

IntClust: A Software Package for Clustering Replicated Microarray Data*

Wei Peng and Tao Li
School of Computer Science, Florida International University
Miami, FL 33199
{wpeng002,taoli@cs.fiu.edu}

Abstract

IntClust is a software package for clustering gene-expression data with repeated measurements based on interval data analysis. By utilizing interval data for representing replicated microarray data, **IntClust** is able to take into account the scopes where replicate microarray data are distributed instead of simple data points. The software package offers several transformation models for interval data representations, supports different extended dissimilarity/distance measures for interval data analysis, provides some variations of modified K-means clustering, and presents three popular clustering quality evaluation measures. Our experiments show that **IntClust** improves the clustering performance of gene-expression microarray data over traditional approaches. The software package is available at <http://cadse24.cs.fiu.edu/IntClust>.

1 Introduction

Gene-expression microarray technology is a widely applied high-throughput means for understanding, exploring, and relating genes. It intrigues geneticists in quickly identifying the functional characterization of gene candidates by collecting and mining the microarray data [14]. Due to possible errors in the data, biological variations from different specimens, and limitations of the algorithm or data in the analysis of microarray data, experimental replicates are usually performed in microarray measurements [9, 3]. Applying replicate microarray data improves the precision of inherently noisy data and assessing the reproducibility of observed patterns.

After the microarray data are obtained, the clustering technique is a popular method to identify patterns of gene co-expression. However, the majority of

*Tao Li is partially supported by NSF CAREER grant IIS-0546280 and Wei Peng was supported by a Florida International University Presidential Graduate Fellowship.

current clustering techniques are not able to accommodate appropriately replicate microarray data [9]. Limited work has been reported on addressing this challenging issue to adapt the clustering algorithm to repeated measurements.

Traditional clustering techniques can be easily applied to interval data types by replacing each interval with a representative (e.g, the median of the points in the interval). However, this approach ignores the structure information of the interval. Kerr and Churchill [13] utilize bootstrapping technique on replicate array data to evaluate the clustering stability. They focus on the assessment of stability or reproducibility of clusters. Medvedovic et al. [10] develop a clustering procedure to analyze gene-expression data with repeated measurements based on the Bayesian infinite mixture model(IMM). Hughes et al. [17] apply a software package *Resolver* to estimate error model constructed from repeaweights to the data points with lower confidence. Yeung et al. [9] use the IMM-based approach with built-in error models and variability estimates of repeated measurements in similarity measures to improve clustering accuracy and stability. However, the IMM modeling and estimate methods in the above approaches are little bit intricate. In addition, the weight estimate for data points is also time-consuming.

In this paper, we developed the **IntClust** software to study the problem of clustering replicated microarray data, an important yet largely under-addressed problem. **IntClust** takes into account the scopes where replicate microarray data are distributed instead of simple data points. The employed clustering technique in **IntClust** is a natural extension of traditional K-means clustering [8]. It is simple yet efficient. **IntClust** provides multiple dissimilarity/distance measures for traditional and interval data clustering. By representing the microarray data with repeated measurements as interval data, **IntClust** improves the clustering performance. Another advantage of **IntClust** is its ability to compare clustering qualities of traditional approaches and interval approaches with three popular evaluation metrics. Moreover, it can obtain more descriptive cluster representations by applying interval data clustering rather than two-dimensional tabular data presentations.

The rest of the paper is organized as follows: Section 2 introduces the interval data representations for replicate microarray data; Section 3 presents various distance measures for interval data; Section 4 describes the K-mean type algorithm for interval data clustering; Section 5 shows the experimental results on Yeast galactose data; and Finally Section 6 concludes.

2 Interval Data Representations

The input to **IntClust** is a gene-expression dataset with repeated measurements. To pre-process the raw dataset, **IntClust** offers several transformation models to change it to the corresponding representative datasets. They include the average expression levels over all repeated measurements, the values for each individual measurement, and four interval data presentations derived from repeated measurements. Interval data is described by a group of variables, each

of which contains a range of continuous values instead of the traditional single continuous or discrete value. **IntClust** provides four interval data transformation models including MinMax, MinMax', MeanVar1, MeanVar2 to transform repeated measurements to intervals. Let $A_j = (a_j^1, a_j^2, \dots, a_j^i)$ be the value set of j -th attribute with i repeated values inside, $mean_j$ be the average/mean of set A_j , and δ be the standard deviation. Let min_j , max_j , min'_j , and max'_j be the minimum value, the maximum value, the second minimum value, and the second maximum value of A_j , respectively. These four interval data transformation models are listed as below:

1. *MinMax*: $A_j = [min_j, max_j]$;
2. *MinMax'*: $A_j = [min'_j, max'_j]$;
3. *MeanVar1*: $A_j = [mean_j - \delta, mean_j + \delta]$;
4. *MeanVar2*: $A_j = [mean_j - 2 \times \delta, mean_j + 2 \times \delta]$.

The output to **IntClust** contains the cluster assignment for each gene, cluster prototypes, and clustering quality values.

3 Interval Distance Measures and Their Relationships

In this section, we discuss various distance measures for interval data. It should be pointed out that: although the following discussion is based on datasets having only interval type data, it can be easily generalized to datasets having interval data type as well as traditional single-value data type.

Interval data can be represented by a vector of interval values. Let $A = (A^1, A^2, \dots, A^p)$ and $B = (B^1, B^2, \dots, B^p)$ be two interval objects with p attributes(variables) where $A^i = [a^i, b^i]$ and $B^i = [c^i, d^i]$ indicate the values of the interval for the i^{th} variable.

First, as mentioned in Section 1, a naive way for measuring the distance between the intervals is to compute the distances between their representatives. We refer to the naive approach as *traditional method*. Hence based on different choices of distance, we obtain *traditional L1 distance* and *traditional L2 distance* for interval data. Second, there are other distance measures which explicitly consider the boundary or the structure of the intervals. Typical examples include Hausdorff distance [11], city-block distance [15], and Minkowski(or Euclidean) distance [2]. We refer to these type of measures as extended/modified dissimilarity measures for interval data.

Table 1 lists various distance measures for interval data. The first row in Table 1, $U1$, is one of most common dissimilarity measures for interval data [5]. It computes the distance between symbolic data by comparing their positions, spans, and contents. Specifically, the distance $U1$ between these two interval objects A and B consists of three types of dissimilarity measures(normalized

to $[0, 1]$) $D_\pi(A^i, B^i)$, $D_s(A^i, B^i)$, and $D_c(A^i, B^i)$. D_π indicates the relative positions of two attribute values on the real line, where $|Y^i|$ is the maximum interval length along variable i . D_s computes the span of interval data where $|A^i| = b^i - a^i$, $|B^i| = d^i - c^i$, and $\max(b^i, d^i) - \min(a^i, c^i)$ (also denoted by $|A^i \cup B^i|$) is the span length of A^i and B^i . D_c considers the non-common parts of A^i and B^i , where $inter$ is the length of $|A^i \cap B^i|$. Actually D_c is the normalized length of non-common part of A^i and B^i . When A^i and B^i intersect each other, $inter$ can be represented by $\min(b^i, d^i) - \max(a^i, c^i)$, otherwise it is zero.

Note that $|A^i \cup B^i| - |A^i \cap B^i|$ computes the outer-side nearness between A^i and B^i . and $2|A^i \cap B^i| - |A^i| - |B^i|$ computes the inner-side nearness between A^i and B^i . Hence, $U2$ dissimilarity measure computes the length of non-common parts of interval values with a parameter γ that controls the effect of the inner-side nearness and the outer-side nearness and it can be thought as an approximation to D_c in $U1$. In traditional L_1 and L_2 , $\frac{a^i+b^i}{2}$ and $\frac{c^i+d^i}{2}$ are centroids of A^i and B^i respectively. Modified L_1 and L_2 distances are the natural generalization of traditional L_1 and L_2 distances by taking into account the interval boundaries. The last dissimilarity measure in Table 1 is the Hausdorff distance which was initially defined to compare two sets [11].

Name	Object-wise dissimilarity measure	Component-wise dissimilarity measure
U1	$d_{u1}(A, B) = \sum_{i=1}^p D(A^i, B^i)$	$D_{u1}(A^i, B^i) = D_\pi(A^i, B^i) + D_s(A^i, B^i) + D_c(A^i, B^i)$, where $D_\pi(A^i, B^i) = \frac{ a^i - c^i }{ Y^i }$, $D_s(A^i, B^i) = \frac{ A^i + B^i }{\max(b^i, d^i) - \min(a^i, c^i)}$, and $D_c(A^i, B^i) = \frac{ A^i \cup B^i - A^i \cap B^i }{\max(b^i, d^i) - \min(a^i, c^i)}$.
U2	$d_{u2}(A, B) = \sqrt[q]{\sum_{i=1}^p [\phi_{u2}(A^i, B^i)]^q}$	$\phi_{u2}(A^i, B^i) = A^i \cup B^i - A^i \cap B^i + \gamma(2 A^i \cap B^i - A^i - B^i)$.
Traditional L_1	$d_{TraL1}(A, B) = \sum_{i=1}^p D_{TraL1}(A^i, B^i)$	$D_{TraL1}(A^i, B^i) = \left \frac{a^i+b^i}{2} - \frac{c^i+d^i}{2} \right $.
Modified L_1	$d_{ModL1}(A, B) = \sum_{i=1}^p D_{ModL1}(A^i, B^i)$	$D_{ModL1}(A^i, B^i) = (a^i - c^i + b^i - d^i)$.
Traditional L_2	$d_{TraL2}(A, B) = \sum_{i=1}^p D_{TraL2}(A^i, B^i)$	$D_{TraL2}(A^i, B^i) = \left(\frac{a^i+b^i}{2} - \frac{c^i+d^i}{2} \right)^2$.
Modified L_2	$d_{ModL2}(A, B) = \sum_{i=1}^p D_{ModL2}(A^i, B^i)$	$D_{ModL2}(A^i, B^i) = (a^i - c^i)^2 + (b^i - d^i)^2$.
Hausdorff	$d_{Hau}(A, B) = \sum_{i=1}^p D_{Hau}(A^i, B^i)$	$D_{Hau}(A^i, B^i) = \max(a^i - c^i , b^i - d^i)$.

Table 1: Dissimilarity measures for interval data. $U1$ denotes Gowda and Diday's dissimilarity measure [5] and $U2$ denotes Ichino and Yaguchi's first formulation of a dissimilarity measure [7]. Modified L_1 is also known as city-block distance. $|X|$ denotes the length of the interval X .

In our experiments, we often use modified L_2 dissimilarity measure. The connections among various dissimilarity measures can be found in Appendix A.

4 Clustering Interval Data

In this section, we present an alternative optimization procedure for clustering interval data used in **IntClust**. This procedure is a natural extension of the popular K-means type algorithm.

4.1 Introduction

Interval data can be represented by a vector of interval values. Let $A = \{A_1, A_2, \dots, A_n\}$ be a set of interval objects. Each object A_i can be represented by a vector $A_i = (A_i^1, A_i^2, \dots, A_i^p)$, where there are p interval values that $A_i^j = [a_i^j, b_i^j]$ and $a_i^j \leq b_i^j$. Suppose we want assign the symbolic objects in A into K clusters $C = (C_1, C_2, \dots, C_K)$, where $C_k, 1 \leq k \leq K$ denotes the k -th cluster. We also use $i \in C_k$ to denote that the i -object is in cluster C_k . The clusters have their corresponding representations or prototypes $G = (G_1, G_2, \dots, G_K)$, where G_k can be also represented as vectors of interval values such that $G_k = (g_k^1, g_k^2, \dots, g_k^p)$, and $g_k^j = [x_k^j, y_k^j]$.

As discussed in Section 1, the clustering problem is determined by four basic components: the (physical) data representation, the distance/dissimilarity measures, the objective criterion, and the optimization procedure. The data representation for interval data is a vector of interval values and the distance measures are studied in Section 3. We now present the objective criterion and describe the optimization procedure.

4.2 Objective Criterion

The goal of clustering is to find the representation for each cluster such that a corresponding criterion $\delta(k)$, defined as the sum of distances between the representation and all objects in that cluster, is minimized. Let the representation of cluster C_k be g_k , and interval objects in cluster C_k be A_i ($i \in C_k$). Based on different distance measures, $\delta(k)$ has different representations as follows ¹:

1. $\delta(k) = \sum_{i \in C_k} d_{TraL1}(A_i, g_k)$;
2. $\delta(k) = \sum_{i \in C_k} d_{ModL1}(A_i, g_k)$;
3. $\delta(k) = \sum_{i \in C_k} d_{TraL2}(A_i, g_k)$;
4. $\delta(k) = \sum_{i \in C_k} d_{ModL2}(A_i, g_k)$;
5. $\delta(k) = \sum_{i \in C_k} d_{Hau}(A_i, g_k)$.

¹We don't include $U1$ and $U2$ distance measures here as, in practice, they are usually reduced to other measures [1].

4.3 Clustering Procedure

The optimization procedure is a variant of K-means type algorithm. The clustering is carried out by an iterative procedure that alternates between identification of the cluster representations to minimize δ and allocation of interval data to the closest cluster.

Proposition 1 *The prototype $G_k = (g_k^1, g_k^2, \dots, g_k^p)$ of cluster C_k that minimizes δ , defined in Section 4.2, is given as follows:*

1. For traditional L_1 distance, $g_k^j = x_k^j$, where x_k^j is the median of the set $\{\frac{a_i^j + b_i^j}{2} | i \in C_k\}$;
2. For modified L_1 distance, $g_k^j = [x_k^j, y_k^j]$, where x_k^j is the median of $\{a_i^j | i \in C_k\}$ and y_k^j is the median of $\{b_i^j | i \in C_k\}$;
3. For traditional L_2 distance, $g_k^j = x_k^j$, where x_k^j is the mean of the set $\{\frac{a_i^j + b_i^j}{2} | i \in C_k\}$;
4. For modified L_2 distance, $g_k^j = [x_k^j, y_k^j]$, where x_k^j is the mean of $\{a_i^j | i \in C_k\}$ and y_k^j is the mean of $\{b_i^j | i \in C_k\}$;
5. For Hausdorff distance, the representation interval data is $g_k^j = [x_k^j, y_k^j]$, where $x_k^j = \text{median}\{\frac{a_i^j + b_i^j}{2} | i \in C_k\} - \text{median}\{\frac{a_i^j - b_i^j}{2} | i \in C_k\}$, and $y_k^j = \text{median}\{\frac{a_i^j + b_i^j}{2} | i \in C_k\} + \text{median}\{\frac{a_i^j - b_i^j}{2} | i \in C_k\}$.

Remark 1 *Note that the representation prototypes for traditional L_1 distance and L_2 distance are shown in [6]. The representation prototypes for modified L_1 distance and L_2 distance are natural generalizations of the traditional ones. The derivation of the representation prototype for Hausdorff distance follows from Equation 4.*

Proposition 1 provides the basis of the **Identification Step** for the clustering procedure, i.e., to identify the representations of clusters to minimize δ .

Proposition 2 *An interval object A_j is assigned to the cluster m with the prototype which is nearest to that object:*

1. $m = \text{argmin}_{m=1, \dots, K} d_{\text{Tra}L_1}(A_j, g_m)$;
2. $m = \text{argmin}_{m=1, \dots, K} d_{\text{Mod}L_1}(A_j, g_m)$;
3. $m = \text{argmin}_{m=1, \dots, K} d_{\text{Tra}L_2}(A_j, g_m)$;
4. $m = \text{argmin}_{m=1, \dots, K} d_{\text{Mod}L_2}(A_j, g_m)$;
5. $m = \text{argmin}_{m=1, \dots, K} d_{\text{Hau}}(A_j, g_m)$.

Proposition 2 establishes the **Allocation Step** of the clustering procedure, i.e., assigning each interval object A_j to the cluster m with the prototype which is nearest to that object.

4.4 Clustering Procedure

Based on the above Proposition 1 and Proposition 2, the clustering procedure can be described as follows: An initial cluster configuration is first generated. This can be done by randomly assigning interval objects into K clusters or by choosing K interval objects as the initial representations of the clusters. Then the clustering procedure iterates between the identification step and allocation step until it converges or some stopping criterion is met.

5 Experiments

In this section, we conduct experiments of applying the interval data clustering to cluster replicated microarray data. The software **IntClust** is written in J# in Microsoft .NET framework. It has been tested on Windows XP operating systems. It can be applied to not only replicate gene-expression microarray data clustering, but also any other k-means clustering on classical data or interval data. The tool can be downloaded from <http://cadse24.cs.fiu.edu/IntClust>.

5.1 Datasets Description

Yeast galactose data of Ideker et al. [16] which is expression gene data with repeated measurements was used in our experiments. 205 genes galactose data are described by 20 experiments or expression levels (nine deletions and one wild-type without galactose and raffinose, nine single-gene deletions and one wild-type experiment with galactose and raffinose). Each experiment contains four replicate hybridization expression values based on four different measurements. The expression patterns of these genes reflect four functional categories which is used to calculate our clustering qualities. The missing data values are pre-processed by KNN imputation [9].

5.2 Clustering quality evaluation

IntClust provides three clustering quality evaluation metrics: Entropy, Purity, and Adjusted Rand Index. These measures have been widely used in microarray data analysis [4, 9].

Purity measures the extent to which each cluster contained data points from primarily one class [19]. The purity of a clustering solution is obtained as a weighted sum of individual cluster purity values and is given by

$$Purity = \sum_{i=1}^K \frac{n_i}{n} P(S_i), P(S_i) = \frac{1}{n_i} \max_j (n_i^j),$$

where S_i is a particular cluster of size n_i , n_i^j is the number of documents of the i -th input class that were assigned to the j -th cluster, K is the number of

clusters and n is the total number of points ². In general, the larger the values of purity, the better the clustering solution is.

Entropy measures how classes distributed on various clusters [19]. The entropy of the entire clustering solution is computed as:

$$Entropy = -\frac{1}{n \log_2 m} \sum_{i=1}^K \sum_{j=1}^m n_i^i \log_2 \frac{n_i^j}{n_i}, \quad (1)$$

where m is the number of original labels, K is the number of clusters. Generally, the smaller the entropy value, the better the clustering quality is.

The Rand Index is defined as the number of pairs of objects which are both located in the same cluster and the same class, or both in different clusters and different classes, divided by the total number of objects [18]. Adjusted Rand Index which adjusts Rand Index is set between $[0, 1]$ [12]. The higher the Adjusted Rand Index, the more resemblance between the clustering results and the labels.

5.3 Experimental results

By summarizing and computing on the raw yeast galactose dataset, we got nine alternative datasets of raw data: i) Four datasets, denoted as Measure 1, Measure 2, Measure 3 and Measure 4, are derived by extracting a particular value from the four experiment, respectively; ii) One is composed by selecting means of repeated values as representative experiment values, denoted by *Mean*; iii) Four datasets obtained by four interval data transformation models, denoted by *MeanVar1*, *MeanVar2*, *MinMax*, and *MinMax'* respectively.

To measure the clustering performance, we use entropy, purity and adjusted Rand Index as they are widely used in microarray data analysis. The clustering results are shown in Figure 1. We use modified interval k-means to cluster the former four representatives by modified L_2 dissimilarity, and traditional k-means to cluster the rest ones by traditional L_2 . From performance comparisons, we observe that generally all interval approaches yield better results than traditional approaches. To get a better understanding of the advantages of interval approaches, we take a closer look at some examples from experimental results. For instance, Gene *RPS8A* and Gene *RPL23A* are supposed to be categorized into the same class while Gene *RPS6B* stays in different class according to external knowledge. Using the modified interval approach on *MinMax*, Gene *RPS8A* and Gene *RPL23A* are perfectly assigned to the same cluster. However, using the traditional approach based on *Mean*, Gene *RPL23A* and Gene *RPS6B* are grouped together and Gene *RPS3A* is separated from Gene *RPL23A*. In other word, $D_{Inter}(RPS8A, RPL23A) = 18.7492 < D_{Inter}(RPL23A, RPS6B) = 21.5498$, whereas $D_{Tra}(RPL23A, RPS6B) = 9.47405 < D_{Tra}(RPS8A, RPL23A) = 20.3835$.

In general, modified interval dissimilarity measures explicitly consider the structure of interval data and yield better clustering results. Moreover, another

² $P(S_i)$ is also called the individual cluster purity.

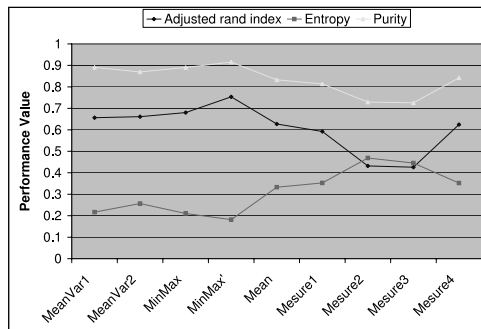


Figure 1: Performance Comparisons of the Clustering Results on Yeast galactose Dataset. Results are obtained by averaging 15 trials.

advantage of modified interval approaches over traditional ones is that the output cluster prototypes are represented by intervals which are more descriptive than simple quantitative values.

6 Conclusion

In this paper, we present a software package **IntClust** for clustering replicate microarray data. We introduce different interval data representations and discuss various interval data distance measures. Our experimental results show that extended interval data clustering achieves better performance than traditional ones, and extended interval approaches excel the traditional ones by taking fully advantages of repeated values.

Appendix A: Relations Among Various Measures

In this section, we investigate the non-trivial relationships among various distance measures. The relationships among various measures are summarized in Figure 2.

First, the $U2$ dissimilarity measure can be viewed as an approximation to D_c in $U1$ as it computes the length of non-common parts of interval values with a parameter γ that controls the effect of the inner-side nearness and the outer-side nearness.

Second, different choices of γ yield different distance measures. When $\gamma = 0$, ϕ_{u2} becomes $|A^i \cup B^i| - |A^i \cap B^i|$. It can be easily shown that in this case, with $q = 1$, the object-wise dissimilarity measure $U2$ is equivalent to the city-block distance (i.e., modified L_1 distance). When $\gamma = 0.5$, component-wise dissimilarity $\phi_{u2}(A^i, B^i)$ can be denoted as

$$n = \phi_{u2}(A^i, B^i) = \frac{b^i - a^i}{2} - \frac{d^i - c^i}{2} = \frac{|A^i| - |B^i|}{2}. \quad (2)$$

where $A^i \subseteq B^i$ or $B^i \subseteq A^i$ (one component contains the other component). Similarly, when A^i and B^i intersect, $\phi_{u2}(A^i, B^i)$ becomes

$$m = \phi_{u2}(A^i, B^i) = \frac{a^i + b^i}{2} - \frac{c^i + d^i}{2}, \quad (3)$$

In this case, the object dissimilarity measure d_{u2} is thus equal to traditional Minkowski dissimilarity measures for interval data. In particular, when $q = 1$, d_{u2} is equivalent to traditional L_1 distance d_{TraL1} ; when $q = 2$, d_{u2} is equivalent to traditional L_2 distance d_{TraL2} . Hausdorff distance synthesizes two possible situations of U2 in the case of $\gamma = 0.5$ at the same time. It can be represented as

$$d_{Hau}(A, B) = \sum_{i=1}^p \max(|m - n|, |m + n|) = |m| + |n|. \quad (4)$$

Observe that when A^i and B^i intersect, the numerator of D_c can be replaced by $|a^i - c^i| + |b^i - d^i|$. So in this situation D_c is a normalized version of the modified L_1 distance. When $A^i \subseteq B^i$ or $B^i \subseteq A^i$, D_s is also the normalized version of modified L_1 distance. D_π is the normalized Hausdorff distance when $|a^i - c^i| > |b^i - d^i|$. In addition, from formulas of D_{TraL1} , D_{ModL1} , D_{TraL2} , and D_{ModL2} , it is easy to deduce that $D_{ModL1} > 2 \times D_{TraL1}$ and $D_{ModL2} > 2 \times D_{TraL2}$.

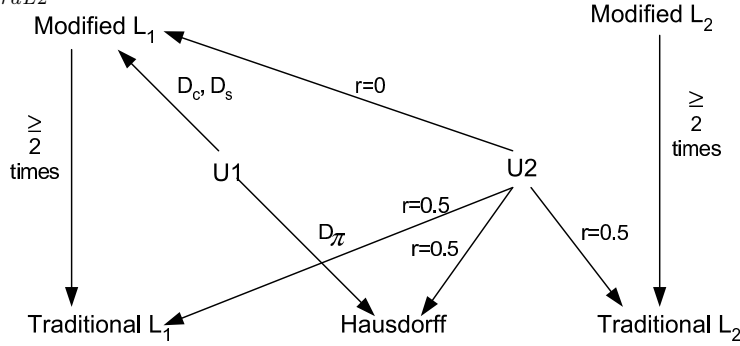


Figure 2: The Relations Among Various Dissimilarity Measures for Interval Data

References

- [1] H.H. Bock and E. Diday. *Analysis of Symbolic Data*. Analysis of Symbolic Data. Exploratory methods for extracting Statistical Information from Complex Data, Series: Studies in Classification, Data Analysis, and Knowledge Organisation, Vol.15. Springer-Verlag, Berlin, 2000.
- [2] F.De Carvalho, P.Brito, and H.H.Bock. Dynamic clustering for interval data based on l_2 distance. Technical report, Cidade Universitaria, 2004.

- [3] Chris Cheadle, Marquis P.Vawter, William J.Freed, and Kevin G.Becker. Analysis of microarray data using z score transformation. *Journal of Molecular Diagnostics*, 5(2):73–81, May 2003.
- [4] Berrar D.P., Sturgeon B., Bradbury I., Downes C.S., and Dubitzky W. Microarray data integration and machine learning techniques for lung cancer survival prediction. In *Critical Assessment of Microarray Data Analysis (CAMDA 2003)*, pages 43–54, Durham, North Carolina, USA, 2003.
- [5] K.C. Gowda and E. Diday. Symbolic clustering using a new dissimilarity measure. In *Pattern Recognition*, 24:567–578, 1991.
- [6] John A. Hartigan. *Clustering Algorithms*. Wiley, 1975.
- [7] M. Ichino and H. Yaguchi. Generalized minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 24:698–707, 1994.
- [8] J.B.MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, 1967. University of California Press.
- [9] Yeung KY, Medvedovic M, and Bumgarner RE. Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34, 2003.
- [10] Medvedovic M and Sivaganesan S. Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, 18:1194–1206, 2002.
- [11] M.Chavent and Y.Lechevallier. Dynamical clustering algorithm of interval data: Optimization of an adequacy criterion based on hausdorff distance. In: *Sokolowsky and H.H. Bock Eds., Classification, Clustering and Data Analysis. Springer, Heidelberg*, 1:53–59, 2002.
- [12] Glenn Milligan and Martha Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21:441–458, 1986.
- [13] Kerr MK and Churchill GA. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. In *Proc Natl Acad Sci USA*, volume 98, pages 8961–8965, 2001.
- [14] Kerr MK and Churchill GA. Statistical design and the analysis of gene expression microarray data. *Genetic Research*, 77:123–128, 2001.
- [15] Renata M.C.R.de Souza and Francisco de A.T.de Carvalho. Clustering of interval data based on city-block distances. *Pattern Recognition Letters 25*, pages 353–365, 2004.

- [16] Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner RE, Goodlett DR, Aebersold R, and Hood L. integrated genomic and proteomic analyses of a systemically perturbed metabolic network. *Science* 2001, 292:929–934.
- [17] Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, Bennett HA, Coffey E, and He YD Dai H. Functional discovery via a compendium of expression profiles. *cell*, 102:109–126, 2000.
- [18] Rand WM. Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc*, 66:846–850, 1971.
- [19] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.