



**HAL**  
open science

## Sparse data analysis strategy for neural spike classification

Vincent Vigneron, Hsin Chen

► **To cite this version:**

Vincent Vigneron, Hsin Chen. Sparse data analysis strategy for neural spike classification. Computational Intelligence and Neuroscience, 2014, (elec. proc.). 10.1155/2014/757068 . hal-01054317

**HAL Id: hal-01054317**

**<https://hal.science/hal-01054317v1>**

Submitted on 29 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Research Article

# Sparse Data Analysis Strategy for Neural Spike Classification

Vincent Vigneron<sup>1</sup> and Hsin Chen<sup>2</sup>

<sup>1</sup> IBISC-Lab, Université d'Évry Val d'Essonne, 40 rue du Pelvoux, 91020 Courcouronnes, France

<sup>2</sup> Department of Electrical Engineering, National Tsing Hua University, No. 101, Sec. 2, Kuang-Fu Road, Hsin-Chu 30013, Taiwan

Correspondence should be addressed to Vincent Vigneron; [vvigne@iup.univ-evry.fr](mailto:vvigne@iup.univ-evry.fr)

Received 31 July 2013; Accepted 8 April 2014; Published 2 July 2014

Academic Editor: Sung Phil Kim

Copyright © 2014 V. Vigneron and H. Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many of the multichannel extracellular recordings of neural activity consist of attempting to sort spikes on the basis of shared characteristics with some feature detection techniques. Then spikes can be sorted into distinct clusters. There are in general two main statistical issues: firstly, spike sorting can result in well-sorted units, but by with no means one can be sure that one is dealing with single units due to the number of neurons adjacent to the recording electrode. Secondly, the waveform dimensionality is reduced in a small subset of discriminating features. This shortening dimension effort was introduced as an aid to visualization and manual clustering, but also to reduce the computational complexity in automatic classification. We introduce a metric based on common neighbourhood to introduce sparsity in the dataset and separate data into more homogeneous subgroups. The approach is particularly well suited for clustering when the individual clusters are elongated (that is nonspherical). In addition it does not need to select the number of clusters, it is very efficient to visualize clusters in a dataset, it is robust to noise, it can handle imbalanced data, and it is fully automatic and deterministic.

## 1. Introduction

Neurophysiologists assume that the brain encodes information in the firing rate of neurons, that is, the number of “spikes” over a temporal interval. While many powerful imaging techniques have been used in neuroscience, extracellular recording remains the only choice that provides resolution of neuron activity in the brain. However, multiple extracellular recordings are useful only when the spikes generated by different neurons can be correctly sorted.

Lewicki [1] reviewed numerous methods that have been proposed to classify spikes. The usual assumptions for spike sorting are (1) that all spikes generated by a specific neuron are characterized by a similar waveform, (2) that this waveform is unique, and (3) that this waveform is conserved for each neuron during a stationary recording [2]. Analysis of neural recordings requires first detecting action potentials, *spikes*, from noise, which is achieved with thresholding discrimination by manual or semiautomatic classification methods. The second process is spikes sorting and produces a number of “spike trains” corresponding to the temporal sequence of real signals [3–5].

Among different methods used for spike sorting, template matching is one of the most popular procedures. The usual practice to produce templates is to use a “supervisor,” that is, an experienced and knowledgeable operator, to preliminarily classify the waveforms following a selection of templates corresponding to distinct neurons. Few methods have dealt with unsupervised template creation. Atiya [3] for instance used the Isodata clustering algorithm to estimate typical spike shapes and then compared all possible combinations of templates to find the combination with the highest likelihood. Letelier and Weber [6] applied Bayesian probability theory to quantify the probability of both the form and the number of spike shapes. Zouridakis and Tam [7] proposed a procedure based on fuzzy *k*-means clustering algorithms to create reliable spike templates. Some authors [8–10] used independent component analysis (ICA) for distinguishing the spikes according to their sources; the independence assumption of the firing neurons helps to identify spikes from the same source. In [11] the occurrence time information of spikes and features related to the shape simultaneously is applied to estimate the interspike interval for each neuron and sort the spikes using a Monte Carlo algorithm. Pouzat et al.

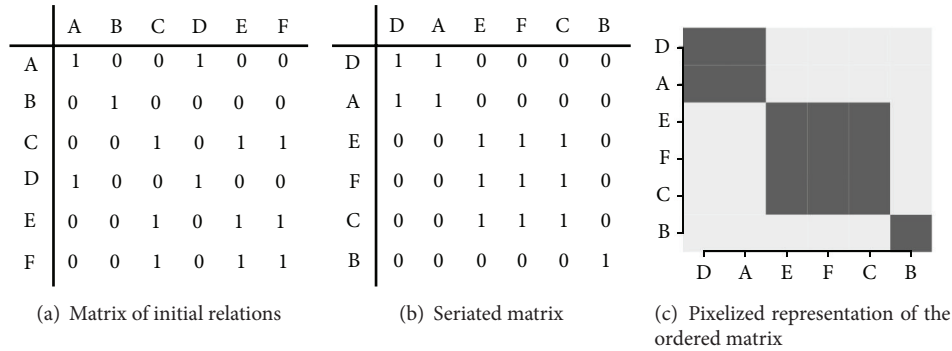


FIGURE 1: Effects of an algorithm of arrangement on a data set.

[12] used an empirical characterization of the recording noise to optimize the action potentials clustering and for assessing the quality of each cluster. Zhang et al. [13] reconstructed the spike templates according to the clustering results from principal component analysis (PCA) and subtractive clustering techniques. Probabilistic methods have been proposed [14, 15] and have focused on the modeling of each group in specific subspaces of low dimensionality.

Several approaches are associated with a visualization objective such as factorial methods [16, 17]. The latter methods can be *global* when they are based on the proximity between various groups such as graph methods or *local* when they evaluate the proximity between the individuals, like the hierarchical methods, they can also combine the local and global relations, as in the case of *seriation*.

A taxonomy of the methods was proposed by Carroll and Arabie [18] which associates a particular mode of seriation with each type of table.

Seriation aims to display and to reveal natural clusters and their dependencies in a dataset only by reordering rows and columns so that the adjacent rows and, respectively, columns are the most similar. This situation is illustrated by Figure 1 where, starting from a table of relations presented by Figure 1(a), the lines and the columns permuted to form a partition in which the similar elements were gathered together, thus forming groups (Figure 1(b)), and, in order to better appreciate the presence of the diagonal structure per block, this ordered matrix is pixelized (Figure 1(c)). Such an approach could be connected with a local technique of ordered clustering in so far as information is brought on one hand about the local relations between individuals because of an order in the data and on the other hand about the total structure of the data. Seriation has other advantages outlined by several authors such as Arabie et al. [19], like the no need of prior knowledge on the number of clusters and direct visualization of the structure on the table of values.

These advantages might disappear when the data are noisy or *imbalanced* or when groups of data are superimposed. The presence of noisy data prevents a clear visualization of the various blocks and distinguishing the clusters becomes a difficult task. Our approach is based on symmetric binary matrices of similarities (or dissimilarities) linked to common neighborhood. Such matrices indicate similarities

between pairs of observations and can be computed by different measures depending on the nature of the dataset such as Euclidean distances or more generally  $p$ -norm, correlation coefficients [20], or divergences [21] for example. A criterion derived from the problems of data compression selects the most compact ordered matrix—in the form of diagonal blocks—in order to obtain the most informative visualization off the intrinsic data structure. In some situations, too great a parsimony generates the ousting of underrepresented data forming very small clusters. To mitigate this nondetection, we propose a multiscale approach combining various levels of sparsity of the data.

This paper is organized in the following way: in Section 2, seriation is presented according to two different points of view, one as a mathematical optimization problem to be solved and the other on its algorithmic bases. Section 3 details our original approach as well as a multiscale algorithm of the proposed arrangement, called Parsimonious Block-Clustering. Experiments on simulations and benchmark data are presented in Section 4.

## 2. Method

**2.1. The Optimization Problem.** Seriation seeks an order in the data that reveals the locality/proximity between adjacent lines or columns to thus reveal a structure. This order is obtained by successive permutations of lines and columns which makes it possible to tackle seriation optimization problem (The number of possible combinations of permutations of lines and of columns is  $n!p!$  for a rectangular  $n \times p$  table or  $n!$  in the case of a symmetrical matrix of dissimilarity.) through two different angles: one being to determine all the best possible permutations, the other being related to the complexity of the solution ( $np$ -complete problem).

The seriation approach can be applied to any type of matrices but we focus in this work on dissimilarity matrices. Let us consider a set of  $N$  samples  $(x_1, \dots, x_N)$  described by a symmetrical matrix of dissimilarity  $D = (d_{ij})_{i,j \in \{1, \dots, N\}}$  of size  $N \times N$  where each element  $d_{ij}$  gives a “measure” of dissimilarity between the pair of observations  $(x_i, x_j)$ . Let  $\Psi$  define a permutation function which orders the elements of matrix  $D$ , according to a given criterion  $C$ . The objective of

the seriation is thus to find the optimal permutation  $\Psi^*$  which optimizes the arrangement criterion  $C$ , such that

$$\Psi^* = \arg \max_{\Psi} \mathcal{C}(\Psi(D)). \quad (1)$$

These criteria are based on a measure of similarity  $s(\cdot)$  between the successive elements of the matrix  $D$  and maximize  $\max \sum_{i=1}^{n-1} s(i, i+1)$ .

This measure of similarity is declined in a different way according to the authors as one can observe in Table 1. McCormick et al. [22] and Arabie and Hubert [23], for example, seek to maximize a *measure of effectiveness* (cf.  $\mathcal{C}_6$  criterion in Table 1) based on the sum of the scalar products in lines and columns of the data matrix; this measure was generalized thereafter by Climer and Zhang [20]. Other authors, such as Hubert et al. [24] or Chen [25], based their optimization on the divergence measure between the matrix of dissimilarity and an anti-Robinson structure seeking to gather the values of the smallest dissimilarities around the diagonal (cf.  $\mathcal{C}_4$  criterion). On the other hand, some authors such as Caraux and Pinloche [26] (cf.  $\mathcal{C}_1$  and  $\mathcal{C}_2$  criteria) or Brusco and Steinley [27] (cf.  $\mathcal{C}_3$  criterion) rather seek to place the smallest dissimilarities out of the diagonal (Robinson structure). Lastly, in the framework of data compression, Johnson et al. [28] proposed to minimize a criterion based on the number of sequences of consecutive elements (on a line) different from 0 (cf.  $\mathcal{C}_5$  criterion). Many authors proposed new criteria of arrangement like Niermann [29] who seeks to compare each observation with its adjacent neighbors through vicinity criteria (cf.  $\mathcal{C}_7$  criterion) or Batagelj [30] or Doreian et al. [31] who propose criteria of structural equivalence or Dhillon et al. [32] who use mutual information and an entropy-based criterion.

These recent approaches require a *prior* knowledge of the number of clusters formed by the individuals and the variables whose determination is not trivial.

**2.2. A Family of Embedded Binary Matrices.** To deal with the problem of imbalanced datasets, noisy data, overlapping clusters, and outliers, we propose a new algorithm based on a family of embedded binary matrices which stands for different degrees of sparsity of the data. The binary matrices are ordered according to an algorithm named Parsimonious Block-Clustering (PB-clus). This algorithm makes it possible to select the level of parsimony to produce the optimal compact block structure.

In our approach, the degree of vicinity is defined as a “threshold value” equal to the number of common neighbors between pairs of observations after which pairs of observations are eliminated. The larger the number of common neighbours imposed is, the more parsimonious the matrix will be (filled with zeros). Hence, the degree of parsimony is associated with the degree of common vicinity. Let us consider a data matrix  $X$  with elements in  $\mathbb{R}^p$  and  $X^d = (x_{ij}^d)$ ,  $i, j \in \{1, \dots, n\}$  the dissimilarity matrix associated to  $X$ , the choice of the distance function depending on the type of data: it can be an Euclidean distance between individuals  $i$  and  $j$  (and more generally  $p$ -norm), a correlation, or any other function characterizing the concept of proximity between

pairs of observations (see Table 1). Let  $A = (a_{ij})$ ,  $i, j \in \{1, \dots, n\}$ , and the (0,1)-matrix with elements

$$a_{ij} = \begin{cases} 1 & \text{if } x_{ij}^d \leq \epsilon \\ 0 & \text{if } x_{ij}^d > \epsilon, \end{cases} \quad (2)$$

where  $\epsilon$  is the threshold characterizing the proximity of the pairs of observations. Its value can be given arbitrarily; we propose to fix it at the first quartile of the distribution of the distances between pairs of observations. In addition, the matrix of similarity is symmetrical; that is,  $a_{ij} = a_{ji}$ . Let the Gram matrix  $B = A^T A$  where each element  $b_{ij}$  is the number of neighbors of the two data  $i$  and  $j$ . This matrix corresponds to a matrix of common vicinity.

*Definition 1.* A binary matrix  $B_{\lambda_m} = (b_{ij}^{\lambda_m})$ ,  $i, j \in \{1, \dots, n\}$ , parsimonious with a degree  $\lambda_m$  (with  $m \in \{1, \dots, M\}$ ) is characterized by

$$b_{ij}^{\lambda_m} = \begin{cases} 1 & \text{if } b_{ij} \geq \epsilon \\ 0 & \text{if } b_{ij} < \epsilon, \end{cases} \quad (3)$$

where  $b_{ij}$  represent the elements of the Gram matrix  $B$  defined previously. The set  $(B_{\lambda_1}, \dots, B_{\lambda_M})$  forms a family of binary matrices whose level of parsimony is related to the number of common neighbors.

Taking into consideration this definition, the greater  $\lambda_m$  the fewer the number of pairs of observations which satisfy this condition. The associated matrix will contain a greater number of zeros and will thus be more parsimonious. The sequence  $(\lambda_m)_{m \in \{1, \dots, M\}}$  such that  $\lambda_1 < \dots < \lambda_M$  makes it possible to establish an order relation  $\subset$  between the  $M$  elements of the set  $B_{\lambda_m} \ m \in \{1, \dots, M\}$ :

$$B_{\lambda_M} \subset B_{\lambda_{M-1}} \subset \dots \subset B_{\lambda_1}, \quad (4)$$

in which the most parsimonious matrix is contained in all the other matrices of its family. One of the advantages of such a matrix is the cancellation of the extreme values and of the noise when the level of parsimony increases, which facilitates the arrangement of the matrix as well as the appearance of adiaagonal block structure. In relation to this family of matrices, a question remains: how to obtain the “best” level of parsimony, that is, the one which will make it possible to obtain a comprehensive visualization of the data structure?

The ordered matrix  $B_{\lambda_m, \text{ord}}^* = (b_{ij, \text{ord}}^{\lambda_m})$ ,  $i, j \in \{1, \dots, n\}$ ,  $m \in I$  with the set  $I \in \{1, \dots, M\}$  contained in a set of ordered matrices, verifies that

$$B_{\lambda_m, \text{ord}}^* = \arg \min_{m \in I} \mathcal{C}_{\lambda_m} = \arg \min_{m \in I} \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \frac{|b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|}{|b_{ij}^{\lambda_m} - b_{i(j+1)}^{\lambda_m}|}. \quad (5)$$

This criterion is based on the idea that the fewer the alternations between the 0 and the 1 on the lines of the matrix considered, the more compact a structure this matrix

TABLE 1: Criteria of arrangement used within the framework of the clustering one-mode.

Type	Criterion to optimize depending on the dissimilarity matrix $D = \{d_{ij}\}_{i,j \in \{1, \dots, n\}}$
Structural criteria	$C_1 = \sum_{i=1}^n \sum_{j=1}^n d_{ij}  i - j ^2$
	$C_2 = \sum_{i=1}^n \sum_{j=1}^n (d_{ij} - \alpha  i - j ^2)$
	$C_3 = \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n (d_{ij} - \alpha  i - j ^2)$
	$C_4 = \sum_{1 \leq i < j < k \leq n} f(d_{ik}, d_{ij}) + \sum_{1 \leq i < j < k \leq n} f(d_{kj}, d_{ij}) \quad \text{with}$
	$f(x, y) = \text{sign}(x - y)$ $f(x, y) =  x - y  \text{sign}(x - y)$ $f(x, y) = \mathbb{1}_{x > y}$ $f(x, y) =  x - y  \mathbb{1}_{x > y}$
Similarity criteria	$C_5 = \sum_{j=i+1}^{n-1} \sum_{k=j+1}^n  d_{ij} - d_{i,j+1} $
	$C_6 = \sum_{i,j=1}^n d_{ij} (d_{i,j-1} + d_{i,j+1} + d_{i+1,j} + d_{i-1,j})$
	$C_7 = \sum_{i,j=1}^n f_{ij} \quad \text{with}$
	$f_{i,j} = \sum_{k=\max(1,i-1)}^{\min(n,i+1)} \sum_{\ell=\max(i,j-1)}^{\min(n,j+1)} (d_{ij} - d_{k\ell})^2$
	$f_{i,j} = \sum_{k=\max(1,i-1)}^{\min(n,i+1)} (d_{ij} - d_{kj})^2 + \sum_{\ell=\max(i,j-1)}^{\min(n,j+1)} (d_{ij} - d_{i\ell})^2$

will have. Indeed, in Table 2, if one considers the quantity  $\sum_{i=1}^n \sum_{j=1}^{n-1} |b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|$  accounting for the number of changes between the 0 and the 1 of an ordered matrix of degree  $\lambda_m$  and the quantity  $|b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|$  associated with the nonarranged matrix of the same degree, it is notable that the number of changes between the 0 and the 1 stays smaller in the case of the ordered matrices. As the degree of parsimony increases, the number of alternations between the 0 and the 1 falls: in the example, the numerator  $\sum_{i=1}^n \sum_{j=1}^{n-1} |b_{ij, \text{ord}}^{\lambda_m} - b_{i(j+1), \text{ord}}^{\lambda_m}|$  is equal to 9 for a level  $\lambda = 1$  and to 3 when the degree of parsimony is 3. In order for the selection criteria not to be biased in favour of an infinite sparsity,  $\mathcal{E}_{\lambda_m}$  is standardized by the number of alternations between the 0 and the 1 of the nonordered binary matrix associated with the same degree of parsimony. Thus, according to the example of Table 2, the level of parsimony retained is  $\lambda \geq 2$ .

Let us note that, at this level, a structure with two groups is selected and a piece of data that can be regarded as extreme data is excluded. This criterion derives from the concept of *run* used in *data compression* [28, 33], characterizing the biggest sequences of 1 on a line in a Boolean matrix. The chosen criterion  $\mathcal{E}_{\lambda_m}$  is related to the full number of changes present in the nonordered binary matrix of the same degree of parsimony so that it is not skewed in favour of an infinite parsimony or conversely, of too low a parsimony.

2.3. *The Pb-Clus Geometry-Based Criterion.* There are a plethora of criteria for the task of seriation [34] but the reordering algorithm that we proposed is based on the inner product because of its geometric interpretation. Since our work is based on symmetric matrices, the Tanimoto's norm (is also based on the dot product but adapted for binary data.) defined by  $x_i^T x_j / (x_i^T x_i + x_j^T x_j - x_j^T x_i)$  can be used for binary matrices  $B^{\lambda_m}$  of parsimony degrees  $\lambda_m$  ( $\forall m \in \{1, \dots, M\}$ ) defined in Section 2.2.

The permutation function  $\Psi$  which seeks to optimize the sum of the consecutive scalars can be written as

$$\Psi^* = \arg \max_{\Psi} \sum_{i=1}^{n-1} \frac{b_{\Psi(i)}^{\lambda_m T} b_{\Psi(i+1)}^{\lambda_m}}{\|b_{\Psi(i)}^{\lambda_m}\| + \|b_{\Psi(i+1)}^{\lambda_m}\| - b_{\Psi(i)}^{\lambda_m T} b_{\Psi(i+1)}^{\lambda_m}}. \quad (6)$$

This criterion is based on the principle of *connected components*: when several observations share the same neighborhood then these observations will belong to the same cluster or to the nearest clusters. The algorithm is based on a branch and bound method meaning that an exhaustive search is made in various subsets that are determined by the geometric properties of the dot product: the algorithm first searches the independent vectors which the separated clusters produce, then considers the connected component of each of these vectors and finally, and reorders the correlated vectors in each group. These steps can be done for a binary neighborhood matrix  $B_{\lambda}$  with level  $\lambda$  in the following way.

TABLE 2: An example of calculation of the  $C_{\lambda_m}$  criterion calculated from the matrix of the introductory example of Figure 1(a).

Matrix of common neighbors	$B = \begin{matrix} 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 3 & 3 \\ 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 & 3 & 3 \\ 0 & 0 & 3 & 0 & 3 & 3 \end{matrix}$		
Parsimony level	$\lambda \geq 1$	$\lambda \geq 2$	$\lambda \geq 3$
Binary matrices of common neighbors	$B = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{matrix}$	$B = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{matrix}$	$B = \begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{matrix}$
$\sum_{i=1}^6 \sum_{j=1}^5  b_{i,j}^{\lambda_m} - b_{i,j+1}^{\lambda_m} $ (calculus per line)	17 (3 + 2 + 3 + 3 + 3 + 3)	15 (3 + 0 + 3 + 3 + 3 + 3)	9 (0 + 0 + 3 + 0 + 3 + 3)
Sorted binary matrices	$B_1^{\text{sort}} = \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$	$B_2^{\text{sort}} = \begin{matrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$	$B_3^{\text{sort}} = \begin{matrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$
$\sum_{i=1}^6 \sum_{j=1}^5  (b_{i,j}^{\lambda_m})_{\text{sort}} - (b_{i,j+1}^{\lambda_m})_{\text{sort}} $ (calculus per line)	9 (1 + 1 + 2 + 2 + 2 + 1)	85 (1 + 1 + 2 + 2 + 2 + 0)	3 (1 + 1 + 11 + 0 + 0 + 0)
Criterion $C_{\lambda_m}$	$C_{\lambda_m \geq 1} = 9/17 = 1.89$	$C_{\lambda_m \geq 2} = 8/15 = 1.88$	$C_{\lambda_m \geq 3} = 3/9 = 3.03$

- (1) Compute a matrix of dot products (inner products or Tanimoto's product) for each pair of columns of  $X_\lambda$  without considering the columns full of zeros.
- (2) Select a column and find its connected components. Then find an orthogonal vector of the previous column and extract its connected components. This procedure is performed until there are no more vectors. In this way, several independent submatrices are built.
- (3) In each submatrix, place the most correlated vector alongside the first column and keep on doing this process until the submatrix is reordered.
- (4) Gather the rearranged submatrices and apply this order to  $B_\lambda$ .

The most informative visualization in terms of block-matrix is derived from the concept of *run* in compression approaches which characterizes a maximal sequences of nonzero entries in a row of a Boolean matrix [33]. It is intuitive that the fewer changes between series of ones and zeros are on each row the better the reordered matrix is. Since the sizes of the binary neighborhood matrices are different, this quantity is normalized by the minimum between the number of zeros or the number of ones of each rows so that

$$\mathcal{E}_\lambda = \sum_{i=1}^{n_\lambda} \frac{\text{card}_i(0, 1) + \text{card}_i(1, 0)}{\min(\text{card}_i(0, 0), \text{card}_i(1, 1))}, \quad (7)$$

where  $n_\lambda$  is the number of nonzero columns of the reordered matrix  $B_\lambda$ .

The algorithm enables us to find all the connected components of a cluster and to display relationships between clusters. This algorithm is straight forward deterministic algorithm, meaning that for a current move, the previous permutations are not challenged. Such an approach does not pretend to be optimal compared with the other approaches proposed in the literature but remains efficient and very fast even for large datasets and performs well when the data are noisy.

Since the proposed algorithm is a forward procedure (see Table 1), the final rearrangement obtained depends strongly on the first column selected in each submatrix. To deal with this problem, we propose to select a central observation for each submatrix to initialize the algorithm. The initialization is based on the idea that if we find a central observation in each cluster, then all connected components can be gathered. So, the first column is selected according to the number of strong correlated vectors which has to be maximum.

Lastly, Pb-Clus has a higher cost of calculation than the other methods of seriation since the arrangement is carried out not on only one matrix but on  $M$  matrices relative to different degrees of parsimony. In the case of a matrix of size  $n \times n$  with  $K$  groups of same size  $n/K$ , there are at most  $K(n/K)!$  calculations. As the degree of parsimony increases, the matrix is filled with columns (lines) of zeros, which decreases the number of elements to be arranged, and

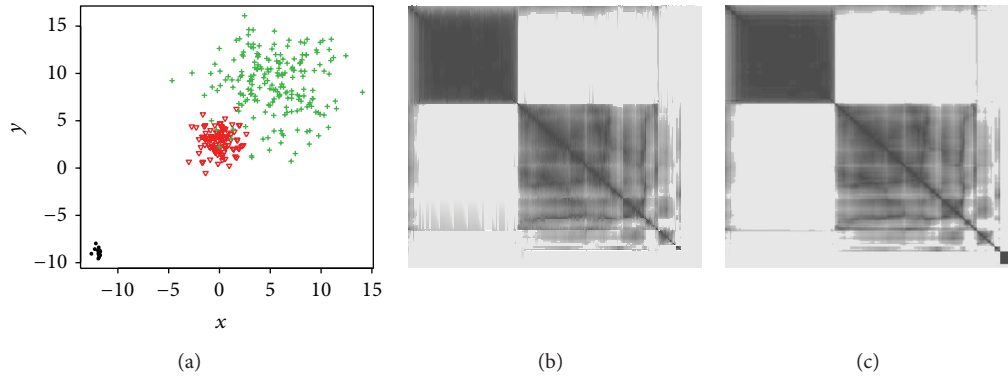


FIGURE 2: (a) Projection of the data in their space (b) visualizations of the central matrix ordered by Pb-Clus with a sparsity level of 16 common neighbors and (c) 8 common neighbors.

consequently the computing time. The calculation cost would remain significantly lower than  $M \cdot K(n/K)!$ .

### 3. Experiments on Simulated Data

**3.1. Case of Non-Separated Clusters.** For this experiment, the data are simulated from three different 2-dimensional Gaussian mixtures with large variances and two clusters are superposed as illustrated in Figure 2(a). The first cluster is formed of 5% of the data (15 observations) while the two others account for 32% and 63% of the data, respectively (i.e., 100 and 200 observations). The central partition linked to this situation is represented in Figure 2(b) with a sparsity threshold of 16 common neighbors.

For this level of parsimony, more than 6% of the data were excluded which results in the removal of the smallest cluster. For a level of 8 common neighbors, it is possible to recover the third cluster.

Even if the central visualization Figure 2(c) is a bit less clear than previously, it is still informative and three different clusters can be seen. Moreover, the superposition of two clusters can be identified since in the central visualization, the two relative squares are inscribed in a bigger square which means correlations or proximities between these two groups. Lastly, among the seriated data, 98% have been correctly classified.

**3.2. Influence of the Level of Superimposition of Clusters.** In this second experiment, we seek to evaluate the influence of the level of covering of clusters in the search for a data structure. With this intention, we simulated 3 Gaussian distributions in a 2-dimensional space so that their respective averages check:  $m_1 = (x, y)^T$ ,  $m_2 = (x, -y)^T$ ,  $m_3 = (0, -y)^T$  with  $y \in [0, 0.3]$ , and  $y \in [0, 0.225]$ . Consequently, the relative position of the averages varies and this variation determines the level of superposition of the groups. Thus, when  $x = 0$  and  $y = 0$ , the 3 groups are mixed and that corresponds to a superposition of 100%. In the opposite case of separate groups where the covering rate is zero, the averages of the clusters check:  $m_1 = (0.3, 0.225)^T$ ,  $m_2 = (0.3, -0.225)^T$ ,  $m_3 = (0, -0.225)^T$ . Table 3 presents the

evolution of the sparsity level and its associated ousting rate, according to the covering of the groups.

First of all, one notices that the greater the superposition of the clusters is the more the  $\mathcal{E}_\lambda$  criterion selects a parsimonious representation of the data. Indeed, when the visible data structure becomes less marked, this effect is balanced by a greater sparsity in the data with a bigger common vicinity. In the same way, as the data structure becomes increasingly complex, the rate of classification related to the subsets of seriated data decreases as well as the quality of visualization. In our example, beyond a rate of covering of the data of 40%, the rate of classification becomes weak (<60%) since the algorithm Pb-Clus no longer detects a structure in the data and this, whatever the level of parsimony imposed.

**3.3. Case of Noisy Data.** In this experiment, 30% of the data are replaced by a uniform noise in a hypercube  $[-1, 1]^4$  and the rest of the data are distributed from a mixture of three closed four dimensional Gaussian distributions as illustrated in Figure 3(a). Figure 4(c) depicts the central visualization which brings out a natural structure of three clusters in the dataset even if the data are noisy.

Figure 3(b) presents the evolution of the compactness criterion  $\mathcal{E}_\lambda$  according to the various degrees of parsimony, namely, the number of common neighbors. The central partition (Figure 3(c)) selected is the one for which the  $\mathcal{E}_\lambda$  criterion is minimal. This corresponds to a common vicinity of 59. This sparsity results in the ousting of 16% of the data and only 84% of the initial data make it possible to obtain a block diagonal representation; the subsets of excluded data are entirely made of noisy data. The rate of correct classification among the seriated data amounts to 99%, which implies that these subsets of seriated data are a structural visualization of the 3 clusters. In order to evaluate the performance of our approach, three methods of seriation based on distance matrices were applied: hierarchical clustering (HC) for the seriation (Figure 4(a)), the approach of Chen based on an anti-Robinson structure [25] (Figure 4(b)), and another method of anti-Robinson seriation by simulated annealing [35] (Figure 4(c)).

TABLE 3: Influence of the degree of covering of the clusters on the structure detection.

Degree of covering (en%)	0	6.7	13.3	20.0	26.6	33.3	40.0	46.6	53.30	73.3	100
$x$	0.30	0.28	0.26	0.24	0.22	0.10	0.18	0.16	0.14	0.08	0
$y$	0.225	0.21	0.195	0.18	0.165	0.15	0.135	0.120	0.09	0.06	0
Degree of parsimony	5	6	9	35	33	34	35	35	35	35	34
% of evinced values	0.00	0.00	0.00	0.26	0.23	0.34	0.37	0.35	0.39	0.35	0.43
Value of $C_\lambda$	1.95	2.01	2.42	2.64	2.90	2.82	3.34	3.29	3.32	3.54	3.65
Classification rate	0.99	0.99	0.99	0.99	0.95	0.90	0.86	0.60	0.49	0.44	0.39

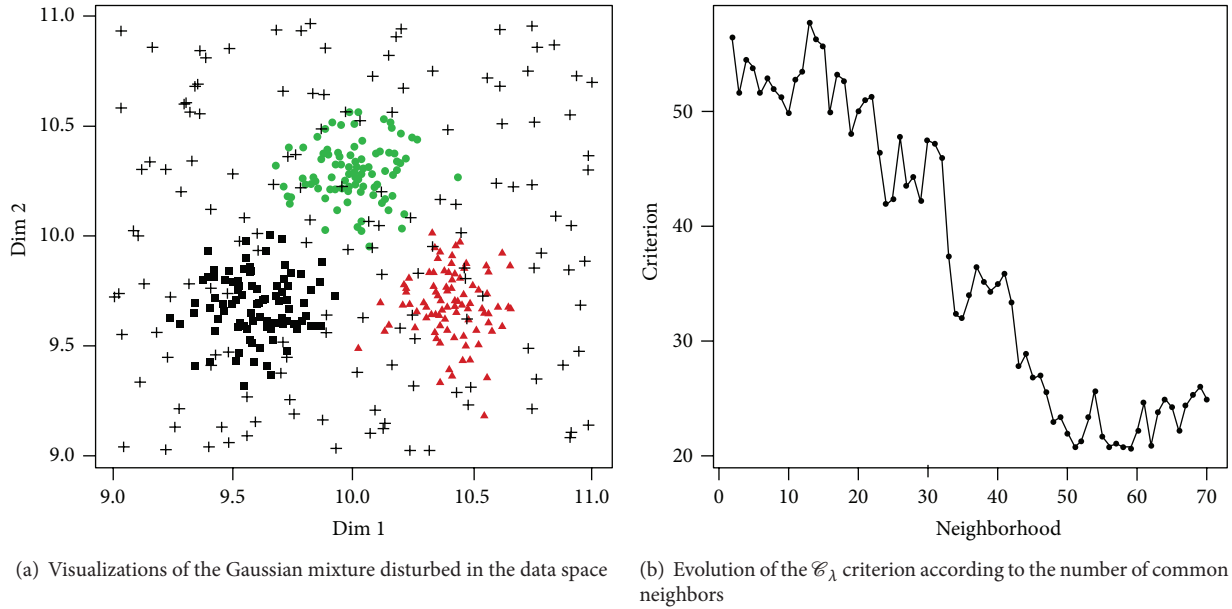


FIGURE 3: Seriation in the case of noisy data.

Among the methods of seriation used, we notice that only the central partition provided by Pb-Clus brings a clear visualization of the three clusters. The representation of this structure in three distinct groups is possible thanks to the family of parsimonious binary matrices. Indeed, the higher the degree of parsimony in the matrices, the greater the decrease in the quantity of noisy data taken into account.

3.4. *Influence of the Noise Level.* This second experiment aims to demonstrate the behavior of Pb-Clus in the case

of very noisy data. For this purpose, we simulated three 2-dimensional Gaussian distributions of 50 observations each with the following means  $m_1 = (-0.4, -0.3)^T$ ,  $m_2 = (-0.4, -0.3)^T$ , and  $m_3 = (0, 0.3)^T$ , respectively, and matrix of variance-covariance  $S = \text{diag}(0.1, 0.1)$ . These groups are voluntarily separated in order to be able to evaluate the sensitivity of the algorithm to the noise. The noisy data were generated according to a uniform law on the support  $[-1, 1]^2$ . To evaluate the impact of the noise on visualization, we varied the quantity of noise from 10% to 200% of the number of data



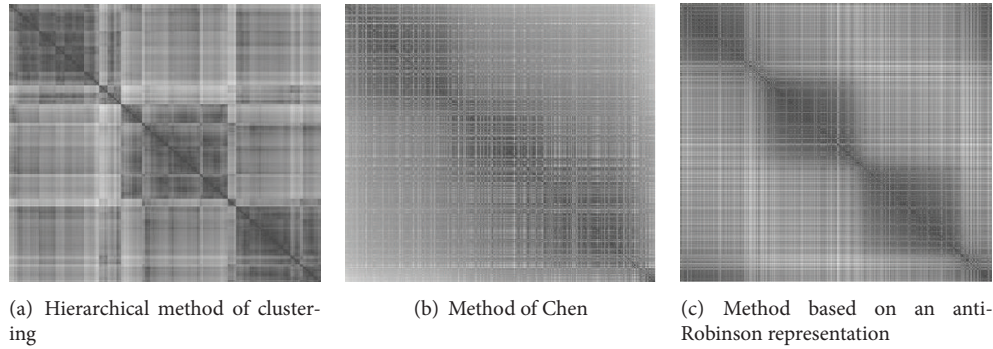


FIGURE 4: Visualizations of the pixelized distance matrix seriated.

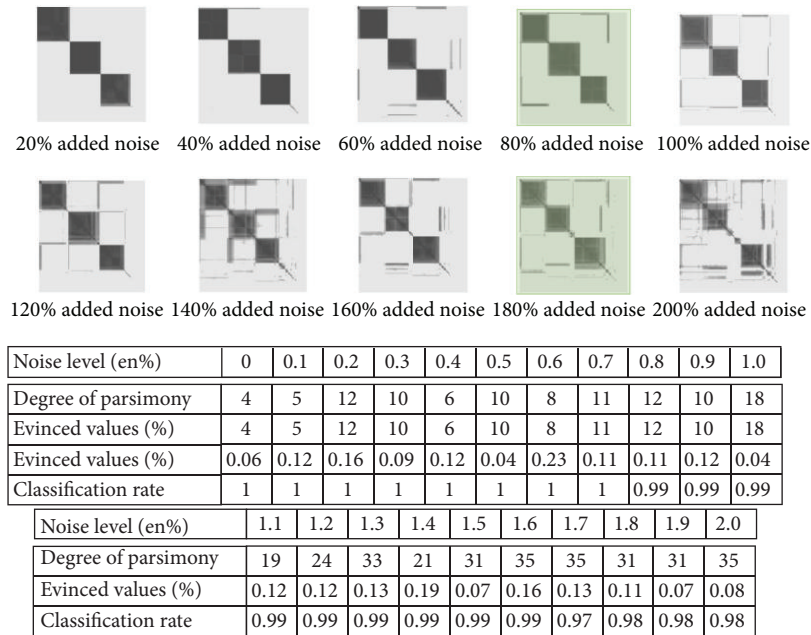


FIGURE 5: Visualization of the data structures according to various levels of noise added to the initial data.

in the initial sample. Figure 5 presents how the visualization of the data evolves with additional noise.

One notes that the group visualization degrades little with the additional noise. Indeed, in Figure 5, the structure is degraded only when the disturbed data represent more than half of the whole data.

**3.5. Comparison on Classical Datasets.** In this section, we compare the performance of PB Clus in terms of visualization firstly with two other methods of seriation, one using hierarchical classification (HC) and the other using a criterion of divergence related to an anti-Robinson structure described in Hahsler et al. [36] and, secondly, with an unsupervised classification method based on the Euclidean distances, the  $k$ -means. The 5 chosen datasets are detailed below.

- (i) Fisher's irises database collects 3 different species of iris in the Gaspé peninsula: *setosa*, *virginica* and the *versicolor*. Each species is represented by 50 flowers

which are described by 4 morphometric characteristics based on the width and the length of their sepals and their petals. This database is extremely popular in the statistical community because of difficulty of distinguishing the *virginica* and the *versicolor*.

- (ii) The *ruspini* data come from work of Ruspini [37] on clustering: they are made of 75 points in 2 dimensions and divided into 4 homogeneous and balanced classes.
- (iii) The *townships* data are binary data reporting the presence or the absence of 9 descriptive characteristics of 16 cities, such as the presence or the absence of universities, agricultural cooperatives, and railroads. There is no information on the number of groups structuring the data.
- (iv) *Old Faithful geyser data* evaluate the time between two eruptions of geysers of the national park of

TABLE 4: Comparison of 3 methods of seriation, PB-Clus, HC, and Chen approach according to Moore and Neumann criteria on the data benchmarks.

Method Criterion	PB-Clus seriation		HC seriation		Chen seriation	
	Moore	Neumann	Moore	Neumann	Moore	Neumann
Dataset						
Iris	1371.2	471.1	31 728.8	10 893.1	19 357.8	7 304.0
Townships	244.5	91.8	1 109.9	441.5	849.0	342.0
Ruspini	1 290.1	442.2	8 724.9	3 036.4	6 503.7	2 277.1
Faithful	2 634.1	889.4	34 045.5	11 503.5	23 390.0	9 894.2
Geysers	2 514.9	850.4	68 205.3	2 302.1	12 866.8	4 501.4

Yellowstone of Wyoming (USA) and their duration. They are characterized by 272 observations [38].

- (v) The *geysers* data represent a full version of the preceding data that were collected by Azzalini and Bowman [39]. These relate to the 299 eruptions which were studied (same types of measurements as previously) between 1st and 15th August, 1985.

The quality of the visualization is calculated from two criteria proposed by Niermann [29] and presented in Section 2; the partition obtained will be evaluated by cross-validation with the true label when available or with the labels estimated by the  $k$ -means. As the latter supposes a prior knowledge of the number of groups of the mixture, we use the number of clusters detected by Pb-Clus in order to obtain comparable partitions.

The right-hand column of Figure 6 represents the consecutive dot products of elements  $i$  and  $i + 1$  ordered out of the 5 previous databases. These curves of consecutive dot product give an evaluation of the proximity between two adjacent observations and points of rupture for the passage of one cluster to another, which makes it possible to select the number of clusters in the mixture and to obtain a partition of the data. In Figure 6 the left-hand column of represents the central visualization of the parsimonious matrix ordered with the algorithm PB-Clus. In the case of the Fisher's irises, the observation of its central matrix of degree of common vicinity 8 shows a total structure of two clusters.

One finds here the particular structure of the irises in which the *versicolor* and the *virginica* are not very distinct species. In addition, this partition in 3 groups is confirmed by the 2 break points present on the curves of its consecutive dot products. These 2 graphs demonstrate the performance of our parsimonious approach for the visualization of the data, especially as the methods of clustering which select one optimal model with 3 iris classes are rare (cf. mixture models of Raftery and Dean [40]). In the case of the *Ruspini* and the *Old Faithful* data, ruptures on the curve of the consecutive dot products are clear and large which show the total disconnection of the clusters between them. The same conclusion is visible on their ordered central matrix of degree 5 for the *Ruspini* data and of degree 2 for the *Faithful* data.

On the contrary, the *Geysers* and the *Townships* data present small breaking points. In the case of *Geysers* data, they are explained by the proximity of the clusters. Then, in

the case of the *Townships* data, the curve of the consecutive scalars shows that the first city is, certainly, connected to the 7 following cities but less strongly than these 7 cities between each other. The central visualization of the parsimonious ordered matrix of degree 2 with Pb-Clus brings a better comprehension of the relationships between the cities. Indeed, it is noticed that the first data is strongly correlated with two distinct blocks of cities. This is confirmed by an analysis of Hahsler et al. who showed the existence of a structure with 3 groups: urban cities, country towns, and transition cities. This first evaluation based on our visual perception is supplemented by the measure of quality based on seriation criteria evaluating the vicinity in the ordered matrix. Table 4 evaluates the performances of 3 methods of seriation, the best method being the one whose criterion is minimum. It is noticed that the 2 criteria of Niermann are minimum for a parsimonious approach for all the databases.

Lastly, Table 5 presents the tables of cross-classification with the true label for the irises of Fisher and with the labels obtained by  $k$ -means in the case of the data *Ruspini*, *Townships*, *Geysers*, and *Faithful*. Let us note that in the case of the *irises* and *Geysers* data, we threshold the scalars in order to obtain a label for each data. Concerning the Fisher irises, the correct classification rate of PB-Clus is 89.0%, slightly weaker than that obtained by the  $k$ -means (90.6%). This difference in rate is related to the data located at the intersection of the *virginica* and the *versicolor* and with initialization of our algorithm. For the other data files, one observes that the partitions obtained by Pb-Clus and the  $k$ -means agree almost perfectly, the rates of classification bordering 98%.

## 4. Experimental Methods

In this section, we approach the task of classifying spike waveforms using PB-Clus.

**4.1. Animal Training and Behavioral Tasks.** The detection of neural spike activity is a technical challenge that is a prerequisite for studying many types of brain function (for more details see Vigneron et al. [41]).

The study, approved by the Institutional Animal Care and Use Committee at the National Chiao Tung University, was conducted according to the standards established in the Guide for the Care and Use of Laboratory Animals. Four male rats weighing 250–300 g (BioLASCO Taiwan Corp., Ltd.)

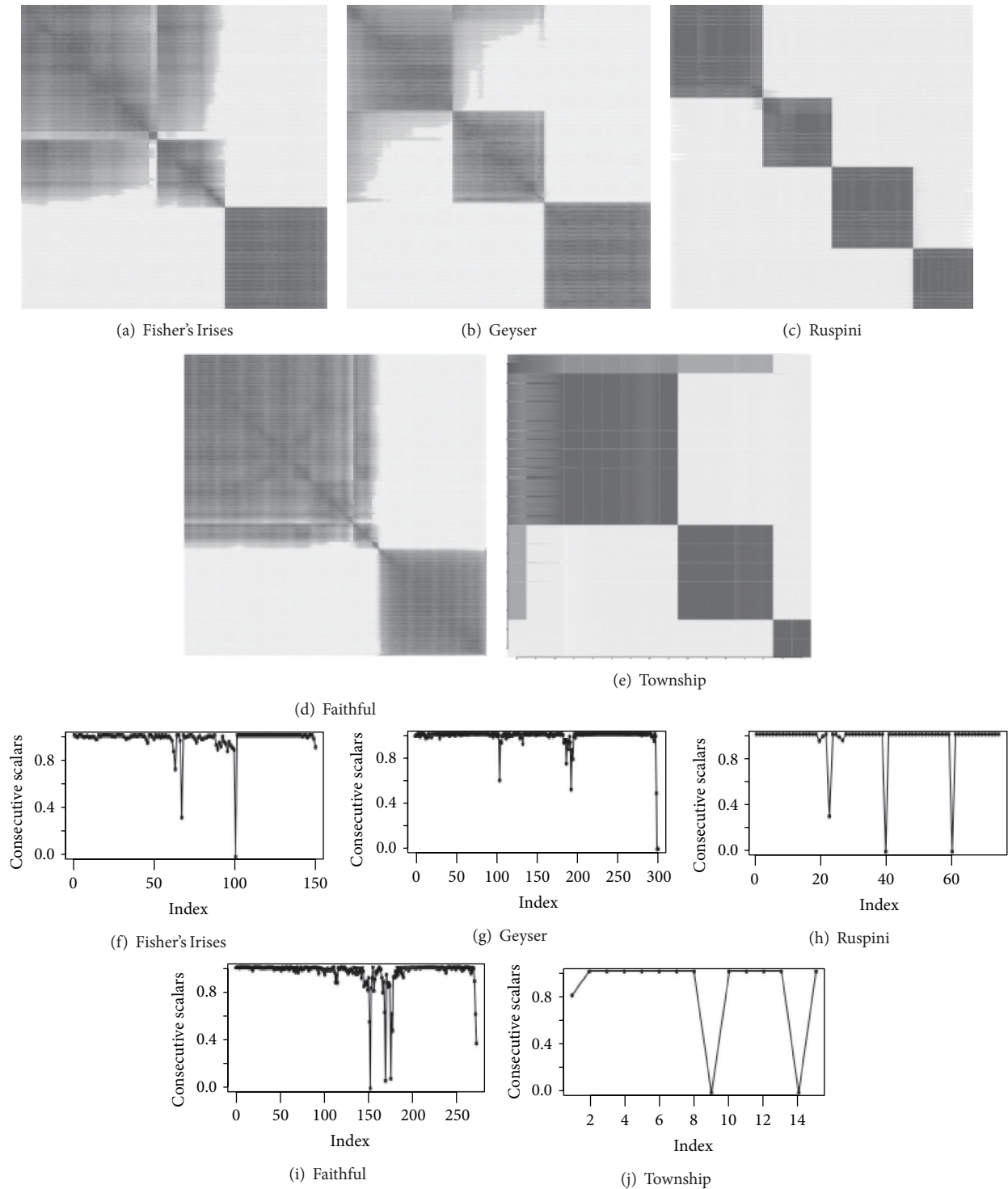


FIGURE 6: (a)–(e) rearranged matrices obtained with the PB-Clust (f)–(j) consecutive scalars resulting from the rearranged matrices.

were individually housed applying a 12 h light/dark cycle, with access to food and water *ad libitum*.

Dataset was collected from the motor cortex of awake animals performing a simple reward task. In this task, male rats (BioLACO Taiwan Co.,Ltd) were trained to press a lever to initiate a trial in return for a water reward. The

animals were water restricted 8-hours/day during training and recording session but food was always provided to the animal *ad lib* every day.

*4.2. Chronic Animal Preparation and Neural Ensemble Recording.* The animals were anesthetized with pentobarbital

TABLE 5: Tables of cross-validation of the data benchmarks.

(a)

Known classes	Fisher Iris			Known classes	Township							
	Clusters PB-Clus				Clusters $k$ -means							
	1	2	3		1	2	3	Known classes	Clusters PB-Clus			
Setosa	50	0	0	Setosa	50	0	0	Urban cities	8	0	0	0
Versicolor	0	50	0	Versicolor	0	49	1	Transitions	0	4	0	0
Virginica	0	17	33	Virginica	0	13	37	Country towns	0	0	2	0
								Unclassified	0	1	0	1
Classification rate = 0.88				Classification rate = 0.90				Classification rate = 0.94				

(b)

Clusters $k$ -means	Ruspini				Clusters $k$ -means	Faithful		Clusters $k$ -means	Geysers		
	Clusters PB-Clus					Clusters PB-Clus			Clusters PB-Clus		
	1	2	3	4	1	2		1	2	3	
Group 1	50	0	0	0	Group 1	168	4	Group 1	88	2	7
Group 2	0	35	0	0	Group 2	0	100	Group 2	0	105	0
Group 3	0	0	15	0				Group 3	0	0	97
Group 4	0	0	0	20							
Classification rate = 1.00				Classification rate = 0.98				Classification rate = 0.97			

(50 mg/kg i.p.) and placed on a standard stereotaxic apparatus (Model 9000, David Kopf, USA). The dura was retracted carefully before the electrode array was implanted. The pairs of 8 microwire electrode arrays (no.15140/13848, 50 m in diameter; California Fine Wire Co., USA) were implanted into the layer V of the primary motor cortex (M1). The area related to forelimb movement is located anterior 2–4 mm and lateral 2–4 mm from bregma. After implantation, the exposed brain should be sealed with dental acrylic and a recovery time of a week is needed.

During the recording sessions, the animal was free to move within the behavior task box (30 cm × 30 cm × 60 cm), where rats only pressed the lever via the right forelimb, and then they received 1-mL water reward as shown in Figure 7. A multichannel Acquisition Processor (MAP, Plexon Inc., USA) was used to record neural signals. The recorded neural signals were transmitted from the headstage to an amplifier, through a band-pass filter (spike preamp filter: 450–5 kHz; gain: 15,000–20,000), and sampled at 40 kHz per channel. Simultaneously, the animal's behavior was recorded by the video tracking system (CinePlex, Plexon Inc., USA) and examined to ensure that it was consistent for all trials included in a given analysis.

**4.3. Preprocessing.** Neural activity was collected from 400–700 ms before to 200–300 ms after lever release for each trial. Action potentials (spikes) crossing set thresholds were detected and sorted and the firing rate for each neuron was computed in 33 ms time bins. Since the signals are collected with 10 nanometers invasive probes, the noise effects are limited.

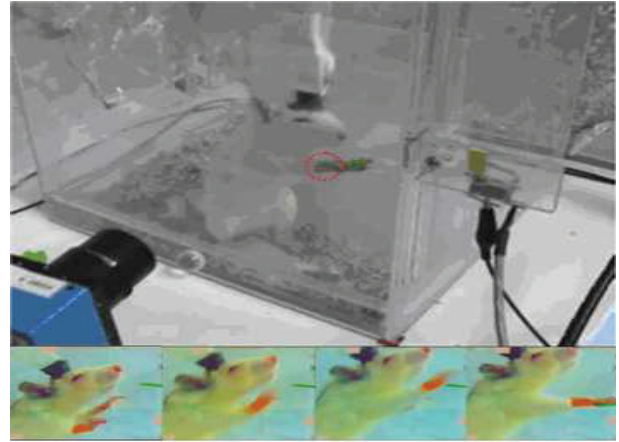


FIGURE 7: The experimental setup (top). Light-color (red virtual ring) was belted up the right forelimb to be recognized the trajectory by video tracking system. The sequence images captured the rat performing the lever press tasks in return for a reward of water drinking (bottom).

The experiment was made on 16 channels which collected EEG signals from microprobes which are implanted in the layer V of the M1 region of a rat.

**4.4. Manual Scatterplot Classification.** A method for classification is by plotting a selection of 2 or 3 spike features in a scatter diagram. This results in a 2- or 3-D graph with separate groups. The groups can only be assigned when there is enough spacing between the groups. Elliptic shaped areas are drawn around the groups isolating the classes.

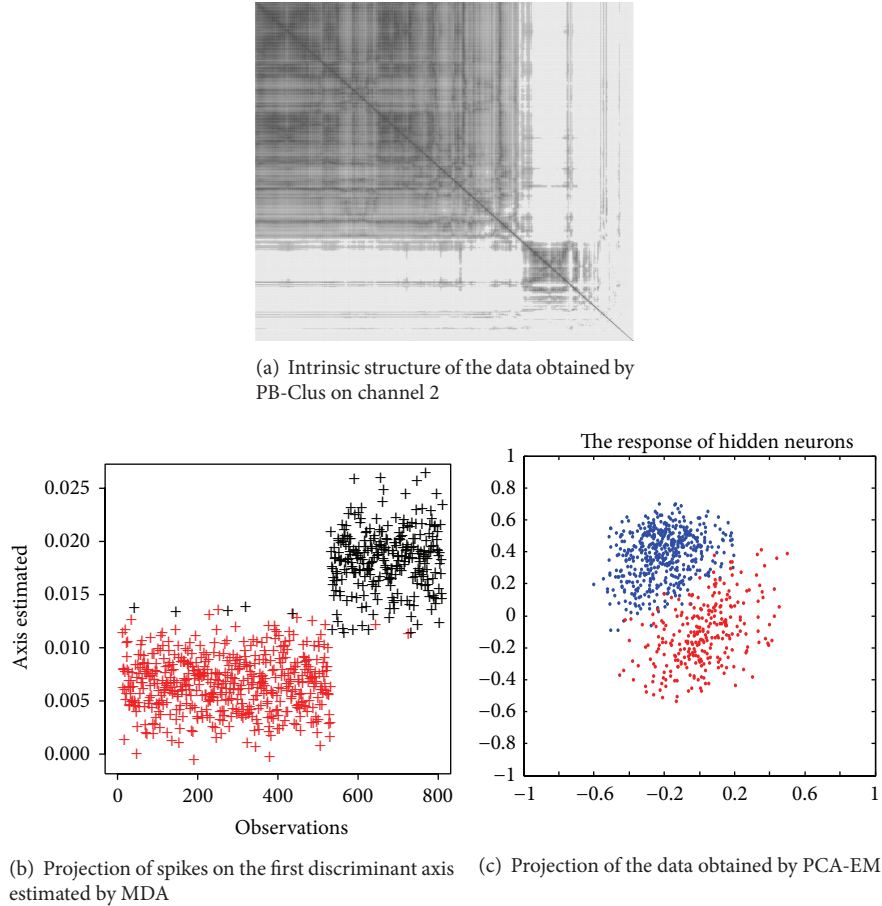


FIGURE 8: Results obtained by PB-Clus and 2 unsupervised approaches MDA and PCA-EM on channel 2.

TABLE 6: Number and type of spikes recorded in the 16 channels.

Channel	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of spikes	—	799	60	405	727	489	300	229	475	224	533	538	21	1833	1491	421
Types of spikes	—	2	2	2	1	2	4	1	1	2	2	2	1	2	1	4

*4.5. Spike Waveforms Classification.* To both reduce the size of these patterns and to cluster the spike mixture in a finite number of classes, we use two different tools: a seriation approach (PB-Clus) and a subspace clustering approach [42], named MDA (Mixture Discriminant analysis). Statistical discriminant analysis methods such as MDA aims to find both a parsimonious and discriminative fit for the data in order to ease the clustering and the visualization of the clustered data in a Gaussian mixture model context. MDA, developed by Hastie and Tibshirani [43], is a generalization of LDA (Linear Discriminant Analysis) in which each class is modeled by a mixture of Gaussians (see [44, chp. 4] for more details). This modelization gives more flexibility in the classification rule than LDA and allows MDA to take into account heterogeneity in a class. Breiman et al. [45], MacLachlan and Basford [46] have actually contributed and tested this generative approach on many fields. This latent subspace orientation is chosen such as it best discriminates

the groups. The quality of the partition obtained by both approaches will be measured by the Fisher index which is defined by the ratio between the within ( $S_w$ ) and the between ( $S_B$ ) scatter matrices:

$$F_{\text{index}} = \frac{S_w}{S_B} = \frac{\sum_{k=1}^K \sum_{i \in C_k} (x_i - m_k)(x_i - m_k)^t}{\sum_{k=1}^K n_k (m_k - \bar{x})(m_k - \bar{x})^t}, \quad (8)$$

where  $m_k = (1/n_k) \sum_{i \in C_k} x_i$  is the empirical mean of the observed column vector  $x_i$  in the class  $k$  and  $\bar{x} = (1/n) \sum_{k=1}^K n_k m_k$  is the mean column vector of the observations. Besides, both methods will be compared with a traditional approach of clustering which first reduces the dimension by principal component analysis (PCA) and then clusters the data in the projected space and refers in this paper to PCA-EM. Clustering accuracy will be computed between the partition obtained by both approaches and that obtained by a  $k$ -means approach.

**4.6. Results for Some Prominent Channels.** This first study aims to satisfy the existence of 4 classes of spikes. For this experiment, the clustering task was made channel by channel and, in each channel, we consider all the different events which correspond to movements of the rat. Finally, for each event, many spikes were recorded. Each normalized spike waveform is a time series that are of 32 dimensions.

Table 6 presents the number of spikes recorded in each of 16 channels and also the a priori number of kinds of spikes found by the preprocessing task. Besides, in the preprocessing task, as PCA components are computed so that different types of spikes are separated, we are going to first consider the projection on the 2 first components of PCA on each channel.

Figure 9 stand for the projection of the spikes of all the events of a selection of channels on the two first components. Whereas Table 6 describes the number of supposed types of spikes and given the preprocessing task, we expect to visualize on Figure 9 the intrinsic structure of the dataset where the number of separated clusters corresponds to those obtained in Table 6. However, it is difficult to visualize in Figure 9 a partition of several clusters in the data for each channel, whereas different clusters for channels 2, 7, and 16 can be observed in Figures 9(a), 9(e), and 9(k); such distinctions cannot be generalized since, on the other channels, it is not possible to visualize a group structure in the projected data. Without the label information of the preprocessing task, nothing enables us to suppose the true existence of different clusters. Furthermore it can be observed in Figure 9(b)—which stands for the projection of data of channel 3 plotted with the labeled spikes elaborated by the preclassification task—that the manual labels give no sense to a partition of the 2 groups of the data.

Consequently, from now, the proposed labels will not be taken into account and the main purpose of this work is to check the relevance of the preprocessing task. This study focuses on channels 2 and 7 whose datasets appear structured.

**4.6.1. On Channel 2.** The possible existence of two types of spikes in the axes of PCA in Figure 9(a) is satisfied by both the seriation and the subspace clustering approaches. In Figure 8(a) which represents the rearranged observations obtained by the Algorithm 1, one can observe 2 different blocks, one for each types of spikes in the data. In Figure 8(b) which stands for the projection of the data in the discriminative axes estimated by MDA algorithm, it can be observed that the clusters appear to be well separated compared with those obtained in the PCA axis. Figure 8(c) illustrates the projection of the data in the discriminative axes estimated by algorithm, it can be observed that the clusters appear to be well separated compared with those obtained in the PCA axis. Figure 8(c) which stands for the response of the supervised classification by PCA-EM approach has a similar representation of the data as those obtained by their projection in the 2 first principal components of PCA illustrated in Figure 9(a). In addition, Table 7 represents the Fisher index which has been computed for the different approaches previously presented. For the PB-Clus partition, the Fisher index is lower than those obtained by PCA-EM and

TABLE 7: Fisher index computed in channel 2 for PCA-EM, PB-Clus, and MDA.

Methods	PCA-EM	PB-Clus	MDA
$F_{\text{index}}$	864.5	210.1	588.7

MDA. It can be explained by the fact that PB-Clus introduces sparsity in the data, which produces smaller clusters that are more compact than those produced by PCA. Besides, the Fisher index for the MDA approach is equal to 588.7, which is equivalent to the result obtained by the PCA-EM classification ( $F_{\text{index}} = 585.0$ ) and lower than the PCA's one ( $F_{\text{index}} = 864.5$ ). Finally, to check the validity of the partition obtained by both methods, a cross-validation on the  $k$ -means results obtained by the work of [25] has been made. The contingency table and the clustering accuracy are presented in Table 8 and for each approach, it can be noted that 99% of the labeled data match with the PCA-EM labels. Consequently, it seems that, in channel 2, there are 2 different kinds of spikes and their respective shape obtained by both PB-Clus and MDA approaches is detailed in Figure 10.

**4.6.2. On Channel 7.** According to Figure 9(e), it can be observed on the first two components of PCA that there are at least 3 different groups of spikes. This remark is satisfied by the seriation approach, since Figure 11(a) which represents the intrinsic structure obtained by PB-Clus stresses 3 different kinds of spikes. In the same way, 3 components have been selected by using the Bayesian information criterion (BIC) for the mixture model in the case of PCA-EM, whereas both the preprocessing task and MDA, with the computation of BIC, have found 4 types of spikes. Figure 11(b) represents the projection of the clustered data on the 3 discriminant axes estimated by PCA-EM. In addition, since the  $k$ -means approach is based on the results of the preprocessing task, the prediction of the class membership of this dataset is made amongst 4 classes as can be seen in Figure 11(c).

Since the number of clusters varies between the different methods, data have been modeled by mixture models with 3 and then 4 components for both PCA-EM and MDA approaches, in order to be able to compare all the approaches. In Table 9 the Fisher index has been computed for the different cases. As expected, this criterion is much lower in the case of PB-Clus since it includes parsimony in the data whereas the ones obtained for MDA or PCA-EM remain high for a mixture of 3 components. Finally, the contingency table and the clustering accuracy are presented in Table 10. It can be observed that for the first case, PB-Clus detects the types 1, 2, and 4 of spikes whereas the 3rd type of spike is mixed with the first one. Furthermore, the classification rate reaches 91% on the spikes retained by PB-Clus when 40% of the data are ousted because of a high level of sparsity. In the second case, the partition obtained by MDA is comparable to these obtained by the PCA-EM classification except for type 1 which is mainly spread on type 3.

Finally, Figures 12(a) and 12(b) show the different spikes clustered by the PB-Clus and MDA algorithms. The difference

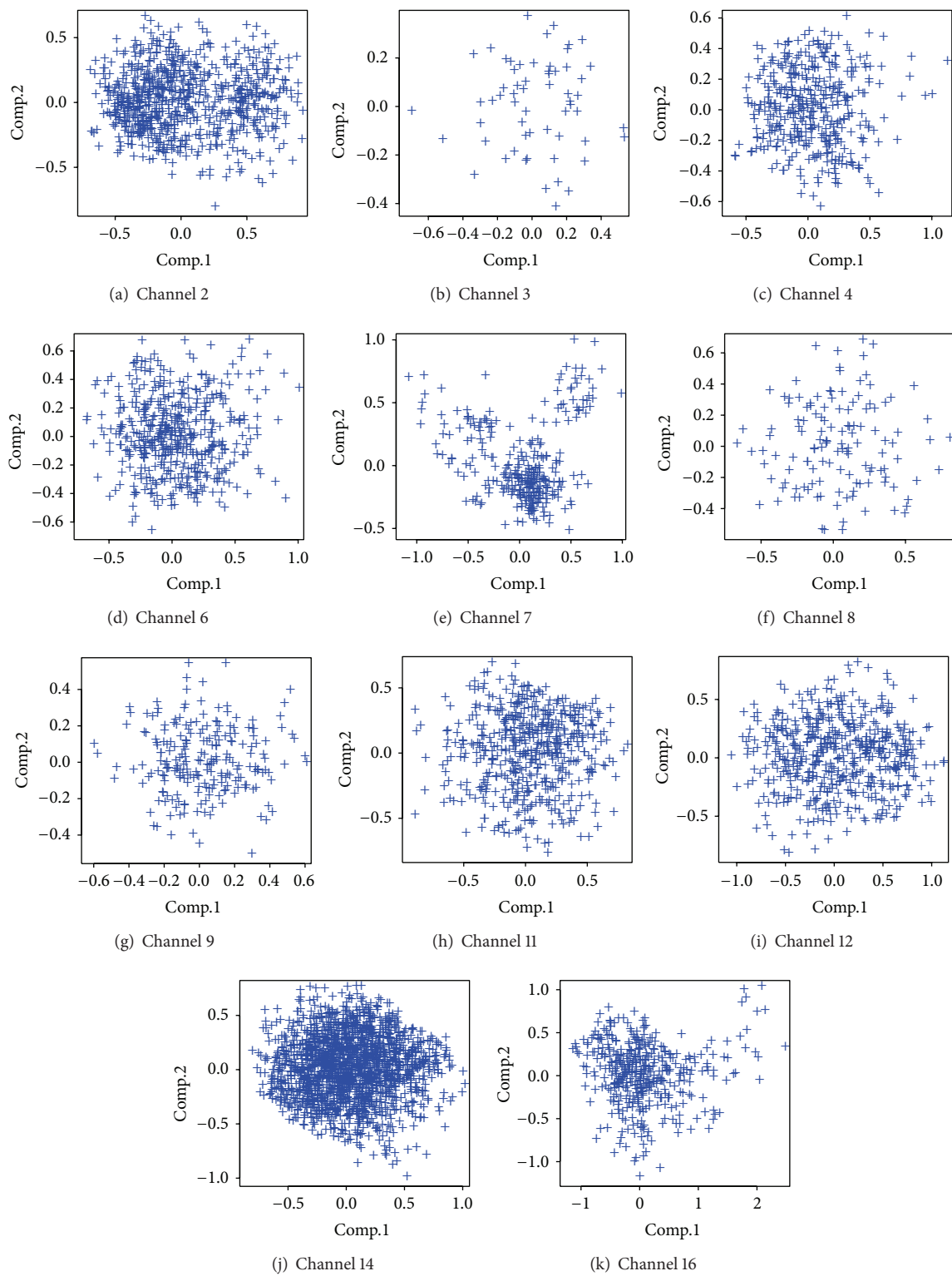


FIGURE 9: Projection of spikes of each channel on the 2 first components of PCA.

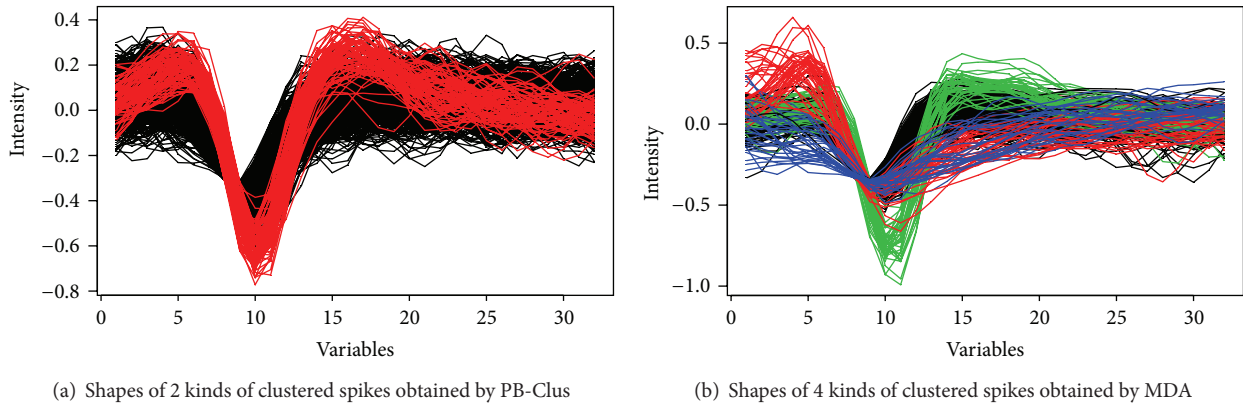


FIGURE 10: Clustered spikes obtained by PB-Clus in channel 2.

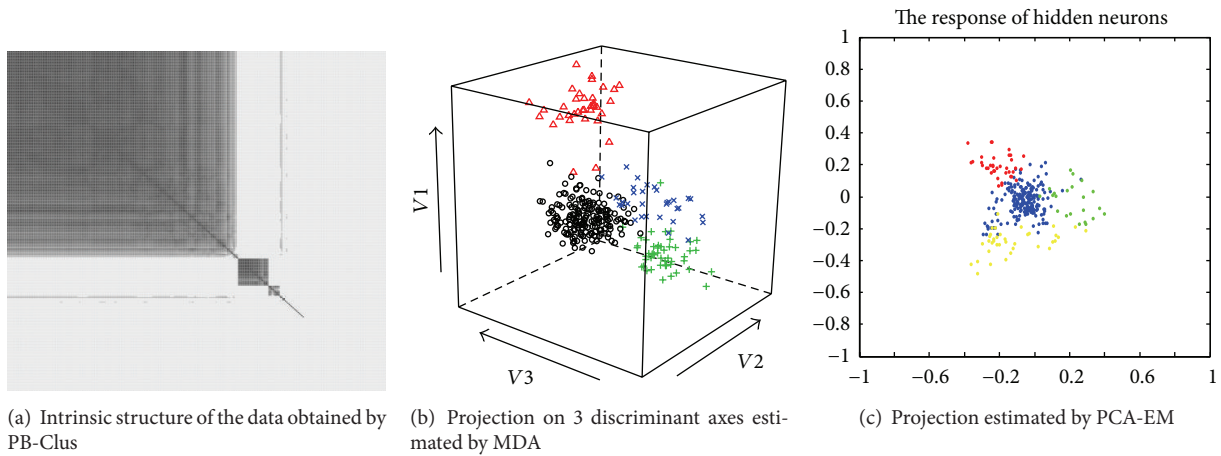


FIGURE 11: Results obtained by unsupervised approach PB-Clus and 2 supervised approaches MDA and PCA-EM on channel 7.

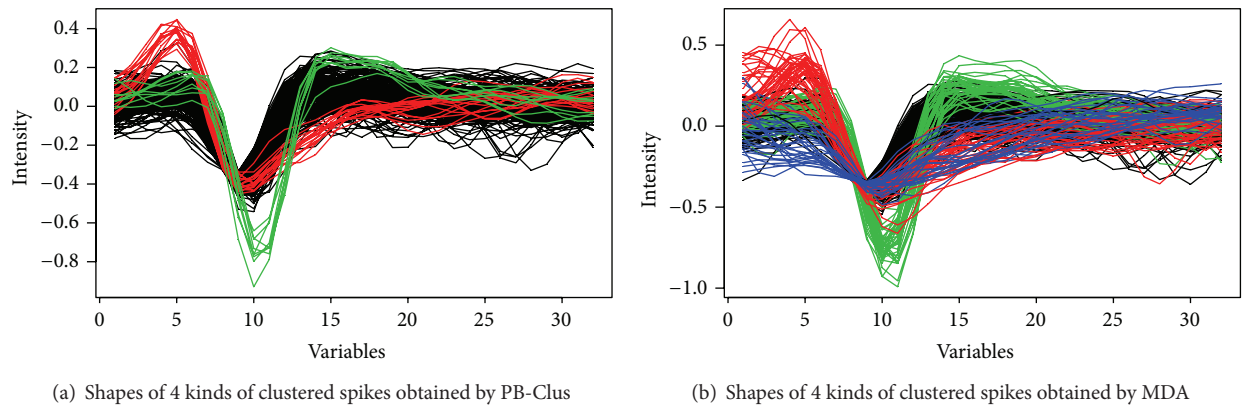


FIGURE 12: Clustered spikes obtained by PB-Clus and MDA for the channel 7.

between the two approaches is clearly seen on the 3rd type of spikes (blue in Figure 12) which is detected by MDA whereas it is not by PB-Clus. This could be explained by the weak dissimilarity between the shape of the 1st and the 3rd type of spikes (resp., black and blue in Figure 12) which is not taken into account by the measure of similarity, the

euclidean distance, used in PB-Clus. Different measures of similarity have been tried on PB-Clus such as Spearman correlation or maximum distances but have not brought any more information or any improvement for the visualization.

To conclude, given these results, the existence of 4 different types of clusters does not seem really relevant



**Require:** seq := vector of expected number of neighbors  
**Require:** liste $\mathcal{E}$  := vector of criterion values

- (1) **for all**  $i \in \text{seq}$  **do**
- (2)   Compute the binary matrix  $B_\lambda$  of common neighbors for  $\lambda = \text{seq}(i)$
- (3)   Compute  $B_\lambda$  the scalar products matrix of  $B_\lambda$
- (4)    $\lambda \leftarrow \text{sort}(S_\lambda)$  (sort the individuals)
- (5)    $C_\lambda \leftarrow \text{crit}(B_\lambda, S_\lambda)$ , compute the criterion
- (6)   liste $\mathcal{E} \leftarrow \text{merge}(\text{liste}\mathcal{E}, C_\lambda)$
- (7) **end for**
- (8)  $\lambda^* \leftarrow \arg \min_\lambda(\text{liste}\mathcal{E})$
- (9)  $B_{\lambda^*} \leftarrow \min(\text{liste}\mathcal{E})$
- (10)  $O_{\lambda^*} \leftarrow \text{ordre}(S_{\lambda^*})$   
 The PB-Clus algorithm returns the minimum of the criterion  $B_{\lambda^*}$ ,  
 the related number of common neighbors and the optimal sorting matrix  $O_{\lambda^*}$ .
- (1) **procedure** SORT( $S_\lambda$ )
- (2)  $V := \text{colinear}(i.\text{obs}, S_\lambda)$
- (3) liste := list of individuals whose common neighbors ( $cn$ ) is non-zero ( $cn = 0$ )
- (4)  $i.\text{max} :=$  the individual for which  $cn$  is maximum
- (5)  $i.\text{perm} := 0$  (list containing the ranking value of the individuals)
- (6) **repeat**
- (7)  $i.\text{perm} \leftarrow \text{merge}(i.\text{perm}, \text{liste}[V.\text{col}])$
- (8) **if** length( $V.\text{cor}$ )  $\geq 1$  **then**
- (9)    $S_j \leftarrow S_j[c(V.\text{cor}, V.\text{ind}), c(V.\text{cor}, V.\text{ind})]$
- (10)   liste  $\leftarrow \text{liste}[c(V.\text{cor}, V.\text{ind})]$
- (11)   **if** length( $V.\text{cor}$ )  $> 1$  **then**
- (12)      $V \leftarrow \text{colinear}(1, B_\lambda)$
- (13)   **end if**
- (14)   **else**
- (15)      $S_j \leftarrow S_j[V.\text{ind}, V.\text{ind}]$
- (16)     liste = liste[ $V.\text{ind}$ ]
- (17)     **if** length( $V.\text{cor}$ )  $> 1$  **then**
- (18)        $i.\text{max} = \arg \max_j cn(S_j)$
- (19)        $B_\lambda = B_\lambda[V.\text{ind}, V.\text{ind}]$
- (20)        $V = \text{colinear}(i.\text{max}, B_\lambda)$
- (21)     **end if**
- (22)   **end if**
- (23) **until** length(liste)  $> 1$
- (1) **procedure** COLINEAR( $i, S$ ) returns 3 different lists
- (2)  $s_i :=$   $i$ th line of  $S$
- (3)  $V.\text{col} :=$  list of individuals that are colinear with  $s_i$
- (4)  $V.\text{cor} :=$  list of individuals that are correlated with  $s_i$
- (5)  $V.\text{ind} :=$  list of individuals that are independent of  $s_i$

ALGORITHM 1: PB-Clus algorithm.

TABLE 8: Contingency tables for PB-Clus and MDA partitions with  $k$ -means classification in channel 2.

$k$ -means classes	PB-Clus		MDA	
	Clusters 1	Clusters 2	$k$ -means classes 1	Clusters 2
Type 1	214	0	Type 1	515
Type 2	0	79	Type 2	5
Classification rate = 100%		Classification rate = 99.12%		

TABLE 9: Fisher index computed in channel 7 for PCA-EM, PB-Clus, and MDA.

Methods	PCA-EM	PB-Clus	MDA
$F_{\text{index}}$ for 3 clusters	614.1	287.9	483.6
$F_{\text{index}}$ for 4 clusters	392.2	—	324.1

since some types of spikes, in particular types 3 and 4, are often mixed with the first type in both PB-Clus and MDA

approaches. Consequently, either the preprocessing task is biased since the different types of spikes do not really exist or the 32 dimensions of the studied spikes are not sufficient to discriminate the 4 different types of spikes.

TABLE 10: Contingency tables for PB-Clus and MDA partitions with  $k$ -means classification for the channel 7.

$k$ -means classes	PB-Clus			$k$ -means classes	MDA				
	1	2	3		1	2	3	4	
Type 1	183	11	0	Type 1	181	0	17	6	
Type 2	0	0	7	Type 2	1	36	0	0	
Type 3	5	0	0	Type 3	0	0	33	4	
Type 4	1	5	1	Type 4	0	0	0	22	
Classification rate = 91%				Classification rate = 90.1%					

## 5. Conclusion

Controlled numerical experiments using spike and noise data extracted from neural recordings indicate significant improvements in detection and classification accuracy compared with amplitude and linear template-based spike sorting techniques. Algorithm 1 makes it possible to visualize subsets of spike data and their dependencies. With this intention, we proposed a family of embedded parsimonious matrices of different levels of parsimony whose level is directly determined by the number of common neighbors between pairs of observations. This is an effective tool for the analysis of data, which offers better results visually than the traditional clustering methods, in particular when the data are noisy or imbalanced or when the groups are superposed.

Moreover, this parsimonious approach facilitates the interpretation of the data and offers a quality of partitioning comparable with the  $k$ -means method with the advantage of not posing any assumption about the number of clusters. In addition, choosing a level of parsimony in the data corresponds to seeking explicative subsets of a structure. This new point of view can be connected with an approach by levels of density, commonly called *level sets*, which was initially approached by Hartigan [47] and then by Nolan [48]. A comparison of these two approaches and the search for a theoretical bond are part of our research tasks in progress.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

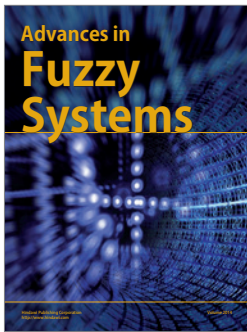
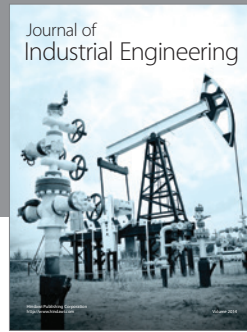
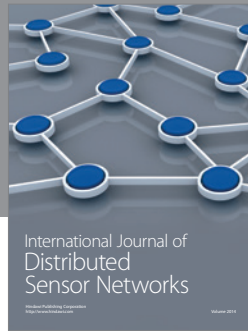
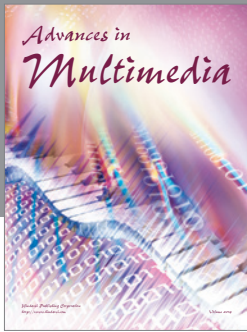
## Acknowledgments

This project was supported in part by funding from the Hubert Curien program of the Foreign French Minister and from the Taiwan NSC. The neural activity recordings were kindly provided by the Neuroengineering lab. of the National Chiao-Tung University.

## References

- [1] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, no. 4, pp. R53–R78, 1998.
- [2] D. A. Willming and B. C. Wheeler, "Real-time multichannel neural spike recognition with DSPs," *IEEE Engineering in Medicine and Biology Magazine*, vol. 9, no. 1, pp. 37–39, 1990.
- [3] A. F. Atiya, "Recognition of multiunit neural signals," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 7, pp. 723–729, 1992.
- [4] M. S. Fe, P. P. Mitra, and D. Kleinfeld, "Automatic sorting of multiple unit neuronal signals in the presence of anisotropic 473 and non-gaussian variability," *Journal of Neuroscience Methods*, vol. 9, pp. 175–188, 1968.
- [5] E. H. D'Hollander and G. A. Orban, "Efficient approximation and on-line classification by an unsupervised learning system," *IEEE Transactions on Biomedical Engineering*, vol. 26, pp. 279–284, 1979.
- [6] J. C. Letelier and P. P. Weber, "Spike sorting based on discrete wavelet transform coefficients," *Journal of Neuroscience Methods*, vol. 101, no. 2, pp. 93–106, 2000.
- [7] G. Zouridakis and D. C. Tam, "Identification of reliable spike templates in multi-unit extracellular recordings using fuzzy clustering," *Computer Methods and Programs in Biomedicine*, vol. 61, no. 2, pp. 91–98, 2000.
- [8] V. Vigneron, "Signal subspace separation based on the divergence measure of a set of wavelets coefficients," in *Independent Component Analysis and Blind Signal Separation*, vol. 5441, pp. 171–177, Springer, 2009.
- [9] S. Takahashi, Y. Anzai, and Y. Sakurai, "Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes," *Journal of Neurophysiology*, vol. 89, no. 4, pp. 2245–2258, 2003.
- [10] A. M. Mamlouk, H. Sharp, K. M. L. Menne, U. G. Hofmann, and T. Martinez, "Unsupervised spike sorting with ICA and its evaluation using GENESIS simulations," *Neurocomputing*, vol. 65–66, pp. 275–282, 2005.
- [11] C. Pouzat, M. Delescluse, P. Viot, and J. Diebolt, "Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: a Markov chain Monte Carlo approach," *Journal of Neurophysiology*, vol. 91, no. 6, pp. 2910–2928, 2004.
- [12] C. Pouzat, O. Mazor, and G. Laurent, "Using noise signature to optimize spike-sorting and to assess neuronal classification quality," *Journal of Neuroscience Methods*, vol. 122, no. 1, pp. 43–57, 2002.
- [13] P. M. Zhang, J. Y. Wu, Y. Zhou, P. J. Liang, and J. Q. Yuan, "Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem," *Journal of Neuroscience Methods*, vol. 135, no. 1–2, pp. 55–65, 2004.
- [14] J. P. Stitt, R. P. Gaumond, J. L. Frazier, and F. E. Hanson, "An artificial neural network for neural spike classification,"

- in *Proceedings of the IEEE 1997 23rd Northeast Bioengineering Conference*, pp. 15–16, May 1997.
- [15] P. Barthó, H. Hirase, L. Monconduit, M. Zugaro, K. D. Harris, and G. Buzsáki, “Characterization of neocortical principal cells and interneurons by network interactions and extracellular features,” *Journal of Neurophysiology*, vol. 92, no. 1, pp. 600–608, 2004.
- [16] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine Series*, vol. 2, no. 6, pp. 157–175, 1901.
- [17] I. T. Jolliffe, *Principal Component Analysis*, Springer, 1986.
- [18] D. J. Carroll and P. Arabie, “Multidimensional scaling,” *Annual Review of Psychology*, vol. 31, pp. 607–649, 1980.
- [19] P. Arabie, L. J. Hubert, and G. de Soete, “An overview of combinatorial data analysis,” in *Clustering and Classification*, pp. 5–63, World Scientific, 1996.
- [20] S. Climer and W. Zhang, “Rearrangement clustering: pitfalls, remedies, and applications,” *Journal of Machine Learning Research*, vol. 7, pp. 919–943, 2006.
- [21] T. Villman and S. Haase, “Mathematical aspects of divergence based vector quantization using fréchet-derivatives,” Tech. Rep., University of applied sciences Mittweida, 2009.
- [22] W. T. McCormick Jr., P. J. Schweitzer, and T. W. White, “Problem decomposition and data reorganization by a clustering technique,” *Operations Research*, vol. 20, no. 5, pp. 993–1009, 1972.
- [23] P. Arabie and L. J. Hubert, “Bond energy algorithm revisited,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 1, pp. 268–274, 1990.
- [24] L. Hubert, P. Arabie, and J. Meulman, *Combinatorial Data Analysis: Optimization by Dynamic Programming*, Society for industrial and Applied Mathematics, 2001.
- [25] C. H. Chen, “Generalized association plots: information visualization via iteratively generated correlation matrices,” *Statistica Sinica*, vol. 12, no. 1, pp. 7–29, 2002.
- [26] G. Caraux and S. Pinloche, “PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order,” *Bioinformatics*, vol. 21, no. 7, pp. 1280–1281, 2005.
- [27] M. Brusco and D. Steinley, “Inducing a blockmodel structure of two-mode binary data using seriation procedures,” *Journal of Mathematical Psychology*, vol. 50, no. 5, pp. 468–477, 2006.
- [28] D. Johnson, S. Krishnan, and J. Chhugani, “Compressing large boolean matrices using reordering techniques,” in *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB '04)*, vol. 30, pp. 13–23, 2004.
- [29] S. Niermann, “Optimizing the ordering of tables with evolutionary computation,” *American Statistician*, vol. 59, no. 1, pp. 41–46, 2005.
- [30] V. Batagelj, “Notes on blockmodeling,” *Social Networks*, vol. 7, pp. 143–155, 1997.
- [31] P. Doreian, V. Batagelj, and A. Ferligoj, “Generalized blockmodeling of two-mode network data,” *Social Networks*, vol. 26, no. 1, pp. 29–53, 2004.
- [32] I. S. Dhillon, S. Mallela, and D. S. Modha, “Information-theoretic co-clustering,” in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 89–98, August 2003.
- [33] T. Apaydin, A. Ş. Tosun, and H. Ferhatosmanoglu, “Analysis of basic data reordering techniques,” in *Scientific and Statistical Database Management*, vol. 5069, pp. 517–524, 2008.
- [34] I. van Mechelen, H. H. Bock, and P. de Boeck, “Two-mode clustering methods: a structured overview,” *Statistical Methods in Medical Research*, vol. 13, no. 5, pp. 363–394, 2004.
- [35] M. J. Brusco, H. F. Köhn, and S. Stahl, “Heuristic implementation of dynamic programming for matrix permutation problems in combinatorial data analysis,” *Psychometrika*, vol. 73, no. 3, pp. 503–522, 2008.
- [36] M. Hahsler, K. Hornik, and C. Buchta, “Getting things in order: an introduction to the R package seriation,” Tech. Rep. 58, 2009.
- [37] E. H. Ruspini, “Numerical methods for fuzzy clustering,” *Information Sciences*, vol. 2, no. 3, pp. 319–350, 1970.
- [38] W. Härdle, *Smoothing Techniques with Implementation*, Springer, New York, NY, USA, 1991.
- [39] A. Azzalini and A. W. Bowman, “A look at some data on the old faithful geyser,” *Applied Statistics*, vol. 39, no. 3, pp. 357–365, 1990.
- [40] A. E. Raftery and N. Dean, “Variable selection for model-based clustering,” *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 168–178, 2006.
- [41] V. Vigneron, Y. T. Chen, H. Y. Chen, and Y. Y. Chen, “Decomposition of eeg signals for multichannel neural activity analysis in animal experiments,” in *Independent Component Analysis and Blind Signal Separation*, vol. 6365, pp. 474–481, Springer, 2010.
- [42] V. Vigneron, C. Brunet, and T. Willman, “Une famille de matrices sparses pour une modélisation multi-échelle par blocs,” in *Revue des Nouvelles Technologies de l'Information*, pp. 123–147, RNTI, Hermann, Mo, USA, 2011.
- [43] T. Hastie and R. Tibshirani, “Discriminant analysis by gaussian mixtures,” Tech. Rep., AT & T Bell laboratories, Murray Hill, NJ, USA, 1994.
- [44] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.
- [45] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, Calif, USA, 1984.
- [46] G. MacLachlan and K. Basford, *Mixtures Models: Inference and Applications to Clustering*, Marcel Dekker, 1988.
- [47] J. A. Hartigan, *Clustering Algorithms*, John Wiley, New York, NY, USA, 1975.
- [48] D. Nolan, “The excess-mass ellipsoid,” *Journal of Multivariate Analysis*, vol. 39, no. 2, pp. 348–371, 1991.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

