithms. Machine Learning 6: 37—66.

Artstein, R. and Poesio, M. (2008) Inter-coder agreement for computational linguistics. Computational Linguistics 34(4): 555—596.

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

Bishop, C.M. (1995) Neural Networks for Pattern Recognition. New York: Oxford University Press.

Breiman, L. (2001) Random forests. Machine Learning 45(1): 5—32.

Byrt, T., Bishop, J., Carlin, J.B. (1993) Bias, prevalence and kappa. Journal of Clinical Epidemiology 46(5):423—429.

Carletta, J. (1996) Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics 22(2): 249—254.

Cicchetti, D.V. and Feinstein, A.R. (1990) High agreement but low kappa: II. Resolving the paradoxes. Journal of Clinical Epidemiology 43(6): 551—558.

Cohen, J. (1960) A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1): 37—46.

Cohen, W.W. (1995) Fast effective rule induction. In: Prieditis, A., Russell, S.J. (Eds.) Proceedings of the Twelfth International Conference on Machine Learning, pp. 115—123. Morgan Kaufmann.

Cooper, G.F. and Herskovits, E. (1992) A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9: 309—347.

Dash, D. and Cooper, G.F. (2004) Model averaging for prediction with discrete Bayesian networks. Journal of Machine Learning Research 5: 1177—1203.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). Journal of the Royal Statistical Society. Series B (Methodological) 39: 1—38.

Dunn, G. (1989) Design and Analysis of Reliability Studies: The Statistical Evaluation of Measurement Errors. Edward Arnold.

Fayyad, U.M. and Irani, K.B. (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: Bajcsy, R. (Ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022—1027. Morgan Kaufmann.

Feinstein, A.R. and Cicchetti, D.V. (1990) High agreement but low kappa: I. The problems of two paradoxes. Journal of Clinical Epidemiology 43(6): 543—549.

Fleiss, J.L. (1971) Measuring nominal scale agreement among many experts. Psychological Bulletin 76(5): 378—382.

Hall, M. (1999) Correlation-based Feature Selection for Machine Learning. PhD Thesis, University of Waikato.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H. (2009) The WEKA data mining software: An update. SIGKDD Explorations 11(1).

Huang, Z. (1998) Extensions to the *k*-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2:283—304.

Jensen, F.V., Lauritzen, S.L. and Olesen, K.G. (1990) Bayesian updating in causal probabilistic networks by local computations. Computational Statistics Quarterly 4: 269–282.

Keerthi, S.S., Shevade, S.K., Bhattacharyya, C. and Murthy, K.R.K. (2001) Improvements to Platt's SMO algorithm for SVM classifier design. Neural Computation 13(3): 637—649.

Landis, J.R. and Koch, G.G. (1977) The measurement of observer agreement for categorical data. Biometrics 33(1): 159—174.

Lauritzen, S.L. and Spiegelhalter, D.J. (1988) Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, Series B, Statistical Methodology 50(2): 157–224.

MacQueen, J.B. (1967) Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

Mandelbrot, B. (1982) The Fractal Geometry of Nature. Freeman.

Minsky, M. (1961) Steps toward artificial intelligence. Proceedings of the Institute of Radio Engineers 49: 8—30.

Mosteller, F. and Tukey, J.W. (1968) Data analysis, including statistics. In: Lindzey, G. and Aronson, E. (Eds.) Handbook of Social Psychology, volume 2, pp. 80—203. Addison-Wesley.

Pearl, J. (1988) Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann.

Pérez, A., Larrañaga, P. and Inza, I. (2006) Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naïve Bayes. International Journal of Approximate Reasoning 43: 1—25.

Platt, J.C. (1998) Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C. and Smola, A. (Eds.) Advances in Kernel Methods: Support Vector Learning, pp. 185—205. Cambridge: The MIT Press.

Popping, R. (1988) On agreement indices for nominal data. Sociometric Research: Volume 1, Data Collection and Scaling, pp. 90—105. St. Martin's Press.

Quinlan, R. (1993) C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann.

Scott, W.A. (1955) Reliability of content analysis: The case of nominal scale coding. Public Opinion Quarterly 19(3): 321—325.

Sim, J. and Wright, C.C. (2005) The kappa statistic in reliability studies: Use, interpretation, and sample size requirements. Physical Therapy 85(3): 257—268.

Wegman, E.J. (1990) Hyperdimensional data analysis using parallel coordinates. Journal of the American Statistical Association 85(411): 664—675.

Witten, I.H. and Frank, E. (2005) Data Mining. Practical Machine Learning Tools and Techniques, 2nd edition. Morgan Kaufmann.