

Marked Point Processes for Microarray Data Clustering

Khadidja Henni, Olivier Alata, Abdellatif El Idrissi, Brigitte Vannier, Lynda Zaoui, and Ahmed Moussa

Abstract Microarray technologies become a powerful technique for simultaneously monitoring expression patterns of thousands of genes under different conditions. However, it is important to identify gene groups that manifest similar expression profiles and are activated by similar conditions. *ClusterMPP: Clustering by Marked Point Process* is a new microarray data clustering algorithm performed in two steps. The first one detects cluster modes representing regions of high density observations in the raw space. Based on the simulation of a proposed Marked Point Process by the well-known Reversible Jump Markov Chain Monte Carlo algorithm, where we consider several movements like birth and death, this algorithm step identifies prototype observations of each cluster. The second step of *ClusterMPP* is the K nearest neighbors (KNN) assignment that affects the remaining observations to the corresponding clusters. We experiment *ClusterMPP* on several complex and scalable microarray datasets. The results show the efficiency of *ClusterMPP* compared to well-known microarray data clustering methods like K-means, Spectral Clustering, and Mean-Shift.

K. Henni

Department of Computer Science, University of Sciences and Technologies Oran “Mohamed Boudia” USTO-MB, Oran, Algeria

e-mail: khadidja.henni@univ-usto.dz

O. Alata

Hubert Courien Laboratory, UMR 5516, Jean Monnet University, Saint-Étienne, France

A. El Idrissi

LabTIC Laboratory, ENSA-Tangier, Tangier, Morocco

B. Vannier

Receptors, Regulation and Tumor Cells (2RCT), University of Poitiers, Poitiers, France

L. Zaoui

LSSD Laboratory, Department of Computer Science, University of Science and Technology, Oran, Algeria

A. Moussa (✉)

SDET Team, ENSA-Tangier, Tangier, Morocco

e-mail: amoussa@uae.ac.ma

1 Introduction

With the recent advances of biomedical technology, a lot of “OMICS” data from genomic, transcriptomic, and proteomic domain can now be collected quickly and cheaply. Microarray technology analyzes genome-wide gene expression patterns and is used in many areas including biotechnology, pharmacology, medicine, and environment.

Identifying the patterns hidden in gene expression data allows an enhanced understanding of functional genomics. However, the large number of genes and their complexity make difficult comprehending and interpreting the resulting mass of data. Therefore, high-throughput expression profiling exploitation requires advanced analysis tools to extract knowledge from the huge amount of data [8].

When experimental conditions are not known, an unsupervised treatment is recommended. Clustering is a powerful unsupervised technique which needs no a priori information, it helps to understand gene function, gene regulation, cellular processes in cell subtypes [8, 14]. Clustering aims to organize similar data into clusters in which data are similar to each other. This data can be genes, samples, or both genes and samples simultaneously (two-way clustering).

K-means [9], Self Organizing Map (SOM) [8], and Hierarchical clustering [8, 13] are widely used in gene expression analysis field. But they are unable to deal with noise, high dimensionality, complexity, and nonlinear separability associated with microarray gene expression data. K-means [9, 13] is a fast and simple algorithm, but it is sensible to initialization and number of clusters, it converges to local minima. SOM is a very used clustering algorithm with a visual output, but suffers from strong initialization dependency, outputs instability, and is powerless in unbalanced classes' cases. Hierarchical clustering [6, 13] is sensible to data modification, noise, and to outliers; they are also powerless over unbalanced classes and convex shapes. The biclustering algorithms (two-way clustering) [8, 14] are also widely used to cluster simultaneously gene and expression. They are organized in four families: (1) variance minimization methods, (2) two-way clustering methods, (3) motif and pattern recognition methods, and (4) probabilistic and generative approaches.

In gene expression data analysis, the quality and robustness of clustering results is crucial. Thus, the choice of clustering algorithm should be quite selective and depends on several issues such as: the separation of genes or samples whatever the complexity and overlapping level of clusters, the result accuracy, its stability (results do not depend on initialization nor parameters), its scalability, and its ability to handle noise.

Today, a huge number of clustering algorithms are available [8], with an impressive practical performance and desirable theoretical grantees. However, many of them are not able to cover all recent applications needs. Clustering algorithms must be multi-objective and cover all clustering issues, they should consider all existing highlights and minimize their disadvantages. Meanwhile, proposing a new clustering algorithm has become a multi-objective optimization problem. The solution is to identify all issues and find a way to take them all into account.

To this aim, we propose a new unsupervised clustering algorithm called “Clustering by Marked Point Process (*ClusterMPP*),” it is a multi-objective algorithm able to deal with cited issues. This algorithm belongs to the density-based algorithms family [10] and it is based on a probabilistic model called Marked Point Process (MPP) [1, 15, 16], used in imaging field to mark and to detect geometrical objects. The main idea of *ClusterMPP* is: first seeking cluster patterns to establish a classification model. To ensure that, it defines cluster as a dense region and simulates a proposed MPP to locate objects (hyper-spheres) on these regions of interest. It gives rise, kills, moves, and resizes objects, under interaction constraints, in attempt to place them in dense regions also called clusters.

At the end of this iterative object manipulating process done by an adapted Reversible Jump Markov Chain Monte Carlo (RJCMC) [7], objects will be located on clusters. They will delimit cluster fundamental area (cluster mode) by their overlapping (connected component). Mostly, objects do not cover all available data, and covered data are named “Prototypes”. The second step of *ClusterMPP* is to assign remaining data (data not covered by objects) to the detected clusters using an improved version of KNN algorithm [4].

In this chapter, Sect. 2 presents a background on clustering problem, introduces some definitions of the MPP theory, and describes a proposed MPP model. Section 3 gives details about “ClusterMPP” implementation. The performance of this clustering algorithm is demonstrated in Sect. 4 using benchmarks of microarray databases.

2 Background and Marked Point Process Model

Let us define the observed data (genes, samples) as multidimensional field Y , with $y = \{y_q\}_{q=1, \dots, Q}$, Q is the number of observations and $y_q = [y_{q,1}, \dots, y_{q,M}]^T \in \chi \subseteq \mathbb{R}^M$. M is the space dimension (number of conditions).

This study aims to organize the Q observations (genes, samples) in k clusters, where a cluster is a set of observations sharing similar M conditions. Each cluster groups genes with similar expression patterns (co-expressed genes) or samples based on the corresponding expression profiles. Clusters have a biological significance. They may further understanding many gene functions, reveal the similar cellular functions or sub-cell types which are hard to identify by traditional morphology-based approaches [8].

This purpose defines cluster by their modes; modes delimit the domains of high local observation concentration, called after prototypes. These cluster modes are assumed to be randomly distributed in the multidimensional raw data space, they reflect carefully the observation patterns inside the raw data space. Thus, the problem comes down to modes seeking by capturing the cumulative density distribution of observations in χ .

2.1 Definitions and Notations

MPP is a random process which models point patterns where the points are mainly positions or centers of geometrical marks in a multidimensional space [1, 15, 16]. If there is no interaction between points and no mark, the process is called Poisson Point Process and it plays a fundamental role in the definition of probability density functions of more advanced point processes [1]. When processes use neighboring relations between points, they belong to the family of Markov Point Processes. This kind of processes was used in statistical physics, under the name of Gibbs processes (see [1–3] for more details).

Let X be an MPP living in a finite simple point process, it is composed of finite random configurations of points; these points are positions of marks; they are chosen randomly from a compact subspace $\chi \subseteq \mathbb{R}^M$ (χ has a finite measure: $\mu(\chi) < \infty$ with μ the Lebesgue measure). In this paper, a realization of X is a set of hyper-spheres $x = \{x_l\}_{l=1, \dots, n(x)}$ (also called a configuration). Each hyper-sphere x_l is defined by its center $c_l \in \chi$ (a point) and its radius $r_l \in [r_{\min}, r_{\max}]$ (its mark) which also defines the object neighboring. $n(x)$ is the number of hyper-spheres in the configuration x . A hyper-sphere (or object or marked point) will be denoted $x_l(c_l, r_l)$ and the configuration of points of x , $p_x = \{c_l\}_{l=1, \dots, n(x)}$. The spherical shape is flexible and easy to adapt in the multidimensional space.

The probability density functions $f(x)$ of the MPP are defined by the reference to the Poisson process. They can be expressed as a Gibbs distribution with an energy $U(x)$ [1]:

$$f(x) \propto e^{U(x)} \quad (1)$$

We assume now that a subset of observations constitutes the positions of a realization of the MPP X ; this subset will be the initial set of cluster prototypes. The object of the proposed clustering algorithm is to find this subset by selecting the observations with an associated set of marks that maximize the probability density function of the MPP. This optimal object configuration, \hat{x} , is:

$$\hat{x} = \operatorname{argmax}_{x, p_x \subseteq \mathcal{X}} f(x) \quad (2)$$

The density f [see Eq. (1)] is related to a normalizing constant which cannot be calculated. A typical solution for this drawback is the use of Markov Chain Monte Carlo (MCMC) simulator. Several MPP simulators have been proposed in the literature [1, 15], but the one that has been mainly used these twenty last years is the RJMCMC algorithm [7]. This is due to the good convergence speed, its low computational time, and its flexibility. The RJMCMC samplers can employ several movements and use the Green's ratio (*GR*) [7] to accept or decline them. The *GR* depends on the move probability and the process state before and after the move application.

2.2 Proposed Model

The probability density function of the MPP is proportional to the exponential of a Gibbs energy function [see Eq. (1)]. The energy of MPP is classically the sum of two terms [1, 3]:

$$U(x|\theta) = U_{data}(x|\theta) + U_{inter}(x|\theta) \quad (3)$$

where θ is the set of model parameters. $U_{data}(x|\theta)$ is the data driven energy and $U_{inter}(x|\theta)$ is the internal energy.

The Data Driven Energy $U_{data}(x|\theta)$ represents relationships between objects and observed data. For our purpose, objects are placed in high density regions. This energy is the sum of local contributions of each object x_i (detector model):

$$U_{data}(x|\theta) = \sum_{x_i \in x} V(x_i) \quad (4)$$

where $V(x_i)$ is the potential function of hyper-sphere $x_i(c_i, r_i)$.

In order to obtain realizations with objects localized in high concentration areas, this function must favor the acceptance of well-positioned objects. Let us define $no(x_i)$, the number of covered observations by $x_i(c_i, r_i)$ and $do(x_i)$ the density of observations inside $x_i(c_i, r_i)$. We consider that an object x_i is well positioned if it satisfies two criteria: $no(x_i) > n_{\min}$ and $do(x_i) > d_{\min}$, where n_{\min} and d_{\min} belong to the set of model parameters θ .

Therefore, $V(x_i)$ can be expressed as follows:

$$V(x_i) = \begin{cases} no(x_i) & \text{if } no(x_i) > n_{\min} \ \& \ do(x_i) > d_{\min} \\ -no(x_i) - v_{\max} & \text{otherwise.} \end{cases} \quad (5)$$

where v_{\max} is a high value that allows to greatly reduce the probability of a configuration which contains badly positioned objects. In order to strongly penalize objects that are not correctly localized, v_{\max} must be a high value (equal to 1000, for example).

The Internal Energy $U_{inter}(x|\theta)$ interactions between objects are modeled by the use of potential functions which are chosen according to a priori information about searched configurations. Thus, two basic rules are imposed: (1) estimate the exact number of connected components by driving the creation of connected objects to the observed data and their dispersion in space. (2) Prevent the object overlapping phenomenon which leads to exponential growth of objects number. According to those considerations, we propose the following internal energy:

$$U_{inter}(x|\theta) = n(x) \log \beta - |co(x)| \log \gamma + n_v \log \delta \quad (6)$$

where $\beta > 0$ is the point process intensity [15]. The two first terms of this internal energy are those of the connected component process [2] with $co(x)$ the set of connected objects in x , $|co(x)|$ the number of connected components in x , and $\gamma > 0$ the interaction parameter. The connected component process is chosen regarding to the first constraint previously written. In the internal energy, each connected component defined by x , $co_i \in co(x)$, $i = 1, \dots, |co(x)|$ has a contribution equal to $n(co_i) \log \beta - \log \gamma$ with $n(co_i)$ the number of hyper-spheres contained in co_i . See [15] for more details.

The third term of the internal energy is inspired from the pairwise point process such as defined in [15], and is used to penalize the hyper-sphere tangle. This term is based on the definition of the following neighboring relation, for $x_i \in x$ and $x_j \in x$, $i \neq j$:

$$x_i \sim x_j \text{ if } d(c_i, c_j) < (r_i + r_j)/5 \quad (7)$$

the denominator value was chosen experimentally. This third term is also based on the following interaction potentials of second order:

$$\phi(x_i, x_j) = \begin{cases} \log \delta & \text{if } x_i \sim x_j \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The contribution of this term in the internal energy is written as $\sum_{(x_i, x_j), i < j} \phi(x_i, x_j) = n_v \log \delta$ with n_v the total number of neighboring relations in x and $\delta \in [0, 1]$. In the following we will choose δ near to 0 (such that $\log \delta = -1000$, for example) in order to strongly penalize overlapping objects.

Following the definition of the data driven energy and the internal energy, it is now possible to define the set of parameters associated with the proposed MPP:

$$\theta = \{\beta, \gamma, n_{\min}, d_{\min}, r_{\min}, r_{\max}\}. \quad (9)$$

with r_{\min} and r_{\max} , the minimum and maximum radius of the hyper-spheres, respectively.

3 ClusterMPP: A New Clustering Algorithm

ClusterMPP is a new density-based clustering algorithm of microarray datasets able to cover all main clustering requirements. This algorithm is composed of two steps:

- Cluster modes detection: is a simulation of the proposed MPP to capture and delimit the high density regions.
- Classification performing: is a classification process finalization, it classifies the remaining observations to the detected cluster modes.

3.1 Cluster Modes Detection

This section presents the mode detection algorithm (the first step of *ClusterMPP*). It is a sampling algorithm designed to discover regions of high observation concentration by simulating the proposed MPP model (see Sect. 2.2). The proposed mode detection algorithm is a variant of RJMCMC algorithm, it is an iterative and three-step procedure that generates an MPP configuration at each iteration. In what follows, we will write: x^i , $i \geq 0$, the configuration of objects at the i th iteration of *ClusterMPP*. p_{birth} , p_{death} , p_{disp} , and p_{mchg} are, respectively, movement probabilities choose: birth, death, move, or changing radius of an object.

MPP Initialization In this step, *ClusterMPP* generates an initial configuration $x^0 = \{x_l\}_{l=1, \dots, n(x^0)}$: an objects set which cover all available observations (see Algorithm 2). Parameter estimation() is a function which estimates parameter values as follows:

- β is the intensity parameter of process. The desired number of objects in the proposed MPP realizations depends on observed data, we propose to fix it equal to the cardinality of y ($n(y)$).
- γ is the interaction parameter of a connected component Markov Point Process. γ can be chosen equal to the variance of the observed data. Here we focus on multidimensional datasets, we use the total variance (the trace of the variance-covariance matrix, i.e., the sum of variances).

Algorithm 2 Initialization

Input: Observed data $y = \{y_1, \dots, y_{n(y)}\}$, r_{\max} .
Output: $x^0 = \{x_1, \dots, x_{n(x^0)}\}$, θ .

- 1: $x_1 = (y_1, r_{\max})$
- 2: $x^0 = \{x_1\}$
- 3: $n(x^0) = 1$
- 4: $j = 1$
- 5: **while** ($j \leq n(y)$) **do**
- 6: $l = 1$
- 7: **while** ($l \leq n(x^0)$) **do**
- 8: **if** ($d(y_j, c_l) > r_{\max}$) **then**
- 9: $n(x^0) = n(x^0) + 1$
- 10: $x_{n(x^0)} = (y_j, r_{\max})$
- 11: $x^0 = x^0 \cup \{x_{n(x^0)}\}$
- 12: **end if**
- 13: $l = l + 1$
- 14: **end while**
- 15: $j = j + 1$
- 16: **end while**
- 17: $\theta =$ Parameter estimation()

- n_{\min} and d_{\min} are the parameters defining the data driven energy, they are calculated from the initial configuration and equal to the average number of covered points by hyper-spheres and the average density of objects inside hyper-spheres, respectively.
- r_{\min} and r_{\max} describe the object scales: the first one is the minimum value of object radii. We propose to fix it equal to the minimum distance between points in y , in order to have objects that cover at least two observations. r_{\max} is the maximum of the object radii, it is estimated by a learning stage.

MPP Simulation *ClusterMPP* repeats the following steps, until the stabilization of the number of objects and the process energy: at the $(i + 1)$ th iteration a random draw to select one of the four movements: birth, death, moving, or changing marks. In the *birth* movement, a new object $\omega(c_\omega, r_\omega)$ is created by drawing randomly a center from y and choosing randomly a radius in $[r_{\min}, r_{\max}]$. Next, ω will be added to the configuration $\tilde{x}_b = x^i \cup \{\omega\}$. The *death* is performed if the configuration x^i contains at least one object ($n(x^i) > 0$). The movement is simulated by selecting an object ω randomly from the current configuration x^i . Then ω is removed from x^i . The proposal configuration in a death case becomes $\tilde{x}_d = x^i \setminus \{\omega\}$. *Move* and *changing marks* movements are performed if the configuration x^i contains at least one object ($n(x^i) > 0$). Movements are simulated by selecting an object ω randomly from the current configuration x^i . Choosing randomly a new center $c_{\tilde{\omega}}$ from the observed data field y (*Moving* case) or a new radius $r_{\tilde{\omega}}$ from the interval $[r_{\min}, r_{\max}]$ (*changing marks* case). Then, ω is replaced by $\tilde{\omega}$, $\tilde{x} = \{x^i \setminus \{\omega\}\} \cup \{\tilde{\omega}\}$.

In order to accept or reject movement, the algorithm computes the *GR* for each move (see Table 1). This iterative algorithm step manipulates the MPP by applying different movements on this process objects. It shifts the MPP from the initial configuration in which objects cover all observations to the configurations where objects move toward the desired regions, the connected components give the searched modes. Details of the MPP simulation step are given in Algorithm 3.

Mode Extraction The final configuration will contain objects located in regions of high concentration of observed data. Thus, the connected components of objects give the searched modes. For each component, *ClusterMPP* extracts all covered observations (prototypes) and assigns them by trivial way to the corresponding clusters. The remaining observations are non-prototypes, they will be classified in the second step of *ClusterMPP*.

Table 1 Green's ratio [7]

Movements	Birth	Death	Move or changing marks
Green's ratio	$GR^{Birth} = \frac{p_{death} f(\tilde{x}_b) \mu(\chi)}{p_{birth} f(x^i) n(x^i) + 1}$	$GR^{Death} = \frac{p_{birth} f(\tilde{x}_d) n(x)}{p_{death} f(x) \mu(\chi)}$	$GR^{disp/mchg} = \frac{f(\tilde{x})}{f(x)}$

Algorithm 3 MPP simulation algorithm

Input: Observed data $y = \{y_1, \dots, y_{n(y)}\}, x^0, \theta$.
Output: x^f .

- 1: $i = 0$
- 2: **while** not converged **do**
- 3: \tilde{x} =Movement simulation (y, x^i, θ)
- 4: Compute density $f(\tilde{x})/f(x^i)$ using Eq. (1) and Eq. (3).
- 5: Compute the corresponding Green's ratio GR (Table 1).
- 6: $\alpha = \min(1, GR)$
- 7: Draw $accept \sim U_{(0;1)}$
- 8: **if** $accept < \alpha$ **then**
- 9: $x^{i+1} = \tilde{x}$
- 10: **else**
- 11: $x^{i+1} = x^i$
- 12: **end if**
- 13: $i = i + 1$
- 14: **end while**

3.2 Classification Performing

Prototype observations will be directly assigned to the corresponding clusters. They are mostly well classified in the first stage of “ClusterMPP”. So, the problem lies basically in the classification of the remaining observations (non-prototype observations). In order to classify all observations, we propose to use an improved version of the KNN algorithm [4], which assigns non-prototype observations one by one to the nearest cluster, in a specific order, respecting their distances to the prototype observations. *ClusterMPP* detects different prototype observations and assigns non-prototype observations to the corresponding clusters.

4 Experiments

Distance functions are an important factor of clustering procedures. They measure the similarity between two observations. In this chapter, *ClusterMPP* uses the Euclidean distance which measures the geometric relation between two vectors (the generalization of other distances poses no particular problem). *ClusterMPP* is compared with three well-known algorithms *Mean-Shift*, *Spectral Clustering*, and *K-means*. Note that *K-means* [9] and *Spectral Clustering* [12] require the number of clusters K . *Mean-Shift* [17] requires the use of an appropriate value of bandwidth. *ClusterMPP* does not need a priori knowledge and parameters are chosen from observed data. All five algorithms were tested with six benchmarks of microarray datasets [5], used to validate performance of clustering algorithms (see Table 2).

Table 2 Datasets description [5]

Datasets	Rat CNS	Leukemia	Lymphoma	NCI60	Novartis	Yeast cell cycle
Samples	17	100	100	200	1000	72
Attributes	112	38	80	57	103	698
Classes	6	3	3	8	4	5

Assessment methods of clustering algorithms measure how well a computed clustering solution agrees with the gold solution [5] for the given dataset, where the gold solution is a known data partition. We propose to use Receiver Operating Characteristic curve (ROC) [5] and External validation indexes (Balanced Misclassification Index [5] (BMI), Rand index, Jaccard coefficient (JC), Folkes and Mallows index (FM)) [11] to evaluate the performance of *ClusterMPP* algorithm by comparing it with other algorithms.

Receiver Operating Characteristic Curve (ROC) is a graphical technique to compare classifiers and visualize their performance. ROC plane maps the True Positive Rate TPR (sensitivity) versus False Positive Rate FPR (specificity) [5].

External Validation Indexes different external validation indexes have been used: (1) BMI [5] compares the performance of different clustering algorithms by measuring their ability to capture the structure in a dataset. BMI uses the misclassification error rate and the balancing error rate (the average of the errors on each clusters [5]). The BMI index takes values between 0 and 1 and needs to be minimized. (2) RS, JC, and FM [11], these indexes are computed by four terms which indicate if a pair of points in both solutions (gold and resulting solution) share the same cluster. These indexes need to be maximized.

Figure 1 shows the partition in the ROC plane for each considered algorithm and for each dataset. We plot different *ClusterMPP* results obtained by varying the value of r_{max} parameter, and the best result of the other algorithms. Figures 2, 3, and 4 display the comparison of the classification error rates and the external validation indexes (BMI, RS, JC, and MF). *ClusterMPP* has superior performance against *Mean-Shift* and *Spectral Clustering* even in the worst cases. *ClusterMPP* outperforms *K-means* on most datasets and its strength is the estimation of its important parameter (r_{max}) through a learning step. However, *K-means* requires the number of clusters k , which can be computed by several methods like the Bayesian Information Criterion (BIC) [9].

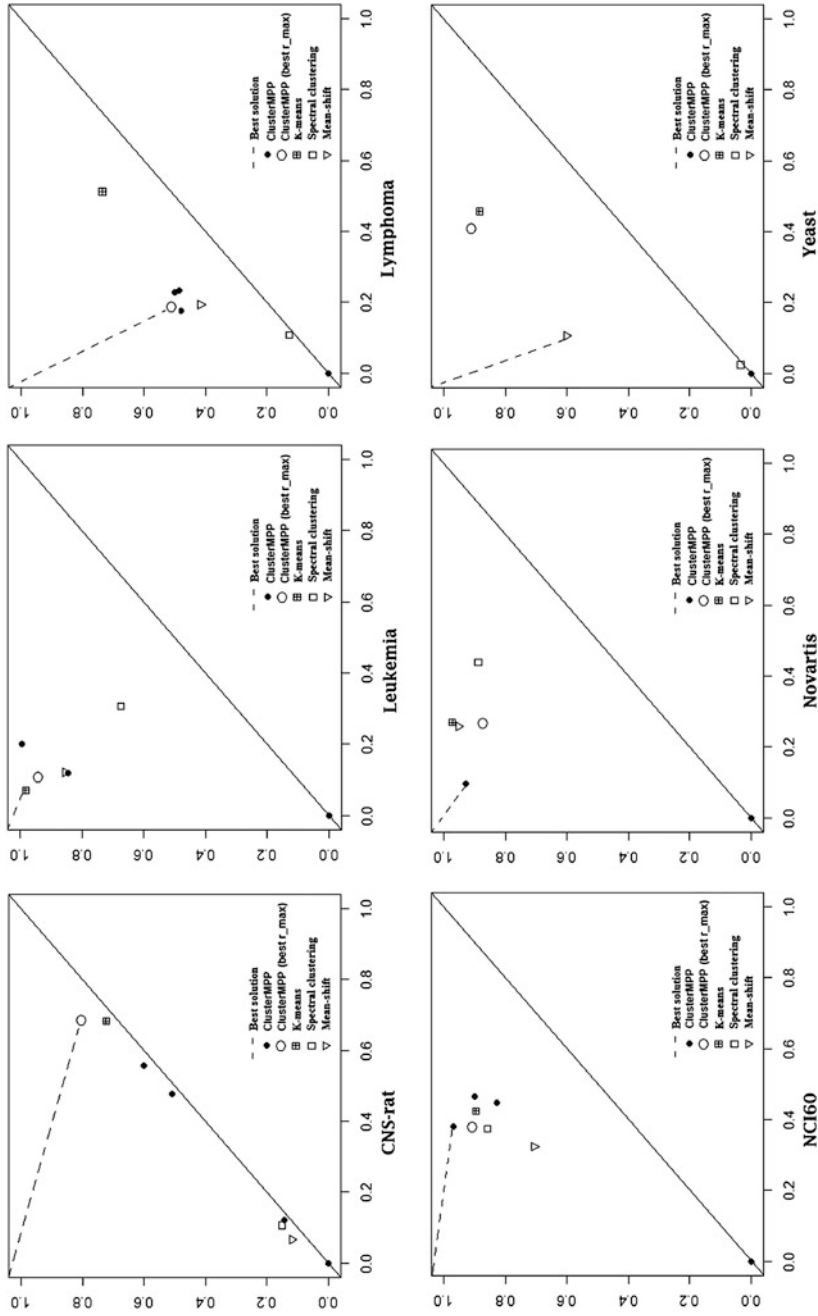


Fig. 1 ROC curves for each dataset. Each sub-figure refers to a dataset and each position in the ROC curves refers to a result solution of each algorithm (*ClusterMPP*, *K-means*, *Mean-Shift*, and *Spectral Clustering*)

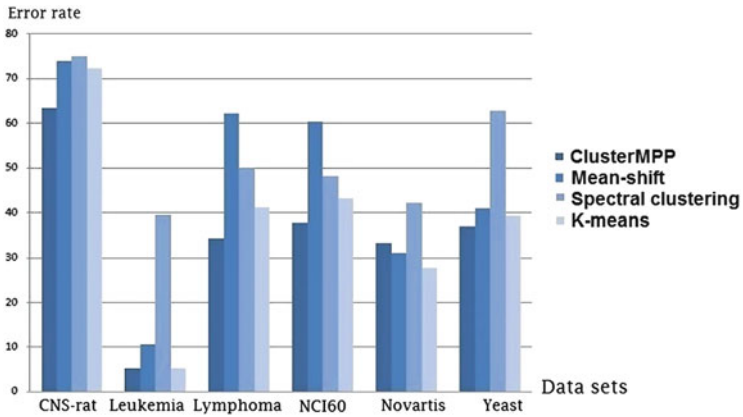


Fig. 2 Comparison of classification results based on error rates

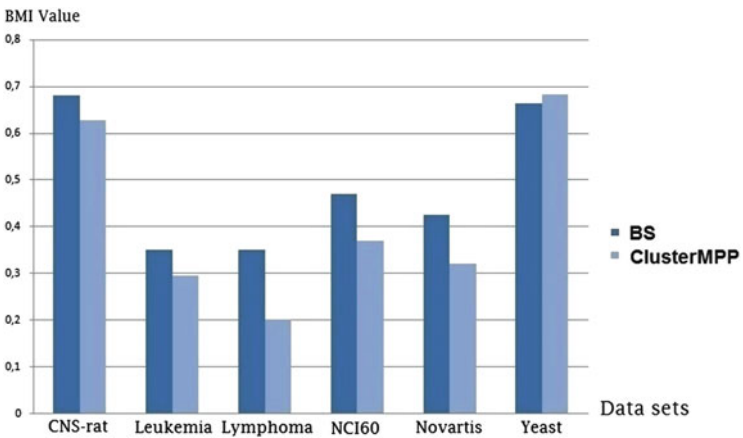


Fig. 3 Comparison of classification results based on BMI values (BS denotes the best solution obtained in [5])

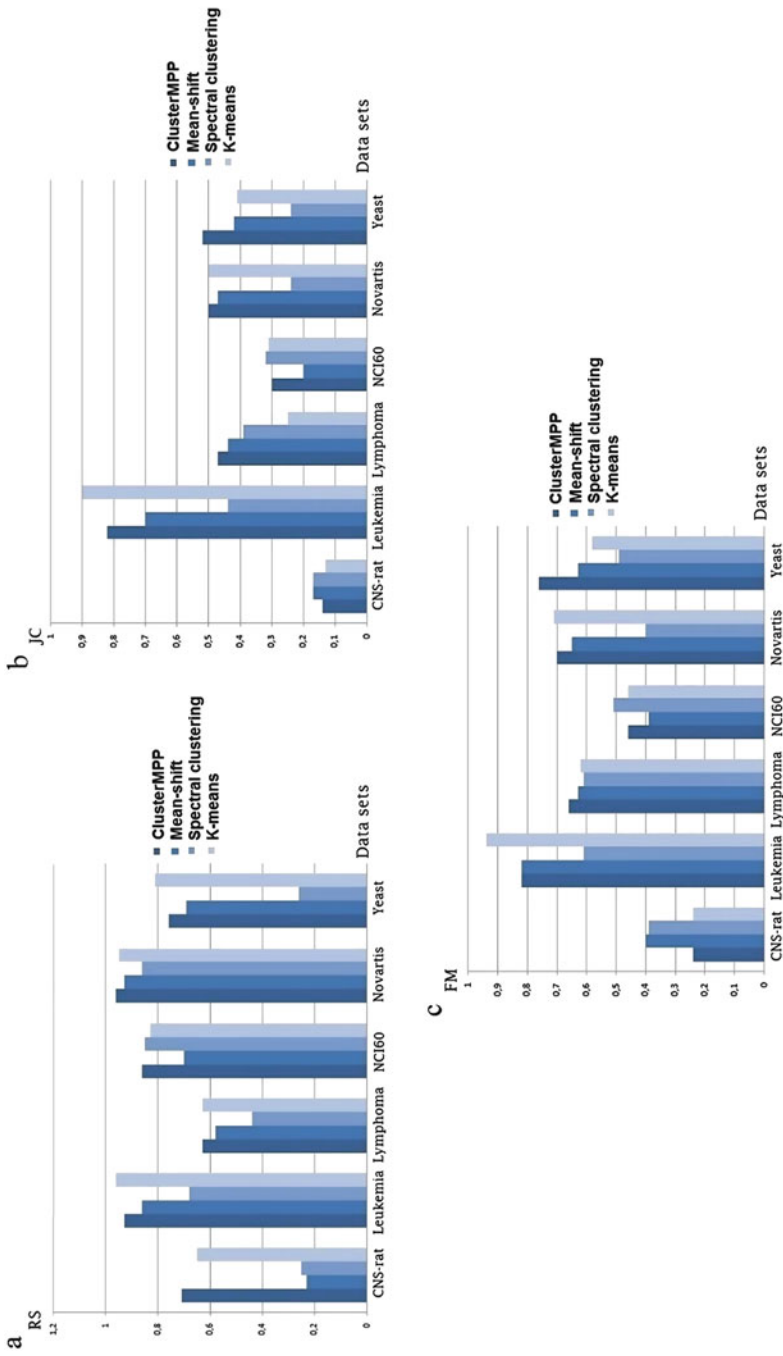


Fig. 4 Comparison of classification results using the external validation indexes: **(a)** Rand static index, **(b)** Jaccard coefficient index, **(c)** Fowlkes and Mallows index

5 Conclusion

This work was intended to describe a new unsupervised clustering algorithm belonging to density-based family. It also implements a probabilistic technique, which makes it able to solve the clustering problem taking into account different issues. The algorithm seeks cluster modes by the simulation of proposed MPP and use an improved KNN version to finalize the classification process. *ClusterMPP* outperforms the other clustering algorithms. In the future, we will integrate ontological information about genes as an a priori information to improve clustering process of biological data.

Acknowledgements This research was supported in part by the Erasmus Mundus—Al Idrisi II program.

References

1. Alata, O., Burg, S., Dupas, A.: Grouping/degrouping point process, a point process driven by geometrical and topological properties of a partition in regions. *Comput. Vis. Image Underst.* **115**(9), 1324–1339 (2011)
2. Chin, Y.C., Baddeley, A.J.: Markov interacting component processes. *Adv. Appl. Probab.* **32**(3), 597–619 (2000)
3. Clifford, P.: Markov random fields in statistics. In: Grimmett, G.R., Welsh, D.J.A. (Eds.) *Disorder in Physical Systems, A Volume in Honour of J.M. Hammersley*, pp. 19–32. Clarendon Press, Oxford (1990)
4. Ferrandiz, S., Boullé, M.: Bayesian instance selection for the nearest neighbor rule. *Mach. Learn.* **81**(3), 229–256 (2010)
5. Giancarlo, R., Bosco, L., Pinello, G.L., Utro, F.: A methodology to assess the intrinsic discriminative ability of a distance function and its interplay with clustering algorithms for Microarray data analysis. *BMC Bioinformatics* **14**(S-1), S6 (2013)
6. Gorunescu, F.: *Data Mining: Concepts, Models and Techniques*. Intelligent Systems Reference Library, vol. 12, pp. 1–43. Springer, Berlin (2011)
7. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711–732 (1995)
8. Harun, P., Burak, E., Andy P., Çetin, Y.: Clustering of high throughput gene expression data. *Comput. Oper. Res.* **39**(12), 3046–3061 (2012)
9. Kaur, S., Kaur, U.: A survey on various clustering techniques with K-means clustering algorithm in detail. *Int. J. Comput. Sci. Mob. Comput.* **2**(4), 155–159 (2013)
10. Khaled, S.: TOBAE: a density-based agglomerative clustering algorithm. *J. Classif.* **32**(2), 241–267 (2015)
11. Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J.: Understanding of internal clustering validation measures. In: *ICDM-10 Proceedings of the 2010 IEEE International Conference on Data Mining*, pp. 911–916 (2010)
12. Mouysset, S., et al.: Spectral clustering: interpretation and Gaussian parameter. In: *Data Analysis, and Knowledge Organization. Studies in Classification*, vol. 4, pp. 153–162 (2013)
13. Reddy, C.K., Vinzamuri, B.: A survey of partitional and hierarchical clustering algorithms. In: Aggarwal, C., Reddy, C.K. (eds.) *Data Clustering: Algorithms and Applications*, pp. 87–110. CRC (2014)

14. Sepp, H., et al.: FABIA: factor analysis for bicluster acquisition. *Bioinformatics*. **26**(12), 1520–1527 (2010)
15. Stoica, R.S., Gay, E., Kretzschmar, A.: Cluster pattern detection in spatial data based on Monte Carlo inference. *Biom. J.* **49**(4), 505–519 (2007)
16. Stoica, R.S., Martinez, V.J., Saar, E.: Filaments in observed and mock galaxy catalogues. *Astron. Astrophys.* **510**(38), 1–12 (2010)
17. Wu, K.L., Yang, M.S.: Mean shift-based clustering. *Pattern Recogn.* **40**(11) 3035–3052 (2007)