JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 26, Number 4, 2019

Pp. 322–335
DOI: 10.1089/cmb.2018.0155

Dendritic Spines Taxonomy: The Functional and Structural Classification • Time-Dependent Probabilistic Model of Neuronal Activation

PAULINA URBAN,[1,2] VAHID REZAEI,[3,4] GRZEGORZ BOKOTA,[1,5] MICHAŁ DENKIEWICZ,[1,2] SUBHADIP BASU,[6] and DARIUSZ PLEWCZYŃSKI[1,7]

ABSTRACT

Categorizing spines into four subpopulations, stubby, mushroom, thin, or filopodia, is one of the common approaches in morphological analysis. Most cellular models describing synaptic plasticity, long-term potentiation (LTP), and long-term depression associate synaptic strength with either spine enlargement or spine shrinkage. Unfortunately, although we have a lot of available software with automatic spine segmentation and feature extraction methods, at present none of them allows for automatic and unbiased distinction between dendritic spine subpopulations, or for the detailed computational models of spine behavior. Therefore, we propose structural classification based on two different mathematical approaches: unsupervised construction of spine shape taxonomy based on arbitrary features (SpineTool) and supervised classification exploiting convolution kernels theory (2dSpAn). We compared two populations of spines in a form of static and dynamic data sets gathered at three time points. The dynamic data contain two sets of spines: the active set and the control set. The first population was stimulated with LTP, and the other population in its resting state was used as a control population. We propose one equation describing the distribution of variables that best fits all dendritic spine parameters.

Keywords: classes of spines, dendritic spines, hidden Markov model, image analyze.

1. INTRODUCTION

Dendritic spines are small membranous protrusional structures located on the surface of neuronal dendrites. They play a crucial role in learning and memory processes, since this is where excitatory synapses are located. The functional and structural reorganization of synapses depends on the brain plasticity, which is important in learning and memory processes (Alvarez and Sabatini, 2007). One of the key

[1]Center of New Technologies, University of Warsaw, Warsaw, Poland.
[2]College of Inter-Faculty Individual Studies in Mathematics and Natural Sciences, University of Warsaw, Warsaw, Poland.
[3]Department of Statistics, Faculty of Mathematics and Computer Sciences, Allameh Tabataba'i University, Tehran, Iran.
[4]School of Biological Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.
[5]Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland.
[6]Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.
[7]Faculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland.

morphological attributes of the dendritic spines is their structural variability, which depends on their length and width.

The structure of a spine is as follows: the spine is connected to the dendritic shaft with a thin neck. The spine head volume correlates positively with the postsynaptic density area, the presynaptic active zone area, the number of AMPA-type glutamate receptors (AMPA is α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid receptor), and the synaptic strength. Moreover, the dendritic spine shape determines the strength of the synaptic connections (Kasai et al., 2003; Alvarez and Sabatini, 2007; Fortin et al., 2012; Gu et al., 2014; Miermans et al., 2017; Segal, 2017).

All these processes constitute the basis for coding and memory storage information in the brain (Segal, 2010, 2017). Many cognitive disorders, for example, autism (Pathania et al., 2014) or fragile-X syndrome (Cruz-Martín et al., 2010; He and Portera-Cailliau, 2013), are connected to alteration in the shapes of dendritic spines, their size, or number of receptors (Penzes et al., 2011).

Depending on their activity, the morphology of spines is changing, which is the ground for their structural plasticity involving formation or loss of synapses. Two major processes that are observed in structural plasticity are long-term potentiation (LTP) and long-term depression (LTD), which are associated with the increase or decrease of synaptic strength (Martin et al., 2000; Segal, 2010; O'Donnell et al., 2011; Fortin et al., 2012; Colgan and Yasuda, 2014; Meyer et al., 2014; Stawarski et al., 2014). Hence, understanding the changes in the shape of dendritic spines is of great importance.

The spines are commonly divided, based on their shape, into the following subpopulations: stubby, thin, filopodia, or mushroom (Von Bohlen Und Halbach, 2009; Ruszczycki et al., 2012; Ebrahimi and Okabe, 2014; Gu et al., 2014; Qiao et al., 2016). However, no appropriate methods enabling the users for an automatic classification of dendritic spines into subpopulations have been developed so far. We provide methods that will show the differences between these subpopulations of dendritic spines in their shapes and transitions in time. Like in the previous studies of Bokota et al. (2016), we potentiated the synapses using LTP stimulation that produces a long-lasting increase in excitatory postsynaptic potential (EPSP).

In the rest of the article, we refer to a population of spines stimulated by LTP as dynamic data, and the unstimulated spines are denoted as static data.

In Section 2, we describe how the data were collected. We also describe the pieces of software that help us classify the spines into one of four classes. Next we check the basic characteristics of the classes of spines from the static and dynamic populations. Afterward, each of the classes is divided into one of three subpopulations: growing, not changing, and shrinking spines.

In Section 3, we provide a model for the shape transitions. We propose clustering analysis for checking the taxonomy development. Next, we introduce a hidden Markov model (HMM) for spines transitions in time. We also propose an equation that fits a distribution of all of the shape parameters.

In Section 4, we present our results and in Section 5 we conclude our study.

2. DATA PREPARATION AND ANALYSIS

In this section, we describe the statistical analysis of mammalian dendritic cell populations: static and dynamic populations. For the dynamic data, preprocessing was necessary, performed by choosing spines from both active and control groups.

For details about data preparation, see Bokota et al. (2016), section ''Data Preparation and Analysis.'' In this article we describe only the differences in chemical substances that were used in experiments in active and control groups for dynamic data. In the active group, the chemical LTP was induced through the application of bath in a mixture of 50 μM forskolin, 50 μM picrotoxin, and 0.1 μM rolipram (each dissolved in DMSO) in a maintenance medium. The control group received a compound-free solvent [dimethyl sulfoxide (DMSO)].

Our next step was to obtain the numerical features of the spines. There are many spine extraction methods, but they appear to be ineffective in our approach, so we decided to use custom software described in Ruszczycki et al. (2012). Consequently, dendritic spine features (denoted as DESCRIPTORS) were chosen on the basis of length, the head width (hw), max width location (mwl), max width (mw), neck width (nw), foot, circumference, area, width-to-length ratio (wlr), length-to-width ratio (lwr), and length-to-area ratio (lar). The detailed description of the software can be found in Ruszczycki et al. (2012).

After preprocessing the samples, we received two groups of spines, that is, the static group consisting of 1751 samples and the dynamic group consisting of two subgroups: the control group (333 samples) and the

active group (448 samples). For each spine, all of the previously described 11 features were measured. For the dynamic data, all the features were measured at three different timestamps: t_0 (the time before stimulation), t_{10} (10 minutes after the stimulation, which was called the short-term activation), and t_{40} (40 minutes after the stimulation, which was called the long-term activation). We know from other studies that we can already observe changes in the spine structure 10 minutes after the stimulation (Stawarski et al., 2014; Szepesi et al., 2014). Consequently, by ''dynamic,'' we denote all the features at all timestamps and, by ''dynamic x,'' we denote all spines from the dynamic data set described only by features at time from both groups.

2.1. Balanced subset selection

In Table 1, we report mean values and standard deviations for descriptors from static data. In Figure 1, we show an example analyzed image. For the dynamic group given in Table 2, we report mean values and standard deviations of control and active groups at a given time. We calculate p values using two-tailed t-tests for the difference of mean values between the control and the active groups.

We report significant differences for almost all descriptors (p value is above the threshold value $p > 0.001$ only in case of three features). Such remarkable differences between both sets may influence the statistical analysis of their behavior. Therefore, we decided to preprocess the data sets by excluding some spines. We drew a number of pairs of closest spines, each pair consisting of a spine from the active group and a spine from the control group. The measure of how close the spines are is based on the normalized Euclidean distance between the vectors of features at a timestamp. We chose 300 such pairs, and still we observed significant differences for almost all descriptors. Once again we chose pairs of spines, but this time we checked the effect on 50, 25, and 15 pairs of spines. For 50 pairs of spines, we report significant differences ($p > 0.001$) in case of four parameters. For 15 and 25 pairs of spines, we did not report any significant differences ($p > 0.001$). In 15 pairs of spines, we report significant difference for $p < 0.05$ in case of three parameters (length, area, and circumference). In 25 pairs of spines, we report significant differences for $p < 0.05$ in case of four parameters (length, width–length ratio, area, and circumference).

Such differences (between non-normalized data and normalized data, and for number of pairs of spines) can be explained by the fact that these cultures were obtained from different animals (which means differences in in vitro cultures).

In Table 3, we report new statistic values for the differences between the samples after choosing 15 closest pairs. The same statistical test that was performed before was used here as well. In the results, we use the data without prior Euclidean normalization.

3. METHODS

In this section, we present three tools used to describe the parameters of dendritic spines, the Principal Component Analysis (PCA) method, and two clustering methods, to construct the spine shape taxonomy in

Table 1. Mean Values and Standard Deviations of Static Data

Parameter	*Mean*	*STD*
Head width	0.819	0.246
Foot	0.484	0.247
Max width location	0.407	0.249
Max width	0.855	0.253
Width–length ratio	0.549	0.246
Length–width ratio	2.199	1.016
Length–area ratio	2.039	0.681
Neck width	0.312	0.249
Length	1.695	0.663
Circumference	5.344	1.864
Area	0.897	0.420

STD, standard deviations.

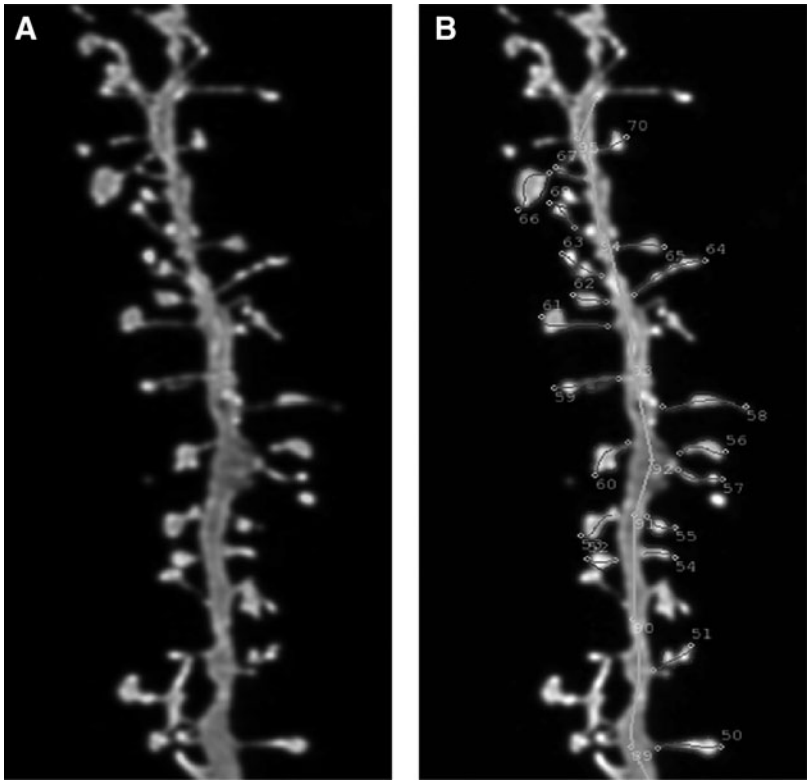

FIG. 1. Analyzed data. Example **(A)** of the image with the markered dendritic spines and examples **(B)** of spines classification.

an unsupervised way. Subsequently, we propose the probabilistic model of transitions using HMM. We also present the results of fitting the best distribution to the data. All the analyses were conducted in Python 2.7 environment and in R version 3.4.

3.1. Classification of spines

Three different pieces of software were used to derive geometrical parameters and classification of spines. The first piece of software was Spine Magic, which enabled us to get the spines' parameters (Ruszczycki et al., 2012). The second piece of software was 2dSpAn (Basu et al., 2016), which, in turn, allowed for classification of the spines into one of the four classes. The third piece of software was SpineTools, which finally allowed for classification of the spines into one of three classes, combining ''thin'' and ''filipodia'' into ''long'' (Jasińska et al., 2016).

Table 2. Differences Between the Active and Control Groups at Time

Parameter	*Mean control group*	*STD control group*	*Mean active group*	*STD active group*	p
Head width	0.756	0.411	0.684	0.311	0.0054
Foot	1.089	0.557	0.766	0.363	0.0000
Max width location	0.659	0.264	0.554	0.289	0.0000
Max width	0.988	0.454	0.789	0.364	0.0000
Width–length ratio	0.490	0.298	0.662	0.409	0.0000
Length–width ratio	3.145	2.648	2.211	1.700	0.0000
Length–area ratio	2.083	1.046	2.172	0.985	0.2243
Neck width	0.459	0.347	0.412	0.284	0.0416
Length	1.893	1.150	1.293	0.732	0.0000
Circumference	6.712	3.229	4.669	2.091	0.0000
Area	1.035	0.587	0.681	0.436	0.0000

Table 3. Differences Between Active[15] and Control[15] Groups at Time t_0

Parameter	*Mean control group*	*STD control group*	*Mean active group*	*STD active group*	p
Head width	0.639	0.176	0.641	0.152	0.981
Foot	0.763	0.452	0.605	0.276	0.321
Max width location	0.553	0.216	0.511	0.228	0.648
Max width	0.740	0.215	0.660	0.157	0.320
Width–length ratio	0.476	0.199	0.604	0.240	0.167
Length–width ratio	2.563	1.266	1.927	0.782	0.160
Length–area ratio	2.030	0.712	2.227	0.635	0.483
Neck width	0.346	0.215	0.347	0.172	0.998
length	1.497	0.450	1.144	0.262	0.03
Circumference	5.344	1.138	3.706	0.646	0.0002
Area	0.788	0.236	0.546	0.169	0.009

3.2. Simplification of shape representations

The PCA (for details see Jolliffe, 2002) is used when the reduction of the data dimensionality is required. We applied PCA for spines from static data, and for both populations (control and active) used for the dynamic data, in all three timestamps.

3.3. Hidden Markov model

The HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e., hidden) states. In the HMM, the state is not directly visible, but the output, dependent on the state, is visible. The parameters of a HMM are of two types: transition and emission probabilities. Under a HMM, two assumptions are made for the transition and the emission probabilities:

1. The tth hidden state, given the $(t-1)$th hidden state, is independent of previous state.
2. The tth observation depends only on the tth state.

The HMM is specified as a triplet $\lambda=(A, B, \pi)$, where A is the transition probability matrix, B is the emission probability matrix, and π is the vector of initial probabilities. The parameters of a HMM can be estimated by using Baum–Welch algorithm. The Baum–Welch algorithm works by guessing the initial parameter values, then estimating the likelihood of the observation under the current parameters. This likelihood will then be used to re-estimate the parameters iteratively until a local maximum is reached. The Baum–Welch algorithm finds $\max_{\lambda} P(\text{observation}|\lambda)$ (Rabiner, 1989; Rabiner and Juang, 1993; Foruzan et al., 2013; Ghavidel et al., 2015).

3.4. Distributions

All the available probability distributions implemented in Python (for details see Python documentation for Scipy stats library) were fit for the static and dynamic data. Distributions were fit for each parameter separately, for each class, and for the data without classification. As an error measure, we used the error sum of squares (SSE). Distributions with smaller SSE were considered better than the others. Next, we generated data for each of the distribution with estimated parameters and created boxplots with our experimental data (Fig. 2). Our measure of fitness criterion is defined as the smallest difference between the medians obtained from our data and from the generated data.

3.5. Clusters

Clustering allows us to assign similar objects in terms of their geometrical properties (in our study it will be spines) to groups. Clusters represent the possible shapes of spines (the spines in a cluster have greater similarity in shapes among themselves than the spines outside the given cluster). We consider two main types of clustering: fuzzy and crisp. In crisp clustering, each spine is classified into exactly one class (shape); in fuzzy clustering, a spine is assigned a set of scores (ranging from 0 to 1, one for each class), which describe its ''similarity'' to each of the classes.

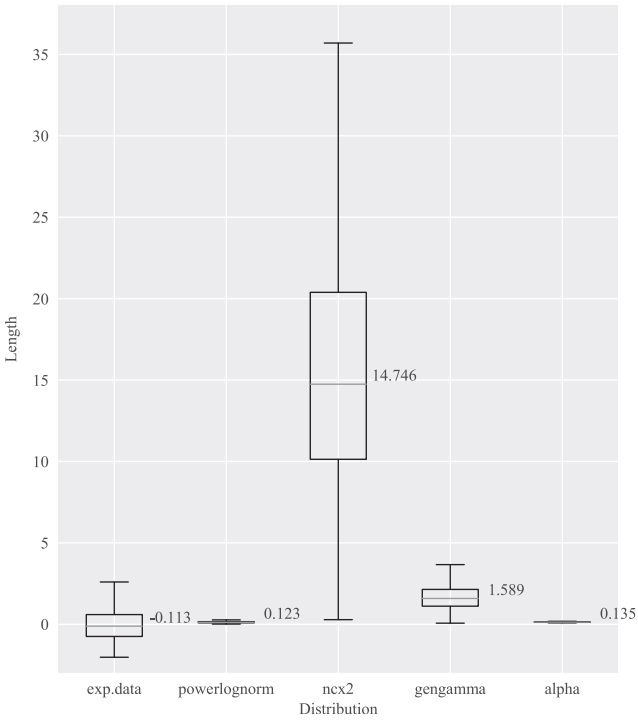

FIG. 2. Boxplot. Example of boxplot for static data for length parameter without classification.

3.6. Transition matrix

After classification of spines into one of the four (or three) classes, we could observe what happens with a particular spine over time (whether it changes its class assignment, as well as whether the parameters change). We calculated the change of spine classification between two timestamps as a transition matrix.

By using this easy method, we can calculate the transition matrix between timestamps: t_0 and t_{10}, t_0 and t_{40}, and t_{10} and t_{40}.

4. RESULTS

4.1. Classification

The 2dSpAn software classified spines into one of the four spine types: stubby, filipodia, mushroom, and thin. This classification procedure is based on mathematical equations called convolution kernels (for details see Basu et al., 2016). SpineTools classified the spines into one of the three spine classes: long, mushroom, and stubby. This software uses hierarchical clustering as a method to split spines into one of the three classes. For static data, 2dSpAn classified 2005 spines, after which we noticed some errors, and some spines were rejected. After removing errors, 1751 spines remained (261 spines were rejected, which is c.a. −13%). Figure 3 shows the percentage of each class for control group from dynamic data in all three timestamps. Among these data sets no spines were rejected, since we did not notice any errors. We observed that stubby class was classified by both 2dSpAn and SpineTools on the same level. However, we saw the differences in the remaining classes: long and mushroom types. We observed that in case of 2dSpAn, the long class dominated, and in case of SpineTools the mushroom class dominated. Such a difference is very interesting, it can be explained by different mathematical equations and the theories on the basis of which each of the pieces of software was built. In Basu et al.'s (2016) method, depending on the seed pixel and binarization threshold definition, the same spines can be classified into different classes. The major problem is that the differences between mushroom and long classes are very small and hard to describe through a single definition.

In Figure 4 we present the results of classification for the static data, with the use of two distinct pieces of software, which make use of different mathematical equations (Ruszczycki et al., 2012). We have chosen to use the software that was used in many different projects and various data sets, as to ensure the quality and comparability of the results.

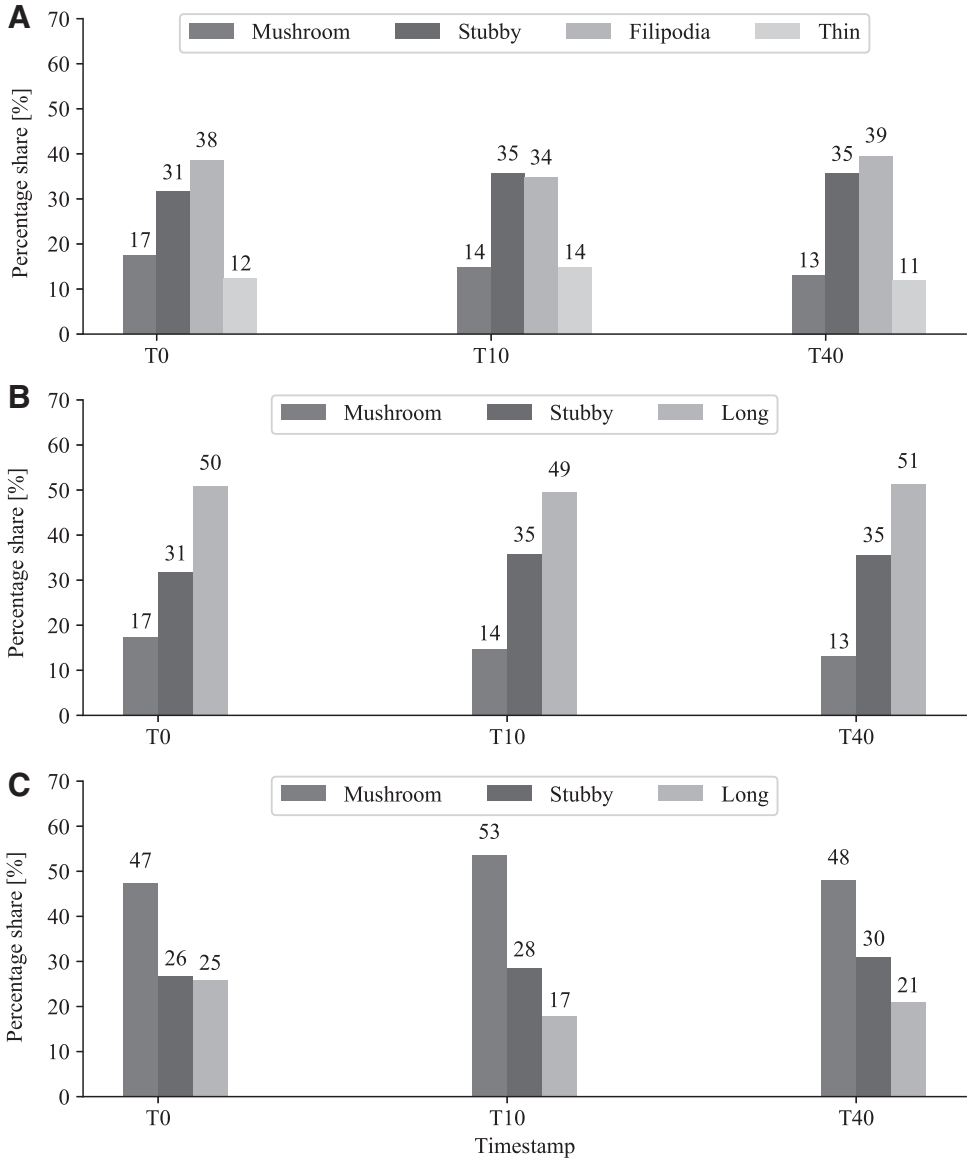

FIG. 3. Classification of dynamic data, the control group. **(A)** Percentage of different spines' classes using 2dSpAn software, **(B)** percentage of different spines' classes using 2dSpAn software, with three classes as in SpineTool software, **(C)** percentage of different spines' classes in control group using SpineTool software.

4.2. Results for principal component analysis (shape representation)

The dendritic spines parameters can be directly divided into three sets: DESCRIPTORS SIZE = length, circumference, area (size-related features) and DESCRIPTORS CONTOUR = hw, foot, mwl, mw, wlr, lwr, lar, and nw (contour-related features) and DESCRIPTORS ALL with all parameters. For the first two features (components) in the reduced representation for DESCRIPTORS ALL, we cover about 70% of the variance in the static data. For DESCRIPTORS CONTOUR, we cover $\sim$66% and for the DESCRIPTORS SIZE we cover $\sim$96% of the variance. For the dynamic data (control group as top values and active group as bottom values) given in Table 4, we show the percentage of the variance for each timestamp for each DESCRIPTOR. We can observe that there is no difference between the static data and the dynamic data (both groups), whereas between the control and the active groups, small differences can be observed. These results were somewhat surprising, because we had suspected larger differences between the control and the active groups.

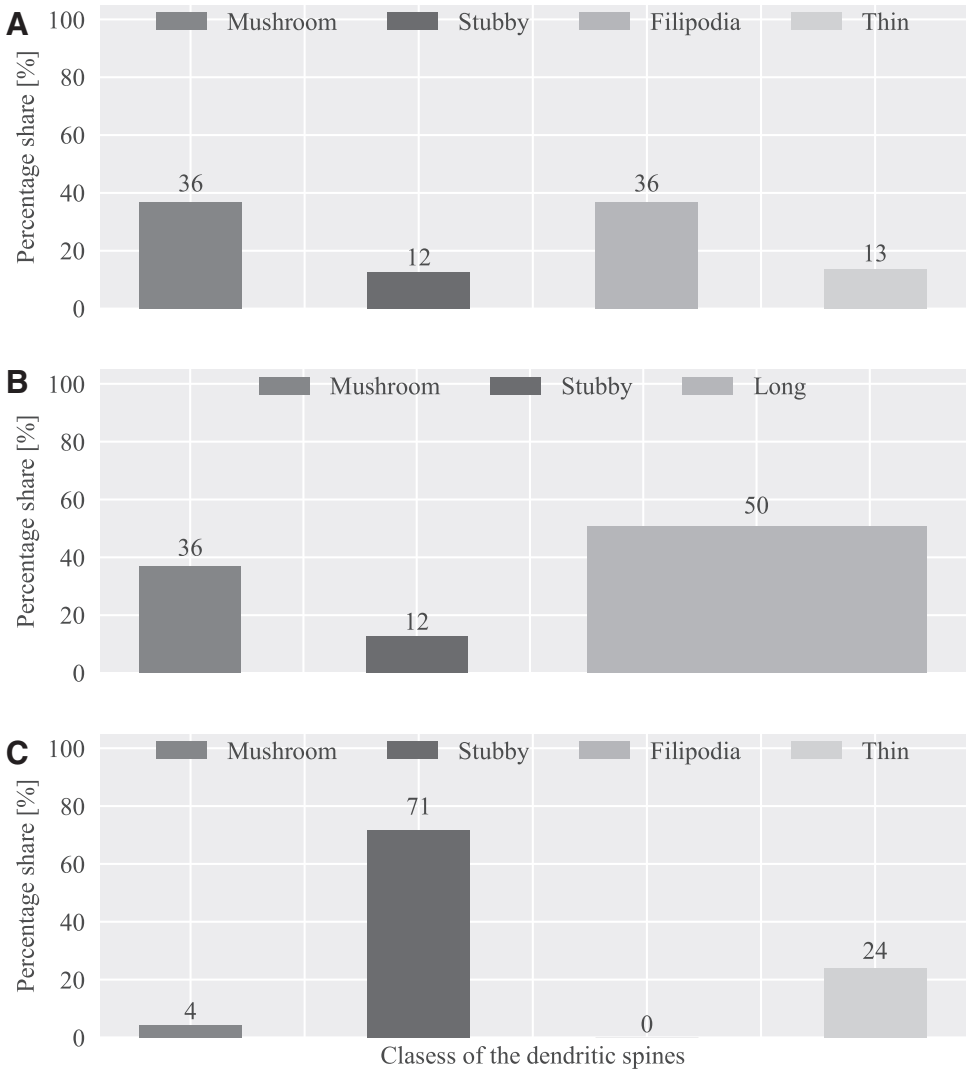

FIG. 4. Classification of static data. **(A)** Percentage of different spines' classes using 2dSpAn software, **(B)** percent of different spines' classes using 2dSpAn software, with three classes as in SpineTool software, **(C)** percent of different spines' classes using equations from Ruszczycki et al. (2012).

4.3. Results for hidden Markov model

To obtain fine results in HMM case, we used transition probability calculated by checking in which class the spine is compared with the timestamp. For the vector with starting probabilities of the spine classes, we used the same probability threshold: 0.25. After using the parameters described earlier in the text together with the sequence of the classes, Baum–Welch algorithm was used, in which we could observe the emission probability that describes the probability between observed states, which is very similar to the calculated

Table 4. Percentage of the Variance for Each Descriptor for Each Timestamp

	Time step 0 (%)	*Time step 10 (%)*	*Time step 40 (%)*
Descriptors all	68	67	71
	75	70	73
Descriptors contour	68	68	72
	78	68	74
Descriptors size	96	95	94
	95	95	95

probability, as described earlier in the text. In Figure 5, two transition and one emission probabilities are shown.

The results from only one neuron are not as good as from all neurons altogether. By ''not good'' we understand three problems: first, we can have a situation in which we have the probability of transition from one class to another, which equals 0; second, in case we have a small number of spines, the emission or transition probability can be very small (<0.0001), which is impossible. The last problem is the oscillations of ''difference''; if more data are available, then we have none to small oscillations. To build probabilistic model, we should acquire the data with more timestamps. We can use the equations in the same manner as in Barrett et al. (2009), in which they use six master equations describing the probability of being a synapse in one of six states. In our situation, we can have five master equations that will describe the probability of being a spine in one of five classes (stubby, thin, filipodia, mushroom, and not existing).

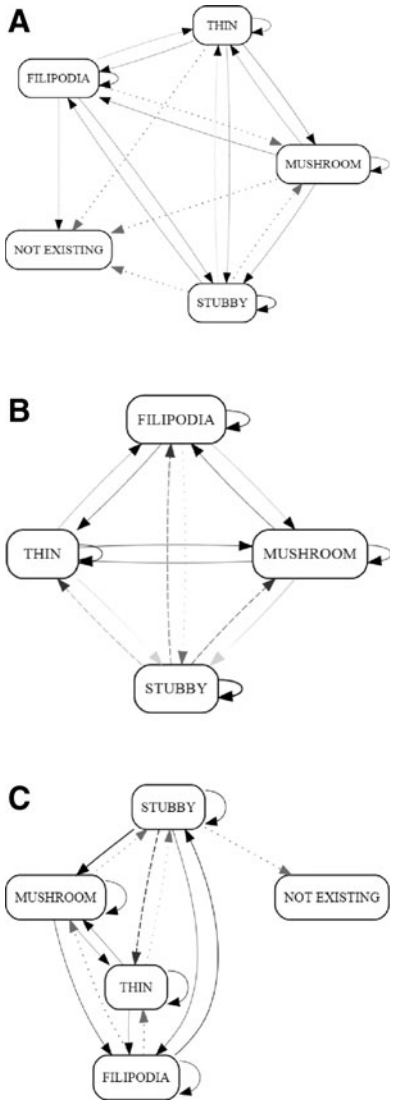

FIG. 5. Transition and emission probability. **(A)** Transition probability calculated by our method, **(B)** transition probability calculated by Baum–Welch algorithm, **(C)** emission probability calculated by Baum–Welch algorithm. Dotted arrows mean that the probability increases by a factor of 10, dashed arrows mean that the probability increases by a factor of 100, tapered arrows mean that the probability increases by a factor of 1000, and the normal arrow means that the probability was not increased by any factor. A nonexisting class is the class in which at the next timestamp we notice that the spine was absorbed by the neuron.

4.4. Results for distributions

We report alpha distribution that is described by Equation (1):

$$\mathrm{pdf}(x, a) = \frac{1}{x^2 * \varphi(a) * \sqrt{2 * \pi} * \exp^{-\frac{1}{2} * (\frac{a-1}{x^2})}}, \tag{1}$$

where

pdf, probability density function;

x, data from one dendritic spine parameter; and

a, shape parameter

fit best the static and dynamic (both control and active group) data (in most cases it is in the first place, but sometimes it is in the second place). Even if the alpha distribution is in the second place in the fitting protocol, we check whether the equations that describe other distributions are very similar to the alpha distribution. In case they are, we could provide one common Equation (2) that will fit best all dendritic spines' parameters. This equation belongs to the exponential family. Using the Minitab (Minitab, Inc., 2010) software, we fit static and dynamic data (both groups, each parameter separately) according to the exponential distribution. All results were statistically significant, which means that our equation should be sufficient.

$$f(x) = \frac{A(x)}{B(x) * \exp\left(C(x)\right)}, \tag{2}$$

where

x, data from one dendritic spine parameter.

4.5. Results for clustering

Both static and dynamic groups (both sets: active and control, with all three timestamps) were taken as one large set altogether. In Figure 6, we present the results for the fuzzy partition coefficient (fpc), which tells us how smoothly our data are described by a certain model. We simulated 25 times FPC to calculate the mean value for each center. In Table 5, we can see that 11 centers are the best options to describe our data and classify them into 11 groups. In Bokota et al. (2016), the spines are classified into 10 groups (our

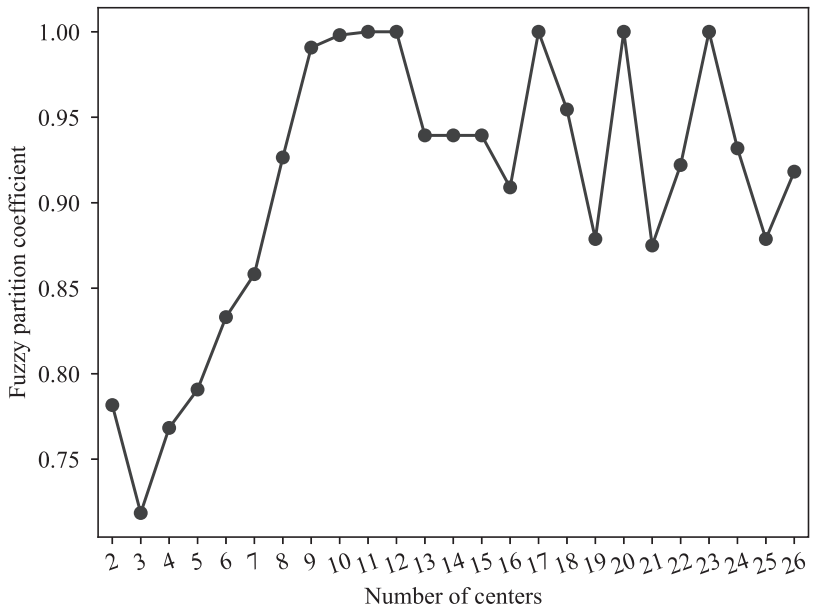

FIG. 6. FPC from fuzzy clustering. This measure is defined within the range 0 to 1. When FPC is maximized, the data are described as the best. FPC, fuzzy partition coefficient.

Table 5. Mean Value for Fuzzy Partition Coefficient Depends on Number of Centers

Number of centers	*Mean value for fuzzy partition coefficient*
1	—
2	0.78
3	0.722
4	0.7604
5	0.7924
6	0.8272
7	0.8888
8	0.9272
9	0.9824
10	0.9964
11	1.0
12	0.98
13	0.9768
14	0.954
15	0.954
16	0.9412
17	0.9356
18	0.9336
19	0.9248
20	0.9332
21	0.9352
22	0.9412
23	0.9308
24	0.9324
25	0.9044
26	0.9104

results are not different from other group results). In Figure 7, the results of hierarchical clustering are presented, calculated for 10, 11, 13, and 16 centers.

5. DISCUSSION AND CONCLUSION

We applied a statistical test and examined a population consisting of 1751 dendritic spines from the static data and 1344 dendritic spines from the active group, and 980 dendritic spines from the control group. We compared the dendritic spine volume and shape changes between two populations at two different states, unstimulated (CONTROL) and LTP-stimulated (ACTIVE), and at three timestamps (with a 0-time, 10-minute, and 40-minute time interval). We show that the preprocessing of data sets and reducing the dendritic spine number were not useful in our study. In this study, we show the differences in dendritic spine classification depending on mathematical equations describing the software that was used to analyze the dendritic spines. We also want to stress how important it is to use the software that was already tested on various sets of data, which appears to be more important than applying the patterns/conditions from other publications. Conditions for data classification used by Ruszczycki et al. (2012) were probably very good, but not universal (our results did not confirm their findings). As has already been said, it is very difficult to work out the definition that could describe a given spine class.

We also propose a probabilistic model based on HMM, connected with the LTP-LTD model presented by Barrett et al. (2009). Of course, everything depends on how we are going to classify particular spines, and what baseline parameters will be attributed to them. However, it is the way to combine experimental data with a mathematical model, which gives hope for discovering a formula or equation to be applied in cases wherein we have to fight various neurodegenerative diseases related to the number of spines.

Furthermore, we introduce a new universal equation (belonging to exponential family) that describes the distributions for each dendritic spine parameter. If parameters A, B, and C from Equation (1) were defined according to the circumstances (LTP, LTD, diseases connected with the number of spines), it would be

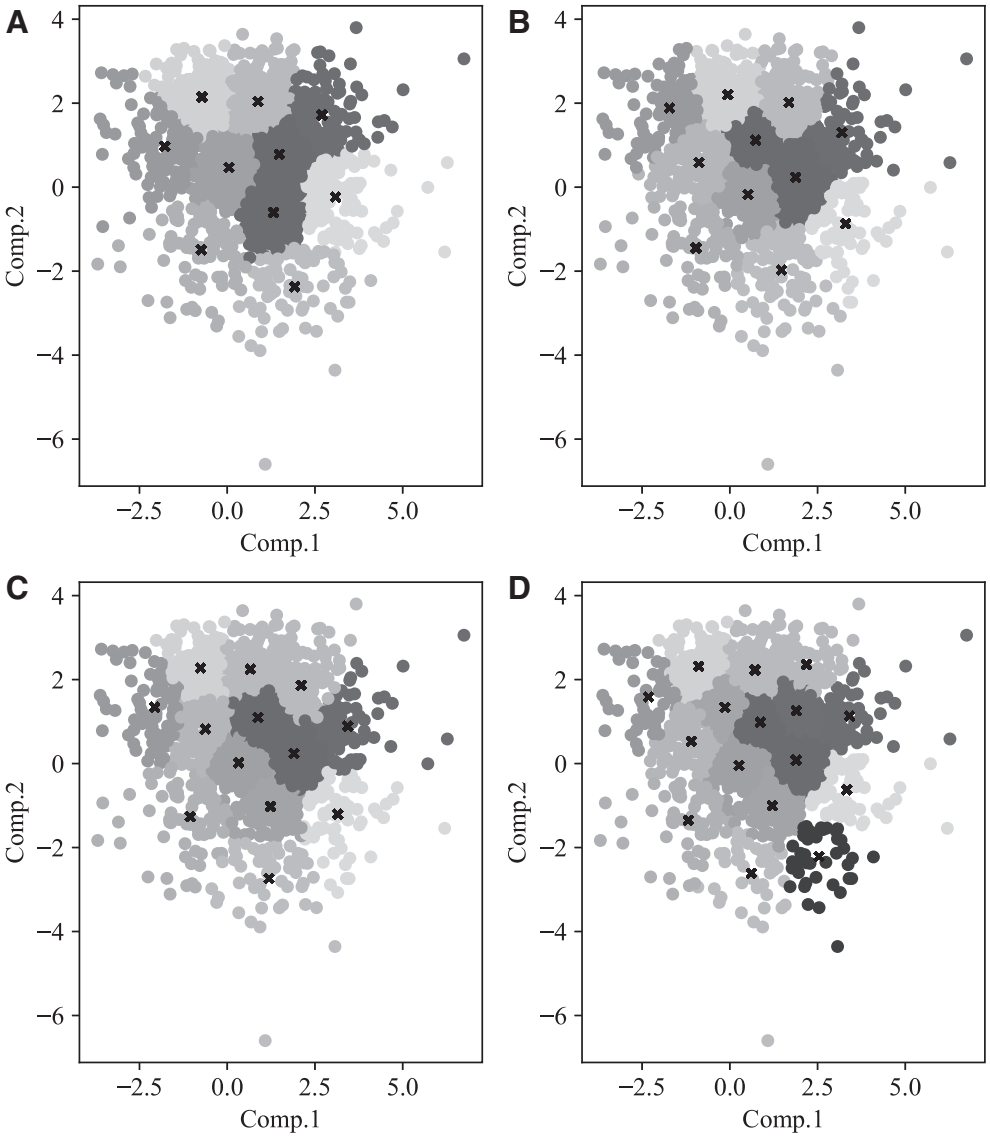

FIG. 7. Distribution of spine clusters obtained by hierarchical clustering. **(A)** 10 classes of clusters, **(B)** 11 classes of clusters, **(C)** 13 classes of clusters, **(D)** 16 classes of clusters. For each subplot, one color and its shades represent the subpopulations of dendritic spines.

easier to create a model that would reflect the state of a particular neuron in relation to the state of spines, and the state of the same neuron after the administration of the medication to stimulate, for example, the growth in number of spines. Of course, such a model could be additionally improved using the information on the molecular level, which would make it easier to find medicines to be used in treatments of the diseases resulting from the number of spines.

We also show how to use fuzzy (fpc) and c-means clustering to have the best number of groups representing each subpopulation of dendritic spines. In our study, we obtained 11 classes, whereas Bokota et al. (2016) obtained 10 classes in their publication, which shows that our procedures are good, yet it has to be noted that depending on the number of data available, we can expect as many as 10 classes (class). The result may be the evidence that still a lot of research is required to be performed to finally determine the number of dendritic spine classes.

As in Bokota et al. (2016), we support the hypothesis that the biological information is not stored in the spine shapes or sizes depending on their classes, but is rather related to the dynamic changes at the spine population level.

ACKNOWLEDGMENT

We thank Michał Pietal for his help in editing the text. **Sources of Funding:** This study was supported by the Polish National Science Center (2014/15/B/ST6/05082), Foundation for Polish Science (TEAM to D.P.), and by the grant from the Department of Science and Technology, India, under Indo- Polish/Polish-Indo project No.: DST/INT/POL/P-36/2016. The study was cosupported by grant 1U54DK107967-01 ''Nucleome Positioning System for Spatiotemporal Genome Organization and Regulation'' within 4DNucleome NIH program. S.B. was funded by Department of Biotechnology grant (BT/PR16356/BID/7/596/2016).

ETHICAL STATEMENT

All experimental procedures were carried out in accordance with the Ethical Committee on Animal Research of the Nencki Institute, based on the Polish Act on Animal Welfare and other national laws that are in full agreement with the EU directive on animal experimentation.

AUTHORS' CONTRIBUTIONS

Paulina Urban and Grzegorz Bokota designed and performed data analysis (also statistical analysis) and visualization. Vahid Rezaei helped in hidden Markov Model and distribution analysis. Michał Denkiewicz helped in hidden Markov Model analysis. Subhadip Basu helped with 2dSpAn software analyzed. Dariusz Plewczyński supervised the project and conducted extensive discussions during our analysis. Paulina Urban, Michał Denkiewicz, and Vahid Rezaei wrote the article. All authors read and approved the final article.

AUTHOR DISCLOSURE STATEMENT

The authors declare that there are no competing financial interests.

REFERENCES

Alvarez, V.A., Sabatini, B.L. 2007. Anatomical and physiological plasticity of dendritic spines. *Annu. Rev. Neurosci.* 30, 79–97.

Barrett, A.B., Billings, G.O., Morris, R.G.M., et al. 2009. State based model of long-term potentiation and synaptic tagging and capture. *PLoS Comput. Biol.* 5, e1000259.

Basu, S., Plewczynski, D., Saha, S., et al. 2016. 2dSpAn: Semiautomated 2-d segmentation, classification and analysis of hippocampal dendritic spine plasticity. *Bioinformatics* 32, 2490–2498.

Bokota, G., Magnowska, M., Kuśmierczyk, T., et al. 2016. Computational approach to dendritic spine taxonomy and shape transition analysis. *Front. Comput. Neurosci.* 10, 140.

Colgan, L.A., Yasuda, R. 2014. Plasticity of dendritic spines: Subcompartmentalization of signaling. *Annu. Rev. Physiol.* 76, 365–385.

Cruz-Martín, A., Crespo, M., Portera-Cailliau, C. 2010. Delayed stabilization of dendritic spines in fragile X mice. *J. Neurosci.* 30, 7793–7803.

Ebrahimi, S., Okabe, S. 2014. Structural dynamics of dendritic spines: Molecular composition, geometry and functional regulation. *Biochim. Biophys. Acta* 1838, 2391–2398.

Fortin, D.A., Srivastava, T., Soderling, T.R. 2012. Structural modulation of dendritic spines during synaptic plasticity, structural modulation of dendritic spines during synaptic plasticity. *Neuroscientist* 18, 326–341.

Foruzan, A.H., Kalantari Khandani, I., Baradaran Shokouhi, S. 2013. Segmentation of brain tissues using a 3-D multi-layer Hidden Markov model. *Comput. Biol. Med.* 43, 121–130.

Ghavidel, F.Z., Claesen, J., Burzykowski, T. 2015. A nonhomogeneous Hidden Markov model for gene mapping based on next-generation sequencing data. *J. Comput. Biol.* 22, 178–188.

Gu, L., Kleiber, S., Schmid, L., et al. 2014. Long-term in vivo imaging of dendritic spines in the hippocampus reveals structural plasticity. *J. Neurosci.* 34, 13948–13953.

He, C.X., Portera-Cailliau, C. 2013. The trouble with spines in fragile X syndrome: Density, maturity and plasticity. *Neuroscience* 251, 120–128.

Jasińska, M., Miłek, J., Cymerman, I.A., et al. 2016. miR-132 regulates dendritic spine structure by direct targeting of matrix metalloproteinase 9 mRNA. *Mol. Neurobiol.* 53, 4701–4712.

Jolliffe, I.T. 2002. Principal component analysis and factor analysis, 150–166. *In Principal Component Analysis*. Springer Verlag, New York, NY.

Kasai, H., Matsuzaki, M., Noguchi, J., et al. 2003. Structure-stability-function relationships of dendritic spines. *Trends Neurosci.* 26, 360–368.

Martin, S.J., Grimwood, P.D., Morris, R.G. 2000. Synaptic plasticity and memory: An evaluation of the hypothesis. *Annu. Rev. Neurosci.* 23, 649–711.

Meyer, D., Bonhoeffer, T., Scheuss, V. 2014. Balance and stability of synaptic structures during synaptic plasticity. *Neuron* 82, 430–443.

Miermans, C.A., Kusters, R.P.T., Hoogenraad, C.C., et al. 2017. Biophysical model of the role of actin remodeling on dendritic spine morphology. *PLoS One* 12, e0170113.

Minitab, Inc. 2010. Minitab 17 Statistical Software [Computer software]. Available at: www.minitab.com.

O'Donnell, C., Nolan, M.F., van Rossum, M.C. 2011. Dendritic spine dynamics regulate the long-term stability of synaptic plasticity. *J. Neurosci.* 31, 16142–16156.

Pathania, M., Davenport, E.C., Muir, J., et al. 2014. The autism and schizophrenia associated gene CYFIP1 is critical for the maintenance of dendritic complexity and the stabilization of mature spines. *Transl. Psychiatry* 4, e374.

Penzes, P., Cahill, M.E., Jones, K.A., et al. 2011. Dendritic spine pathology in neuropsychiatric disorders. *Nat. Neurosci.* 14, 285–293.

Qiao, H., Li, M.-X., Xu, C., et al. 2016. Dendritic spines in depression: What we learned from animal models. *Neural Plast.* 2016, 8056370.

Rabiner, L.R. 1989. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 257–286.

Rabiner, L.R., Juang, B.H. 1993. *Fundamentals of speech recognition*. Englewood Cliffs, New Jersey.

Ruszczycki, B., Szepesi, Z., Wilczynski, G.M., et al. 2012. Sampling issues in quantitative analysis of dendritic spines morphology. *BMC Bioinformatics* 13, 213.

Segal, M. 2010. Dendritic spines, synaptic plasticity and neuronal survival: Activity shapes dendritic spines to enhance neuronal viability. *Eur. J. Neurosci.* 31, 2178–2184.

Segal, M. 2017. Dendritic spines: Morphological building blocks of memory. *Neurobiol. Learn. Mem.* 138, 3–9.

Stawarski, M., Rutkowska-Wlodarczyk, I., Zeug, A., et al. 2014. Genetically encoded FRET-based biosensor for imaging MMP-9 activity. *Biomaterials* 35, 1402–1410.

Szepesi, Z., Hosy, E., Ruszczycki, B., et al. 2014. Synaptically released matrix metalloproteinase activity in control of structural plasticity and the cell surface distribution of GluA1-AMPA receptors. *PLoS One* 9, e98274.

Von Bohlen Und Halbach, O. 2009. Structure and function of dendritic spines within the hippocampus. *Ann. Anat.* 191, 518–531.

Address correspondence to:
Prof. Dariusz Plewczyński
Laboratory of Functional and Structural Genomics
Center of New Technologies
University of Warsaw
Banacha 2c Street
02-097 Warsaw
Poland

E-mail: d.plewczynski@cent.uw.edu.pl