

# Mass Distributed Clustering: A New Algorithm for Repeated Measurements in Gene Expression Data

Shinya Matsumoto<sup>1,\*,\dagger</sup>  
shinya.matsumoto@ncr.com

Ken-ichi Aisaki<sup>2,\*</sup>  
aisaki@nihs.go.jp

Jun Kanno<sup>2,\*,\ddagger</sup>  
kanno@nihs.go.jp

<sup>1</sup> Teradata Division, NCR Japan, Ltd. 2-4-1 Shiba-koen, Minato-ku Tokyo 105-0011, Japan

<sup>2</sup> Cellular & Molecular Toxicology, Biological Safety Research Center, National Institutes of Health Sciences, 1-18-1 Kamiyoga, Setagaya-ku Tokyo 158-8501, Japan

## Abstract

The availability of whole-genome sequence data and high-throughput techniques such as DNA microarray enable researchers to monitor the alteration of gene expression by a certain organ or tissue in a comprehensive manner. The quantity of gene expression data can be greater than 30,000 genes per one measurement, making data clustering methods for analysis essential. Biologists usually design experimental protocols so that statistical significance can be evaluated; often, they conduct experiments in triplicate to generate a mean and standard deviation. Existing clustering methods usually use these mean or median values, rather than the original data, and take significance into account by omitting data showing large standard deviations, which eliminates potentially useful information. We propose a clustering method that uses each of the triplicate data sets as a probability distribution function instead of pooling data points into a median or mean. This method permits truly unsupervised clustering of the data from DNA microarrays.

**Keywords:** data mining, bioinformatics, gene expression data, microarray, repeated measurements, clustering algorithm

## 1 Introduction

### 1.1 Motivation

When large-scale gene expression profiles became available, biologists usually normalized the data to overt biological events, such as monitorable phenotypes. By doing this, biologically important expression data could be selected by linkage analysis to a particular biological events and used for further analysis. This type of analysis tends to be limited to genes that encode the final phases of a gene cascade or signaling system that directly reflects an emergence of phenotype, and hence shows high expression values.

The advent of microarray and other high-throughput technologies has removed such limitations, allowing whole-genome analysis that includes the initial phases of the cascade, where phenotypes are not clear and signal intensity is usually low. These technologies generate huge quantities of data, placing great demands on data analysis. To accommodate this demand, the Division of Cellular and Molecular Toxicology of National Institute of Health Sciences (NIHS), Japan, has developed the Percellome System [2], which generates absolute mRNA-quantity data as the copy number per cell from the microarray system and quantitative PCR. This system essentially enables utilization of all

---

\*These authors contributed equally to this work.

<sup>\dagger</sup>To whom correspondence about mathematical issues should be addressed: E-mail: shinya.matsumoto@ncr.com

<sup>\ddagger</sup>To whom correspondence about biological issues including Percellome system should be addressed: E-mail: kanno@nihs.go.jp

Table 1: Sample data.

	Condition 1			Condition 2		
	1st Exp.	2nd Exp.	3rd Exp.	1st Exp.	2nd Exp.	3rd Exp.
Gene 1	0.9682	0.9924	1.0394	-0.1277	-0.0842	0.2125
Gene 2	1.3656	1.4547	1.3798	-0.2026	-0.2539	0.4596
Gene 3	-0.0109	-0.0619	0.0738	0.9116	0.9532	1.1352
Gene 4	-0.1315	-0.0222	0.1540	1.3569	1.2596	1.5835
Gene 5	-1.1195	-0.9738	-0.9067	0.0605	0.0946	-0.1543
Gene 6	-1.4476	-1.2152	-1.5372	0.0088	0.0508	-0.0587
Gene 7	-0.0070	0.0697	-0.0623	-0.8928	-1.0297	-1.0775
Gene 8	-0.1236	-0.2152	0.3397	-1.3814	-1.3456	-1.4730
Gene 9	1.2004	0.0041	1.0455	-0.2224	0.5194	0.4527
Gene 10	0.1282	0.4077	0.2144	1.1292	0.4488	0.6720
Gene 11	-1.2166	0.2551	0.2115	1.3180	0.7994	0.1325
Gene 12	-0.5777	-0.7242	-0.9481	-0.0263	0.1748	0.6009
Gene 13	-0.2747	-0.3692	-1.6061	-0.8657	-0.0627	0.1783
Gene 14	0.6377	0.1786	-1.5665	-1.2766	-0.7981	-0.1753
Gene 15	-1.1518	0.9327	0.9700	-1.2910	-0.8788	-0.0802
Gene 16	0.0885	1.7689	0.3925	0.0623	0.8591	-1.6715
Gene 17	1.3529	1.8681	-1.7204	1.8635	0.2069	-0.5710
Gene 18	0.7227	0.0423	0.7346	-0.8883	-0.1600	-0.4517
Gene 19	-1.4129	-0.2668	0.1797	1.0153	2.5328	-2.0493
Gene 20	-0.2813	-0.6737	-0.5450	0.4369	-1.0448	-0.8920
Gene 21	-0.2935	2.4749	3.2186	-1.8833	1.6711	0.2152
Gene 22	-0.3168	0.3275	-5.4107	2.0824	0.9931	-3.0695
Gene 23	0.0022	0.1320	-0.1333	2.6916	-0.3704	3.0789
Gene 24	-3.1931	-0.6846	3.8781	-1.8794	-2.8393	-0.6813

of the gene expression data for the clustering analysis. The basis of the clustering strategy for this all-gene data is a phenotype-independent analysis, meaning that there are no auxiliary data that can be used for clustering. We have designed a pure, unsupervised clustering system that can handle low-intensity data along with its variance. It is postulated that low-intensity data may contain relatively larger amounts of measurement error than high-intensity data.

An example of the data collected for this study is shown in Table 1. Two experiments were performed in triplicate (i.e., each experiment was performed on three mice). An average result for each replicated experiment can be calculated, but this eliminates information about any deviations. Alternatively, the data from the three replicates can be treated as a probability distribution and handled by a parametric approach. Applying this approach to gene expression data, we were able to develop our unsupervised clustering algorithm, mass distributed clustering (MADIC).

## 1.2 Related Works

Most clustering algorithms ignore measurement errors. However, measurement errors occur in the real world, especially in gene expression analysis. NIHS has established a measurement method that can analyze all genes, including low-intensity genes that contain substantial measurement errors [2]. Some clustering algorithms, such as the one presented by Kumar et al., can handle data with errors [3]. Yeung et al. demonstrated clustering algorithms in which deviations from repeated measurements can be evaluated [6]. Using a clustering algorithm with SD- or CV-distance, their approach was an improvement over the traditional simple average method, which does not evaluate deviations.

Many clustering algorithms have been proposed for gene expression analysis [4, 5], but existing algorithms cannot use whole genes, stable and unstable genes. Such methods are useful when clustering stable objects, but we wanted to devise a method to cluster both stable and unstable genes. Our proposed algorithm is based on an extension of density-based clustering, DBSCAN [1].

## 1.3 Purpose of Research

The purpose of this research was to develop a clustering algorithm that would handle triplicate gene expression data without losing information about deviation.

## 1.4 Outline of Article

In Section 2, we define how we have extended density-based clustering and how our proposed algorithm differs from those used in conventional clustering. In Section 3, we provide the details of our clustering algorithm. In Section 4, we present results of experiments with synthetic and real gene data. Finally, in Section 5 we offer conclusions.

## 2 Definitions

### 2.1 Notations

(1) Data set.  $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_n\}$ : Data set for clustering.  $\mathbf{o}_i$  is a probability distribution function (PDF) in  $d$ -dimensional Euclidean space. We sometimes denote this by the objects  $\mathbf{o}, \mathbf{p}, \mathbf{q} \in \mathbf{O}$ .  $\mathbf{o}_i := p_{\sigma_i}(x_i) : \mathbf{R}^d \rightarrow \mathbf{R}^+ : \text{PDF}$ .  $\mathbf{x}_i \in \mathbf{R}^d$  is a point of  $d$ -dimensional Euclidean space.  $\sigma_i$  is the parameter of the PDF. We also represent the PDF in another way. We can use this notation if the PDF has no special direction for this integration.  $p_{\mathbf{x}_i\sigma_i}(\mathbf{r}) : \mathbf{R}^+ \rightarrow \mathbf{R}^+ : \text{PDF}$ .  $\mathbf{r}$  is the distance from  $\mathbf{x}_i$ .

(2) Observation.  $\mathbf{y}(i, j, k) : k$ th observed value for  $j$ th dimension for  $\mathbf{o}_i$ .

(3) Distance. Distance is denoted by  $\text{dist}(\mathbf{x}_i, \mathbf{x}_j)$  in usual Euclidean space. We define distance between objects as  $\text{dist}(\mathbf{o}_i, \mathbf{o}_j) = \text{dist}(p_{\sigma_i}(\mathbf{x}_i), p_{\sigma_j}(\mathbf{x}_j)) = \text{dist}(\mathbf{x}_i, \mathbf{x}_j)$ .

(4)  $\varepsilon$ . We use  $\varepsilon$  for the threshold about distance.

(5)  $\theta_m$ . We use  $\theta_m$  for threshold about mass.

(6) Mass function. We defined the mass function as:

$$\frac{\partial}{\partial \mathbf{r}} \mathbf{m}_\sigma(\mathbf{r}) = \mathbf{p}_{\mathbf{x}, \sigma}(\mathbf{r}).$$

We sometimes denote the mass function and PDF with an object index such as  $\mathbf{m}_{\mathbf{o}_2}$ , which means  $\mathbf{m}_{\sigma_2}$ . The mass function has to have the following properties.

- a)  $\mathbf{m}_\sigma(0) = 0$ : additional definition.
- b) Increasing function: the definition is the differentiation form and the right side is equal to or greater than zero.
- c) If  $\mathbf{m}_{\sigma_1}(\mathbf{r}) > \mathbf{m}_{\sigma_2}(\mathbf{r})$  for some  $\mathbf{r} > 0$ , then the inequality is true for any positive number.
- d)  $\mathbf{m}_\sigma(\infty) = 1$ : convenient for giving algorithm parameters.

Well-known probability distributions, such as the chi-square function, have these properties. The function  $\mathbf{p}$  is the PDF. The mass function  $\mathbf{m}$  is the cumulative PDF.

### 2.2 The Expansion of Density-Based Clustering

We expand and redefine definitions used in traditional density-based clustering as follows:

(1)  $\varepsilon$ -neighborhood.  $\varepsilon$ -neighborhood of an object  $\mathbf{p}$ , denoted by  $\mathbf{N}_\varepsilon(\mathbf{p})$ , is defined by  $\mathbf{N}_\varepsilon(\mathbf{p}) = \{\mathbf{q} \in \mathbf{O} : \text{dist}(\mathbf{p}, \mathbf{q}) < \varepsilon\}$ . This is the subset of the whole data set that has a distance less than  $\varepsilon$ .

(2)  $\varepsilon$ -neighborhood mass.  $\varepsilon$ -neighborhood mass of an object  $\mathbf{p}$ , denoted by  $\mathbf{M}_\varepsilon(\mathbf{p})$ , is defined by  $\mathbf{M}_\varepsilon(\mathbf{p}) = \sum_{\mathbf{q} \in \mathbf{N}_\varepsilon(\mathbf{p})} \mathbf{m}_\mathbf{q}(\varepsilon - \text{dist}(\mathbf{p}, \mathbf{q}))$ .

Figure 1 shows the concept of  $\varepsilon$ -neighborhood mass in one dimension. This example shows  $\varepsilon$ -neighborhood mass of the center object. The center object is summarized within the radius  $\varepsilon$ , because  $\text{dist}(\mathbf{p}, \mathbf{p}) = 0$ . The mass of center object is represented by the horizontally filled area, including its left-most expansion into the vertically filled area (crosshatched). There are two objects within  $\varepsilon$  except for own object, center object. These objects are summarized within the radius  $\varepsilon\text{-dist}(\mathbf{p}, \mathbf{q})$ . The mass of left object is represented by the vertically filled area, including its overlap with the horizontally filled area (crosshatched). The mass of right object is diagonally filled area. Summing the three masses,  $\mathbf{m}_q(\varepsilon\text{-dist}(\mathbf{p}, \mathbf{q}))$ , gives  $\mathbf{M}_\varepsilon(\mathbf{p})$ . The Crosshatched in this example is double counted.

$\varepsilon$ -neighborhood mass is the expansion of  $\varepsilon$ -neighborhood. It supposes an infinite limit. If the mass is concentrated in the center,  $\varepsilon$ -neighborhood mass equals the number of objects in  $\varepsilon$ -neighborhood.

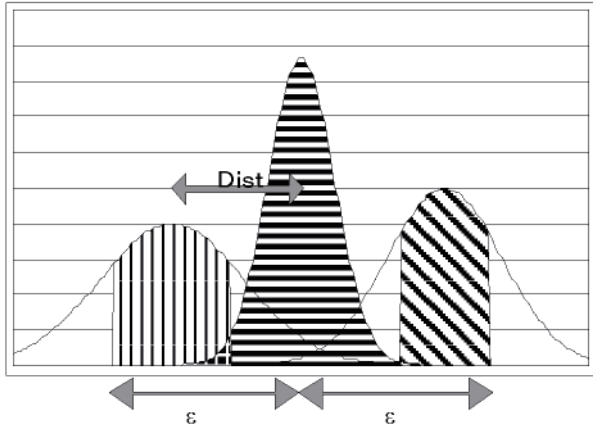


Figure 1:  $\varepsilon$ -neighborhood mass.

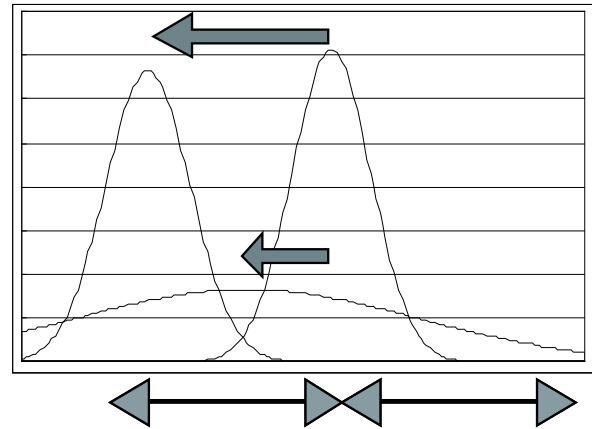


Figure 2: Directly density reachable.

(3) Directly density reachable. An object  $\mathbf{p}$  is directly density reachable from an object  $\mathbf{q}$  wrt.  $\varepsilon$  and  $\mathbf{q}_m$  if:

- a)  $\mathbf{p} \in \mathbf{N}_\varepsilon(\mathbf{q}) \subset \mathbf{O}$ .
- b)  $\mathbf{N}_\varepsilon(\mathbf{q}) > \theta_m$  (core condition 1).
- c)  $\mathbf{m}_q(\varepsilon) > \mathbf{m}_p(\varepsilon)$  (core condition 2).

Core condition 1 is the natural expansion of the core condition in DBSCAN. Core condition 2 shows the direction of the error rate in the experiment. Figure 2 shows the concept of core condition 2. This condition represents flow from a concentrated object to a distributed object, or from a high-density object to a low-density object.

(4) Density reachable. An object  $\mathbf{p}$  is density reachable from an object  $\mathbf{q}$  wrt.  $\varepsilon$  and  $\theta_m$  if there is a chain of objects  $\mathbf{p}_1, \dots, \mathbf{p}_n, \mathbf{p}_1 = \mathbf{p}, \mathbf{p}_n = \mathbf{q}$  such that  $\mathbf{p}_{i+1}$  is directly density reachable from  $\mathbf{p}_i$ . Figure 3 shows the concept of density reachable. This definition is the same as the DBSCAN definition.

(5) Density connected. Density connectivity is a symmetric relation. An object  $\mathbf{p}$  is density connected to an object  $\mathbf{q}$  wrt.  $\varepsilon$  and  $\theta_m$  if there is a chain of objects  $\{\mathbf{o}_1, \mathbf{p}_1, \mathbf{q}_1, \mathbf{o}_2, \mathbf{p}_2, \mathbf{q}_2, \dots, \mathbf{p}_{m-1}, \mathbf{q}_{m-1}, \mathbf{o}_m\}$  such that:

- a) Object  $\mathbf{p}$  is density reachable from object  $\mathbf{o}_1$ .
- b) Object  $\mathbf{q}$  is density reachable from object  $\mathbf{o}_m$ .
- c) Object  $\mathbf{p}_i$  is density reachable from object  $\mathbf{o}_i$ .

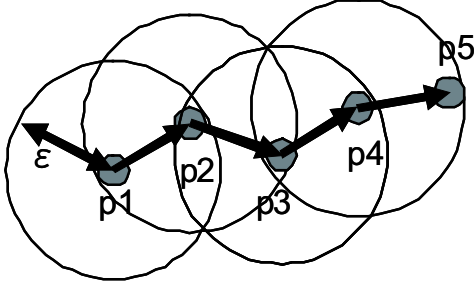


Figure 3: Density reachable.

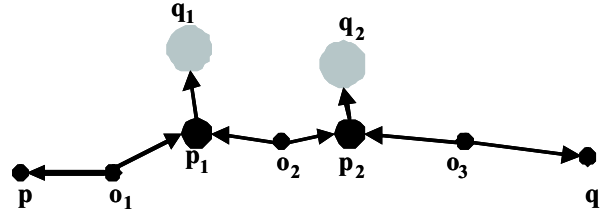


Figure 4: Density connected.

- d) Object  $\mathbf{p}_i$  is density reachable from object  $\mathbf{o}_{i+1}$ .
- e) Object  $\mathbf{q}_i$  is density reachable from object  $\mathbf{p}_1$ .

Because density reachable is defined as flow from a stable object to an unstable object, we cannot define density connected using one object as is done in DBSCAN, so instead we use the chain of objects. Condition 5 shows that each object,  $\mathbf{p}_i$ , is in the core condition. Figure 4 shows an example of density connected. The arrow is the flow of density reachable.

(6) Cluster. A cluster  $\mathbf{C}$  wrt.  $\varepsilon$  and  $\theta_m$  is a non-empty subset of  $\mathbf{O}$  satisfying the following conditions:

- a) If any  $\mathbf{p} \in \mathbf{O}$  satisfies the core conditions, then  $\mathbf{p}$  is a member of some cluster.
- b) For any  $\mathbf{p}, \mathbf{q} \in \mathbf{O}$ : if  $\mathbf{p}$  is a member of  $\mathbf{C}$  and  $\mathbf{q}$  is density connected from  $\mathbf{p}$  wrt.  $\varepsilon$  and  $\theta_m$ , then  $\mathbf{q}$  is a member of  $\mathbf{C}$  (maximality).
- c) For any  $\mathbf{p}, \mathbf{q} \in \mathbf{C}$ :  $\mathbf{p}$  is density connected to  $\mathbf{q}$  wrt.  $\varepsilon$  and  $\theta_m$  (connectivity).

A cluster contains the objects that do not satisfy the core condition. Such an object is called a *border*, and a border object may belong to multiple clusters.

## 2.3 Imitative Hierarchical Tree Structure

### 2.3.1 Lemmas

According to the preceding definitions, the following lemmas are true.

**Lemma 1.** If an object  $\mathbf{p}$  is a core object wrt.  $\varepsilon_1$  and  $\theta_m$ , object  $\mathbf{p}$  is a core object wrt.  $\varepsilon_2 > \varepsilon_1$  and  $\theta_m$ .

**Proof.** An object  $\mathbf{p}$  is a core object wrt.  $\varepsilon_1$  and  $\theta_m$ . This means the following:

- (1)  $\mathbf{M}_{\varepsilon_1}(\mathbf{p}) > \theta_m$  (core condition 1).
- (2)  $\exists \mathbf{q} \in \mathbf{N}_{\varepsilon_1}(\mathbf{p}) \subset \mathbf{O}$  s.t.  $\mathbf{m}_p(\varepsilon_1) > \mathbf{m}_q(\varepsilon_1)$  (core condition 2).

Because  $\mathbf{M}_\varepsilon(\mathbf{p})$  is a strictly increasing function for  $\varepsilon$ ,  $\mathbf{M}_{\varepsilon_2}(\mathbf{p}) \geq \mathbf{M}_{\varepsilon_1}(\mathbf{p}) > \theta_m$  for  $\varepsilon_2 > \varepsilon_1$ . According to the mass function property (c),  $\mathbf{m}_p(\varepsilon_1) > \mathbf{m}_q(\varepsilon_1) \implies \mathbf{m}_p(\varepsilon_2) > \mathbf{m}_q(\varepsilon_2)$ . And, according to the epsilon neighborhood,  $\mathbf{q} \in \mathbf{N}_{\varepsilon_1}(\mathbf{p}) \subset \mathbf{N}_{\varepsilon_2}(\mathbf{p}) \subset \mathbf{O}$ . So,  $\exists \mathbf{q} \in \mathbf{N}_{\varepsilon_2}(\mathbf{p}) \subset \mathbf{O}$  s.t.  $\mathbf{m}_p(\varepsilon_2) > \mathbf{m}_q(\varepsilon_2)$ .

**Lemma 2.** If a subset  $\mathbf{C}$  is a cluster wrt.  $\varepsilon_1$  and  $\theta_m$ , there is a cluster that contains  $\mathbf{C}$  wrt.  $\varepsilon_2 > \varepsilon_1$  and  $\theta_m$ .

**Proof.** Suppose a subset  $\mathbf{C}$  is a cluster wrt.  $\varepsilon_1$  and  $\theta_m$ . According to the connectivity condition, any objects  $\mathbf{p}, \mathbf{q} \in \mathbf{C}$  are density connected. There exists a chain of objects which consists of directly density reachable or density reachable objects. These definitions are valid for  $\varepsilon_2 > \varepsilon_1$ , if satisfied for

$\varepsilon_1$ . So,  $\mathbf{p}$  and  $\mathbf{q}$  are density connected for  $\varepsilon_2$ . According to the maximality condition,  $\mathbf{p}$  and  $\mathbf{q}$  are members of the same cluster.

**Lemma 3.** If an object  $\mathbf{p}$  is a core object wrt.  $\varepsilon$  and  $\theta_{m_1}$ , object  $\mathbf{p}$  is a core object wrt.  $\varepsilon$  and  $\theta_{m_2} < \theta_{m_1}$ .

Proof is the same as Lemma 1.

**Lemma 4.** If a subset  $\mathbf{C}$  is a cluster wrt.  $\varepsilon$  and  $\theta_{m_1}$ , there is a cluster that contains  $\mathbf{C}$  wrt.  $\varepsilon$  and  $\theta_{m_2} < \theta_{m_1}$ .

Proof is the same as Lemma 2.

### 2.3.2 Tree Structure

By proceeding with Lemmas 1, 2, 3 and 4, we can build a hierarchical tree structure if we use the appropriate thresholds and cluster the data. We call this structure a *imitative hierarchical tree structure* to distinguish it from hierarchical clustering.

For example, a sequence of thresholds is  $\{\{\varepsilon_1, \theta_{m_1}\}, \{\varepsilon_2, \theta_{m_2}\}, \dots, \{\varepsilon_n, \theta_{m_n}\}\}$ , and a sequence of clusters  $\{\{\mathbf{C}_{11}, \mathbf{C}_{12}, \dots\}, \{\mathbf{C}_{21}, \mathbf{C}_{22}, \dots\}, \dots, \{\mathbf{C}_{n1}, \mathbf{C}_{n2}, \dots\}\}$  correspond to the thresholds. For any cluster  $\mathbf{C}_{ij}$  and  $k < i$ , there exists a cluster  $\mathbf{C}_{km}$  such that  $\mathbf{C}_{km}$  includes  $\mathbf{C}_{ij}$ . Figure 5 shows a tree structure. Each rectangle indicates cluster.

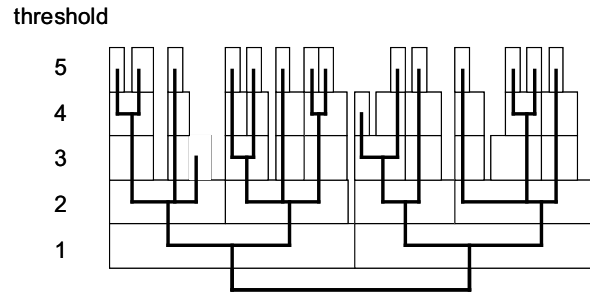


Figure 5: Imitative hierarchical tree structure.

Each rectangle indicates cluster.

## 3 Algorithm

### 3.1 Our Solution

Our proposed algorithm is based on the following ideas:

- (1) Consider the deviation of experimental data to be a mass distribution.
- (2) Expand density-based clustering for the mass distribution.
- (3) Generate the imitative hierarchical clustering tree to adapt the local density.

The deviation in data from identical replicate experiments can be represented as a PDF, and we identify the probability distribution with the mass distribution. By expanding density-based clustering, we created an algorithm to calculate the mass distribution as density. The density of DBSCAN is an integer number that represents the number of objects; in our algorithm, density is a real number. In using our algorithm, unstable genes should not be the core of a cluster, but in sparse regions the criteria of stableness should be loose. Our algorithm clusters for multiple thresholds and generates the imitative hierarchical tree, then chooses the appropriate clusters to adapt the local density.

### 3.2 Probability Distribution Function

We used the *gamma distribution function* as our PDF because cumulative gamma distributions have curves that are shaped like those of chi-square functions. A gamma distribution is a one-dimensional function that gives the distance from the center of an object:

$$p_{\alpha,\beta}(r) = \frac{1}{\beta^\alpha \Gamma(\alpha)} r^{\alpha-1} e^{-r/\beta}.$$

The cumulative gamma distribution is:

$$D_{\alpha,\beta}(r_0) = \frac{1}{\beta^\alpha \Gamma(\alpha)} \int_{r_0}^{\infty} x^{\alpha-1} e^{-r/\beta} dr.$$

We defined the two parameters for a gamma distribution as follows:

$$\alpha = \frac{d}{2}, \quad \beta = \frac{2\sigma^2}{\alpha}.$$

A gamma function has the following properties:

- (1) It is possible to calculate the integral function if alpha is a positive integer; it is called an *incomplete* gamma function.
- (2) It is most dense around the center and least dense far from the center.
- (3) The same deviation must be present in all directions. This condition can be difficult to meet for many domains, but it works for gene expression data because they have the same scale.

After normalizing our data, we defined the mass function as follows:

$$\mathbf{m}_\sigma(\mathbf{r}) = 1 - D_{\alpha,\beta}(\mathbf{r}^2) = 1 - D_{d/2,2\sigma^2/d}(\mathbf{r}^2).$$

### 3.3 Algorithm on Threshold

It is difficult to determine what the threshold should be. An observation error changes the value of gene expression. Because of this, we do not cluster with a single threshold, but make imitative hierarchical clusters by changing threshold values. In this case, we give a threshold at appropriate intervals to perceive to a bigger change, than to perceive a change of the cluster constitution by changing of the delicate value of a threshold.

If there is a pure binary tree structure, the number of relationship within clusters is a power of 2. Figure 6 shows the relationship between tree structure and the relationships within clusters. The threshold marked by a double line indicates the smallest clusters; each cluster contains two objects and four relationships between objects. The threshold marked by a triple line indicates the next-level clusters; each cluster contains four objects and sixteen relations.

We decided to use a rank of the distance between objects. We assigned the ranks using the following formula:

$$\text{Rank} := 10^{i/L} \quad (i = 1, 2, \dots)$$

Where  $\varepsilon_1$  is defined as the 1st nearest distance,  $10^{1/L}$ ,  $\varepsilon_2$  is defined as the 2nd nearest distance,  $10^{2/L}$  and so on.

### 3.4 Representation

After clustering for the threshold, each object is classified as core, as border, or as not belonging to any cluster. According to the Lemmas, when classified with a core object with a certain threshold,

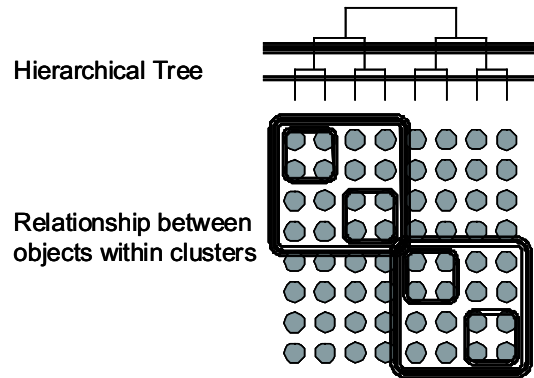


Figure 6: Hierarchical tree and relationship between objects within clusters.

an object is always classified as a core object with a bigger threshold than it. It is thus possible to express core objects with a hierarchical tree structure.

Density-based clustering can find arbitrarily shaped clusters, but in gene expression analysis we want to find the clusters that have similar sizes. In the hierarchical cluster, we can find the appropriate cluster that satisfies the size condition.

### 3.4.1 Appropriate Cluster

For each threshold, we calculate the *diameters* of the clusters. Diameter is defined as the maximum distance between the core objects that belong to the cluster. We define the appropriate cluster as having a diameter less than the threshold and having the maximum diameter for the object.

### 3.4.2 Classification

We call the core objects of the appropriate cluster *rigorous* objects. Core objects that do not belong to an appropriate cluster but which are objects for the loosest threshold are called *shell* objects if they are direct-density-reachable from some rigorous objects, or *adhesive* objects if they are not direct-density-reachable from any rigorous object. The shell objects belong to the cluster that has the nearest rigorous object. There are some objects that are not core objects for the loosest threshold, and we group these into two types. First, the objects that satisfy core condition 1 and do not satisfy core condition 2 are called *unique* objects. These objects satisfy the mass threshold by themselves but they are far from other objects. The remaining objects are classified as *unstable*. All objects are classified into one of these four groups.

## 4 Experiments

### 4.1 Experiment with 2-Dimensional Synthetic Data

#### 4.1.1 Data

The data in the 2-dimensional experiment consisted of 24 objects: 8 objects belonged to the clusters, the others were unstable objects. For each object, 100 points were generated, for a total of 2,400 points. Figure 7 illustrates the data. The four clusters and large amount noise are apparent.

Figure 8 shows the data from three points for each object. The clusters here are much more difficult to see. The difference between Figures 7 and 8 is due to the different number of observations.

Figure 10 shows the average value for each object. The black objects have small errors, whereas the gray objects have large errors. As in Figure 7, four clusters are visible (they are the eight black objects). The 16 gray objects represent background noise. Our algorithm works like Figure 9.

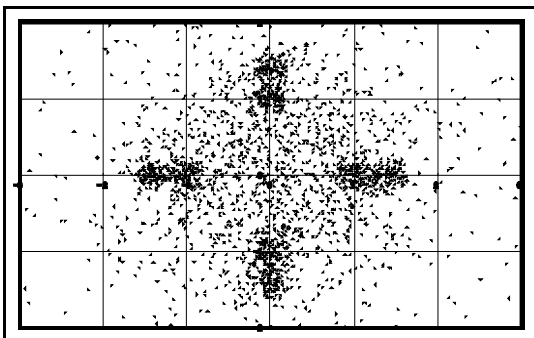


Figure 7: 100 points/object.

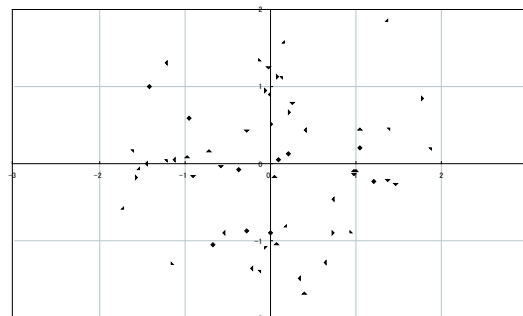


Figure 8: 3 points/object.

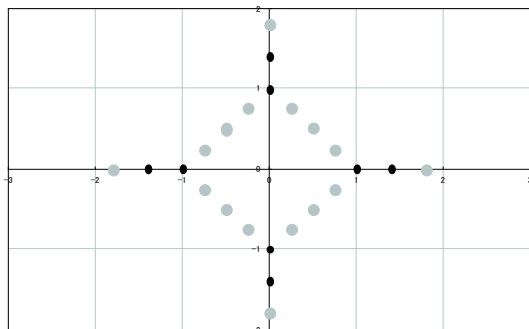


Figure 9: Average points.

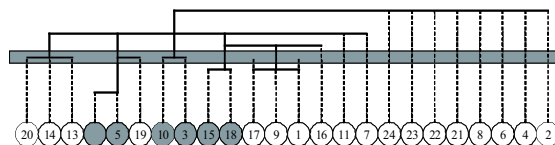


Figure 10: Hierarchical tree results from density-based clustering.

### 4.1.2 Results of Density-Based Clustering

We attempted to cluster with density-based clustering. In this two-dimensional test, we removed the normalization using the  $z$ -score. Figure 10 shows the results from using the density-based clustering algorithm with an imitative hierarchical tree.

Figure 11 is the 2 dimensional plot, which represents the clusters with the gray-labeled threshold in Figure 10. The black dots represent the core objects and the dotted ellipses show each cluster's core objects. The clusters in Figures 7 and 11 seem to be unrelated.

### 4.1.3 Result of MADIC

Figure 12 shows the results of our algorithm. This figure shows the hierarchical tree and appropriate clusters. Our algorithm found five clusters.

Figure 13 shows the classification results. The rigorous objects are black and an ellipse surrounds each cluster. Four of five clusters are the same as those seen in Figure 7. The fifth cluster, in the bottom right quadrant, contains object No. 18. This object has less deviation data than the producing rules. This finding illustrates that errors sometimes affect results. But, it appeared in the loosest threshold. We should use the results with the threshold that appeared. In gene expression analysis, we do analyze the clusters that appeared in the severe threshold.

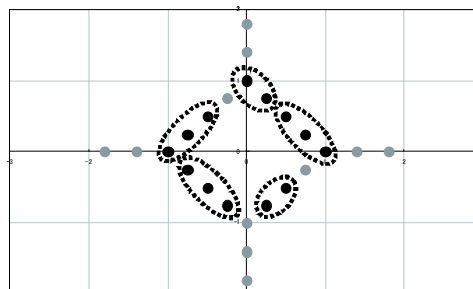


Figure 11: Results of density-based clustering.

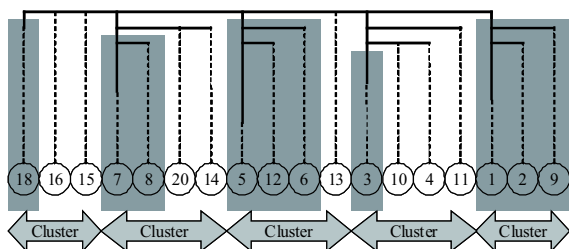


Figure 12: Results of our algorithm.

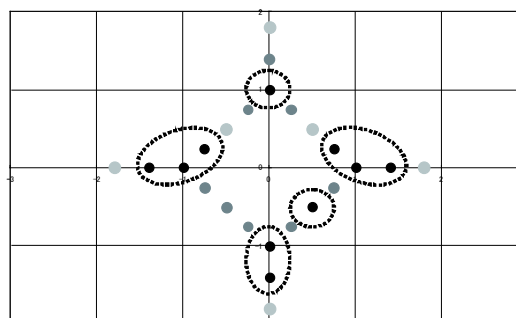


Figure 13: Classification results.

#### 4.1.4 Comparison

Figure 14 shows the analysis flow. The generation rules exist but are always hidden; the objective of data mining is to discover these generation rules. If there are many observation points, we can discover the clusters, as seen in Figure 7. However, we sometimes obtain a limited number of observations.

We consider Figure 13 better than Figure 11. The proposed algorithm solves the new clustering problem. However, we cannot mathematically compare our algorithm and the existing algorithm.

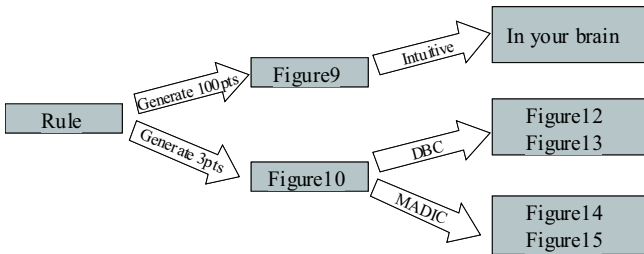


Figure 14: Analysis flow diagram.

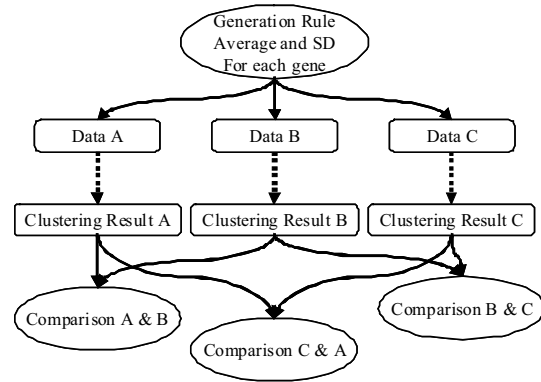


Figure 15: Synthetic data generation.

## 4.2 Experiment with 16-Dimensional Synthetic Data

### 4.2.1 Data

We used three data sets, each produced using the same rule. Each data set has 10,000 objects and 16 dimensions i.e. experimental conditions, and three data for each condition. Figure 15 shows the data generation, clustering, and comparison. We synthesized the average surfaces and then added to them random numbers. These data sets have the same characteristics, and therefore should generate the same clusters. However, the addition of random numbers generates the differences.

### 4.2.2 Evaluation Method

We added a big random number to simulate what we would see with real gene expression data. It is difficult to put them into the same cluster. We evaluated the number calculated by the following equation:

$$\text{Index} = \frac{\sum_{i,j=1} (\#(C_i C_j))^2}{\sqrt{\sum_{i=1} (\#(C_i))^2} \sqrt{\sum_{j=1} (\#(C_j))^2}}.$$

This equation shows the sum of squares of the number of intersections of both clusters, divided by the square roots of the sum of squares of the number of clusters.

This number is bounded from 0 to 1. If two clusters are completely the same, then this number is 1. If whole genes are grouped into one cluster, then this number is 1. So, we have to evaluate the number according to the number of clusters. We do the clustering experiments with various parameters. We gave the various numbers of clusters for  $k$ -means clustering various  $\theta_m$  for MADIC clustering.

### 4.2.3 Results

We performed  $k$ -means clustering and MADIC clustering with various parameters (Figure 16). The indexes of the MADIC method are higher than those for  $k$ -means clustering. This result indicates that MADIC clustering is less affected by the random numbers than is  $k$ -means clustering.

### 4.3 Real Data

#### 4.3.1 Data

NIHS performed experiments to determine how gene expression varied with exposure to four doses of thalidomide (vehicle only, low-dose, mid-dose, and high-dose) and four time points (2, 4, 8, and 24 hours later), i.e., 16 condition points with MOE430A of Affymetrix. Three mice were measured for each condition, creating triplicate data. Thus, 48 chips were used for the experiment.

Thalidomide is a drug for a sleeping aid and a treatment for morning sickness. It was subsequently found to be teratogenic, particularly during the first 25 to 50 days of pregnancy, most visibly causing amelia or phocomelia. The normalization was done by Percellome System for these measurement results. *k*-means clusterings were done for 16-dimensional data, averages for each condition with  $k = 80, 90, 100$ , compared with the MADIC result. It is difficult to predetermine the number of partitions, which is a very important parameter in *k*-means. AIC with EM method gave a partition number of one, which is obviously unacceptable; normally the number of clusters can only be determined from a biological viewpoint. In this study, partition numbers for *k*-means were the number of clusters given by MADIC.

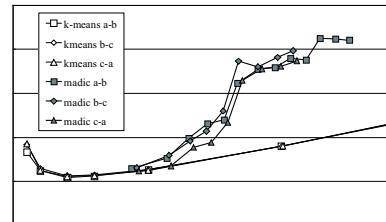


Figure 16: Results of 16-dimensional synthetic data.

#### 4.3.2 Results

MADIC found 138 rigorous probes and 87 clusters. Moreover, 122 unique probes were found. These probes in clusters were summed up by the keywords in the Gene Ontology Biological Process. Ratios were calculated that had the same keywords in each cluster. Figure 17 shows the distributions of number of clusters for highest ratio of keywords in the Biological Process. MADIC could identify the clusters which have ratio more than 18%. This means that MADIC could find clusters which are assumed having biological meaning. Table 2 shows the average of the highest ratios of the same keyword. MADIC was higher than the *k*-means values. It is thought that this reflects division into the cluster that belongs to same key word in the Biological Process. The average about *k*-means was 10.514% and the standard deviation was 0.127%. The MADIC result was 6 SD from the *k*-means' average. This means that MADIC generated more homogenous clusters than *k*-means.

Table 2: Distribution of highest keyword.

<i>k</i> -means $N = 80$	10.369%
<i>k</i> -means $N = 90$	10.575%
<i>k</i> -means $N = 100$	10.599%
MADIC	11.445%

## 5 Conclusion

Traditional clustering algorithms cannot use all the data from repeated measurements. If deviations in repeated measurements are ignored, genes that have big errors can affect the results. Our experiment with 2-dimensional synthetic data shows better results than the *k*-means algorithm. Our experiment with 16-dimensional synthetic data shows the robustness to errors. In the experiment with real data, we assume MADIC creates the appropriate clusters.

The following features make MADIC a useful method for clustering results from gene expression experiments:

- (1) MADIC can handle repeated measurements with error margins; it can identify more stable clusters for stable genes.
- (2) The input parameters do not affect the clusters.
- (3) Random seed numbers are not needed.
- (4) Even if the number of members is one, a peculiar pattern can be extracted as a cluster.

By using our new algorithm, we were able to perform unsupervised clustering of all gene microarray data generated by the Percellome System. This algorithm provides a new option for the analysis of gene expression data.

Our algorithm adopts the gamma function as a density function. Even if an unstable object exists close to a stable object, as a characteristic of the gamma function the unstable object does not affect the stable object. However, because the gamma function does not permit integration by the odd number dimension, it cannot be applied to odd number dimension data. Moreover, it is believed that the device is necessary for very high dimensional data because the smoothness of the incomplete gamma function is lost.

Although our algorithm was designed to analyze microarray data, it should be useful for other types of data that retain error or variation information, and we will subsequently try to apply MADIC to other fields.

## Acknowledgments

This work was supported by the Health Sciences Research Grant from the Ministry of Health, Labor and Welfare, Japan, H13-Seikatsu-012, H13-Seikatsu-013, H14-Toxico-001, and H15-Kagaku-002, and in collaboration with the NTT COMWARE CORPORATION.

## References

- [1] Ester, M., Kriegel, H. P., Sander, J., and Xu, X., A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231, 1996.
- [2] Kanno, J., Aisaki, K., Igarashi, K., *et al.*, “Per cell” normalization method for mRNA measurement by quantitative PCR and microarrays, (in preparation).
- [3] Kumar, M., Patel, N. R., and Woo, J., Clustering seasonality patterns in the presence of errors, *Proceedings of the eighth ACM SIGKDD international conference Knowledge Discovery and data mining*, 557–563, 2002.
- [4] Monti, S., Tamayo, P., Mesirov, J., and Golub, T., Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data, *Machine Learning*, 52:91–118, 2003.
- [5] Papadopoulos, D., Domeniconi, C., Gunopulos, D., and Ma S., Clustering gene expression data in SQL using locally adaptive metrics, *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery (DMKD 2003)*, 235–41, 2003.
- [6] Yeung, K. Y., Medvedovic, M., and Bumbgarner, R. E., Clustering gene-expression data with repeated measurements, *Genome Biol.*, 4(5):R34, 2003.