More Work on *K*-Means Clustering Algorithm: The Dimensionality Problem

Barileé Barisi Baridam
Department of Computer Science
University of Pretoria
South Africa

ABSTRACT
The K-means clustering algorithm is an old algorithm that has been intensely researched owing to its simplicity of implementation. However, there have also been criticisms on its performance, in particular, for demanding the value of *K a priori*. It is evident from previous researches that providing the number of clusters *a priori* does not in any way assist in the production of good quality clusters. The objective of this paper is to investigate the usefulness of the K-means clustering in the clustering of high and multi-dimensional data by applying it to biological sequence data which is known for high and multi-dimension. The squared-Euclidean distance and the cosine measure are used as the similarity measures. The silhouette validity index is used first to show *K*-means algorithm's inefficiency in the clustering of high and multi-dimensional data irrespective of the distance or similarity measure employed. A further study was to introduce a preprocessor scheme to the *K*-means algorithm to automatically initialize a suitable value of *K* prior to the execution of the *K*-mean algorithm. The dimensionality problem investigated suggests that the use of the preprocessor improves the quality of clusters significantly for the biological data sets considered. Furthermore, it is then shown that the *K*-means algorithm with preprocessor produces good quality, compact and well-separated clusters of the biological data obtained from a high-dimension-to-low- dimension mapping scheme introduced in the paper.

General Terms
K means, Clustering, Algorithm.

Keywords
Clustering, Dimensionality, Categorical data, Silhouette validity index

1. INTRODUCTION
Clustering is a statistical concept that has to do with the problem of identifying interesting distribution patterns and similarities between objects in a data set [1], [2], [3]. It is an optimization problem that seeks to classify objects based on their proximity to one another. In this sense, objects that are most similar are grouped together forming groups of similar objects referred to as clusters. The task any clustering algorithm has is to produce clusters that are compact and well-separated from one another. It follows that a clustering task involves minimizing the intra-cluster distance or the within-cluster dispersion and maximizing the inter-cluster distance or the between-cluster dispersion.

There are two broad categories of clustering algorithms, namely hierarchical and partition-based clustering. *K*-means [4] is a well-known partition-based clustering technique. It has been widely used since first introduced in 1967. It, as a general rule, demands the value of *K*, the number of clusters expected, to be provided before the actual clustering. This is common to partition- based clustering algorithms [5]. Besides the provision of the value of *K a priori*, it is actually expected that the clusters centers are also to be identified, and then the algorithm performs the partitioning tasks iteratively until a solution is achieved. On the contrary, hierarchical clustering algorithms group objects into clusters without any knowledge of how many clusters there should be in the clustering task. This paper deals with the partition- based *K*-means clustering.

The task of determining *K* a priori actually results into the problem of determining which cluster each object belongs. Clearly, the initial *K* has impact on the performance of the algorithm. A wrong choice of *K* results in the algorithm converging to a local minimum instead of an expected global minimum solution. Running the algorithm several times with different initializations tend to overcome this problem. However, this process results in high computational time. A number of algorithms have been suggested to determine a suitable value of initial K , see for example ISODATA [6], SYNERACT [7], DYNOC [8] and MLBG [9]. However, all these algorithms contain sensitive parameters, and this means that trying to solve one problem creates another of similar nature.

Under the above circumstances, a preprocessor is incorporated prior to the execution of *K*-means. The silhouette validity index [10] plays an important role in determining the initial *K*. It is shown in the paper that this optimizes *K*-means' performance in clustering the high dimensional data sets.

The remaining part of this paper is divided into sections as follows: Section 2 focuses on the preliminaries and related literature; Section 3 describes the new approach suggested for high dimensional data. Section 4 briefly presents the silhouette validity index. Section 5 shows the experimental results and Section 6 presents the concluding remarks.

2. THE CLUSTERING TASK: DATA SETS, SIMILARITY MEASURES AND ALGORITHMS

2.1 The data set
Associated with a given data set

$$S = \{x^1, x^2, \ldots, x^N\} \quad (1)$$

that needs clustering, are the attributes $(x_1^i, x_2^i, \ldots, x_D^i)$ of $x^i \in S$. For a numerical data set *S*, each j^{th} attribute x_j^i in x^i is real and hence $x^i \in \mathbb{R}^D$. However, for a mixed data set features of x^i are generally two: numerical and categorical. Therefore, the attributes of x^i can be written as $(x_1^i, x_2^i, \ldots, x_p^i, y_1^i, y_2^i, \ldots, y_q^i)$, $p+q = D$, where $y_1^i, y_2^i, \ldots, y_q^i$ are categorical values.

Biological data being considered in this paper are that of nucleic acids - Deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). Thus the categorical biological data set *S*

x^i = (*AAAAUUUUGGGCCAAAGGCCCUUUAAGCCCGG*) for RNA (2)

and

x^i = (*AAAAT TTTGGGCCAAAGGCCCTTTAAGCCCGG*) for DNA (3)

DNA is a double helix structure with two strands of reoccurring nucleotides held together by base pairing. The top of the helix consists of a Guanine-Cytosine (GC) pair, referred to as purines, while the bottom consists of an Adeline-Thymine (AT) pair, referred to as pyrimidines. The GC base pair forms three hydrogen bonds, whereas the AT base pair forms two hydrogen bonds. DNA is a polymer with the nucleotides forming the monomer units. In its double stranded form, DNA is the genetic material of most organisms. The two strands form a double helix with the strands running in opposite directions as determined by the sugar-phosphate backbone of the molecule. DNA is represented in chains of symbols - AGCT (Adenine, Guanine, Cytosine, Thymine). For the purpose of this research they are represented as in (3). RNA is a bi-molecule made up of a chain of nucleotides as DNA, except that RNA introduces uracil (U) in place of Thymine (T). RNA and DNA are functionally and structurally different. A RNA strand folds onto itself. The folds form hydrogen bonds between G and C, A and U, and G and U, and their respective mirror images. The hydrogen bonds bind the base pairs to form DNA. There are enough literature for those interested in studying more of the structure of nucleic acids.

The clustering process involved in this paper uses the above type of data. However, for the clustering purpose the biological sequence data have been converted into numerical data. Nucleic acids are represented in 3 dimension. During the sequencing process (that is, the process of converting nucleic acids into readable sequences) the 3 dimensional structure is rendered in a chain of nucleotides. The sequencing process renders the sequences in high- and multi-dimensions.

2.2 The similarity measure

An important component of a clustering algorithm is the distance measure between data points, say x^i and x^j . For continuous numerical data sets the squared Euclidean distance

$$d_{ij} = d\left(x^i, x^j\right) = \sum_{k=1}^{D}\left(x_k^i - x_k^j\right)^2 \quad (4)$$

is often used. The other well-known similarity measure is the cosine similarity measure:

$$d_{ij} = d\left(x^i, x^j\right) = \frac{x^i . x^j}{\left\|x^i\right\|\left\|x^j\right\|} \quad (5)$$

where x^i is the length of the vector x^i, and $x^i.x^j$ is the dot product between vectors x^i and x^j. Both measures have been implemented for comparison purposes.

2.3 The *K*-means algorithm

The minimization problem involved in the *K*-means algorithm for numerical data set can be formally written as follows [11]:

$\min \sum_{m=1}^{K}\sum_{i=1}^{N} r_{im}\, d\left(x^i, C^m\right), r_{im} \in \{0,1\}$, subject to

$\sum_{m=1}^{K} r_{im} = 1, \forall i$, and $\sum_{i=1}^{N} r_{im} > 0, \forall m$,

where C^m is the centroid of the m^{th} cluster and $d(x^i, C^m)$ is defined by equation (4) or (5). If x^i is assigned to cluster m then $r_{im} = 1$. The clustering process partitions a data set into K clusters S^i $(i = 1, 2, \cdots, m)$ such that

(i) $S^i = \emptyset$, $i = 1, \ldots, K$;

(ii) $\bigcup_{i=1}^{K} S^i = S$;

(iii) $S^i \cap S^j = \emptyset, \forall\, i, j = 1, \ldots, K$ and $i \neq j$.

The basic steps of the K -means algorithm for numerical data set are as follows.

Algorithm 1: *K*-means dustering

Step 1. Step 1. Assign K initial centroids $C^1, C^2, \cdots, C^K$, one for each cluster S^m.

Step 2. For each data element $x^i \in S$ find the nearest C m according to some similarity measure, e.g. the measures (4) or (5), and assign x^i to the cluster S^m.

Step 3. For each cluster S^m calculate a new centroid C^m.

Step 4. Stop Algorithm 1 else goto Step 2 with the new centroids $C^1, C^2, \cdots, C^K$.

2.4 The modified K -means algorithms

A number of modified K -means algorithms have been proposed in the literature. The purpose of these modified versions is to handle the problem related to initial K value.

Turi [12] proposed a K -means algorithm by dynamically changing the value of K as the iterations progress. Central to this algorithm are the merging and splitting of clusters. However, the algorithm requires the user to specify the values of several parameters (e.g. the merging and splitting thresholds). These parameters have a profound effect on the performance of making the result subjective.

Huang [7] proposed a *K*-means algorithms, referred to as SYNERACT. SYNERACT combines *K*-means algorithm with hierarchical divisive approaches to overcome *K*-means' setbacks. SYNERACT employs a hyper-plane to split a cluster into two smaller clusters and then compute their centroids, performs an iterative clustering to assign objects into clusters, and constructs a binary tree to store clusters generated from the splitting process. This method does not demand the initial provision of *K* and the initial location of centroids before the clustering task. However, the user is expected to specify the values of two parameters needed for the splitting process.

The dynamic optimal cluster-seek (DYNOC) algorithm was introduced by Tou [8]. DYNOC is a dynamic clustering algorithm. It achieves a maximization of the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance through an iterative procedure with the capability of splitting and merging clusters. There are however user-specified parameters that suggest whether splitting or merging are necessary.

Rosenberger and Chehdi [9] attempted an improvement on *K*-means by introducing an iterative procedure known as the modified Linde-Buzo-Gray (MLBG) algorithm. MLBG automatically finds the number of clusters in a data set by using intermediate results. A cluster maximizing an intra-cluster distance measure is chosen for splitting iteratively. In the process, two cluster centroids are generated from the initial cluster. The first cluster centroid, C^1, is initialized to the centroid of the original (initial) cluster. The second cluster centroid, C^2, is chosen to be the object in the original cluster

which is the most distant from C^1. At this point, K -means is on the new $K + 1$ centroids. The acceptance of the new set of centroids depends on if an evaluation criterion based on a dispersion measure is satisfied. This process is repeated until there are no valid partitions in the data set. The main problem with this method is that it requires the specification of the values of four parameters which have a fundamental effect on the resultant number of clusters [13].

The *K*-means algorithm is found to be the best applied to numeric data [14], and the modifications, discussed above, dealing with numerical data set are very encouraging. However, the application of the *K*-means algorithm to mixed data set is extremely limited. An at- tempt is made by Gupta *et. al.* [15] to apply the *K*-means algorithm by adopting two different similarity measures. An integrated cost function is suggested which has two components. A cost owing to numeric attributes is minimized by usual way i.e. assigning elements to clusters, while the other cost, owing to categorical attributes, is minimized by selecting the categorical elements of centroid. However, the method of Gupta *et. al.* [15] has neither been justified by mathematical means nor has it been validated by sufficient numerical testing. In addition, the method is not parameter-free. Finally, the clustering of categorical biological data set (e.g. nucleic or amino acids) has not been address in the literature. These data sets contain high and multi-dimensional data sequence x^i, and to the best of my knowledge, there is no *K*-means algorithm developed for clustering of these data sets. Hence, the decision to study this clustering problem.

3. APPLICATION OF *K*-MEANS TO BIOLOGICAL SEQUENCE DATA

It is well known that K-means clustering algorithm is incapable of handling multi-dimensional data - a situation where the input data are of several dimensions. However, it is interesting to note that K-means have been employed in the clustering of biological data with different representation (not sequence data) [16], [17], [18]. In this paper, the ability of K-means in the clustering of high and multi-dimensional data sets is investigated. In addition, the biological sequence data sets considered are naturally not numeric. The original objective of the *K*-means algorithm [4] and the subsequent findings [14] suggest that the numerical presentation of the categorical biological data set is needed for successful applications of *K*-means. For this, conversion from symbols to numeric is done by representing each sequence in the data set in a *D*-dimensional space through the application of a comma delimited conversion format. In particular, the nucleic acid symbols are represented numerically as follows: $A = 1, C = 2$, $G = 3$ and U or $T = 4$. For the clustering of a biological sequence data set, the following two separate approaches are adopted.

- Firstly, the sequences in the data set were truncated to a uniform dimension before the clustering, leaving the sequences in their high-dimensional state.
- Secondly, the dimension of each of the sequences was reduced to a uniform low dimension (D_r) before clustering.

The dimension reduction is defined by introducing the following concepts and definitions. Let N represent the number of nucleotides in a sequence; l, the sequence length; n_i, the i^{th} individual nucleotides (symbols) in a sequence already represented in numeric format. It is important to note that it is conventional to state that $N = l$ in cases where the delimiters are not counted to constitute part of the length e.g. as presented by equations (2) and (3). The coordinates of a sequence $x^i \in S$, can be derived as follows:

$$Q_i = \left(\sum_{i=1}^{\bar{d}_1} n_i, \sum_{i=\bar{d}_1+1}^{\bar{d}_2} n_i, \ldots \sum_{i=\bar{d}_j+1}^{D} n_i \right) \quad (6)$$

where $\bar{d}_p = \sum_{i=1}^{p} d_i$ with $\bar{d}_1 = d_1$, $p = 1, 2, \ldots, j$. The use of $d_p = d_q$, $p \neq q$ for all $p, q = 1, 2, \ldots, j$, is made whenever possible. When this is not possible an integer in $\{d_1, d_2, \ldots, d_j\}$ is selected at random and its value is adjusted so that

$$\left(\sum_{k=1}^{j} d_k \right) + \left(D - \bar{d}_j \right) = D.$$

A two dimensional representation of Equation (6) is given by:

$$Q_i = \left(\sum_{i=1}^{d_1} n_i, \sum_{i=\bar{d}_1+1}^{d_2} n_i \right). \quad (7)$$

If the coordinates of Q_i become large then they can be represented in ratios of least common multiples, but this was not required for the data sets considered for numerical testing. The above procedures are implemented for the clustering of the nucleic acid sequences.

4. THE SILHOUETTE INDEX

The silhouette validity index for each data element is simply a measure of how similar that data element is to elements in its own cluster compared to elements in other clusters [19], [20]. It ranges from -1 to +1. The silhouette validation index is particularly useful when seeking to know the number of clusters that will produce compact and clearly separated clusters [21], [22], [10]. The silhouette index [10], [23] of the element *xi* of a cluster S^j is defined as

$$q_i = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad -1 \leq q_i \leq 1 \quad (8)$$

where *a(i)* is the average similarity between xi and the rest of the objects in cluster S^j and *b(i)* is the minimum average similarity between object xi and the rest of the objects in all the clusters, defined as $\min_{S^m \neq S^j} d\left(x^i, S^m\right)$ ($m = 1, 2, \ldots, K$; $m \neq j$).

Table 1. Effects of *K* in the *K*-means algorithm applied to the high dimensional data

				Iterations		Total distance		Silhouette mean	
i	*N*	*D*	*K*	I_E	I_C	Td_E	Td_C	Sh_E	Sh_C
1	117	128	5	7	11	16264.3	8.71803	0.0335	0.0338
	117	128	10	11	9	14791.7	7.99373	0.0533	0.0396
2	117	198	5	20	10	25844.5	9.03496	0.0180	0.0288
	117	198	10	11	7	23776.3	8.32794	0.0309	0.0318
3	100	50	4	12	10	4654.08	6.19611	0.0622	0.0561
	100	50	6	7	11	4968.8	7.8109	0.0663	0.0656
4	50	50	5	8	5	2366.79	3.24409	0.0916	0.0742
	50	50	4	7	11	2512.22	3.3818	0.0707	0.0720
5	50	20	5	6	5	865.011	2.93337	0.1260	0.1270
	50	20	4	4	11	937.727	3.13017	0.0995	0.1281
6	20	50	4	4	3	829.083	1.16313	0.1193	0.1134
	20	50	3	3	4	919.786	1.30774	0.0962	0.1077

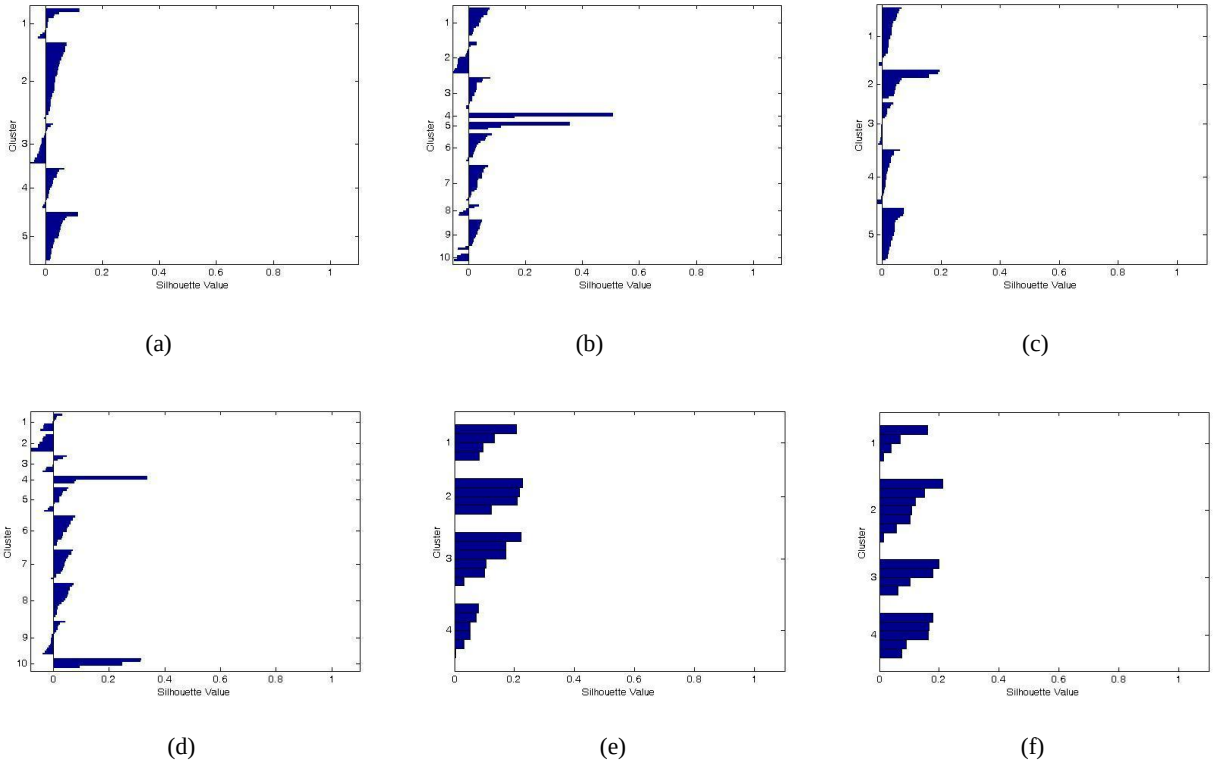


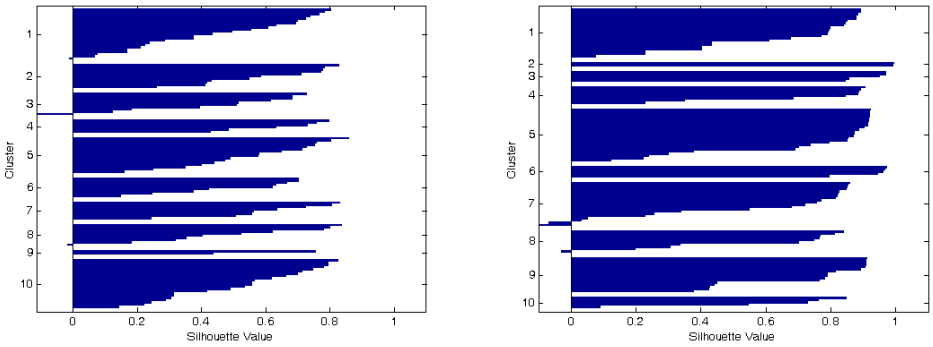
Figure 1. The Silhouette plots for clusters of high dimensional data with various values of *K*, with the squared Euclidean distance measure [(Figures. 1(a), (c) - (128D) and (f) - (50D)) and the Cosine similarity measure (Figures. 1(b) and (d) - (128D) and (e) - (50D))].

Every object x^i with a silhouette index close to 1 indicates it belongs to the cluster being assigned. A value of zero indicates that the object could also be assigned to another closest cluster. A value close to -1 indicates that the object is in a wrong cluster or somewhere in between the clusters. The highest value indicates the best clustering, meaning that the number of clusters selected for the clustering is the best [10].

5. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

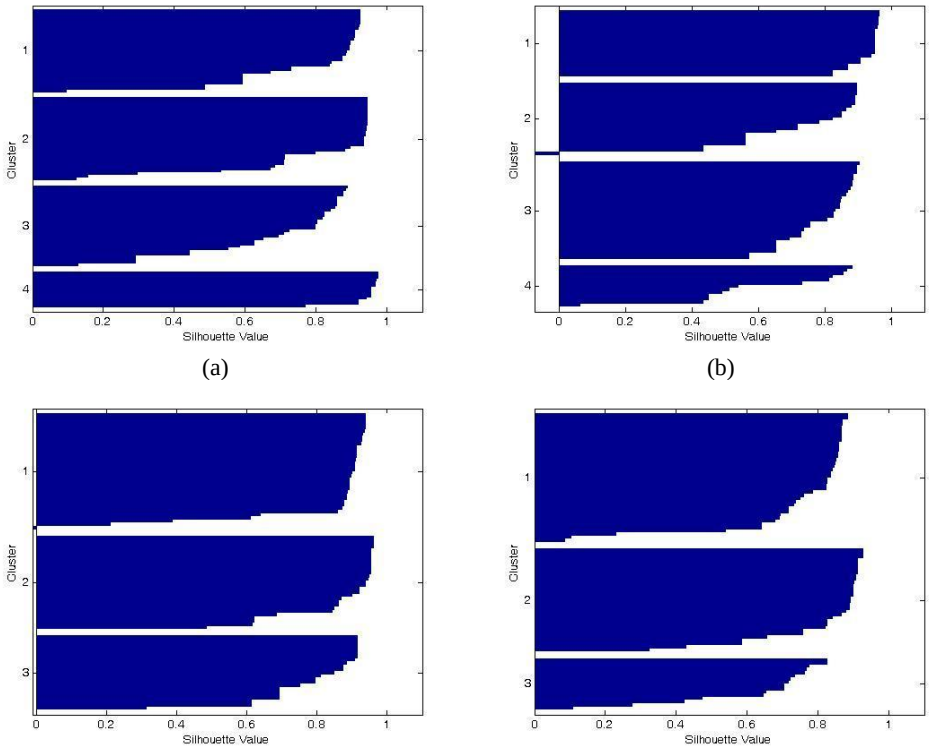
5.1 Application to high dimensional data

Firstly, *K*-means algorithm is applied on high dimensional data sets using 6 data sets. Each data set contains data elements (sequences) of equal length, due to the truncation mentioned earlier. The *K*-means algorithm was applied more than once on a data set to see the effect of *K* in the clustering process. Results of this investigation are presented in Table 1. In Table 1, the following symbols are used: i (data set), *N* (size of data set), *D* (dimension), *K* (number of clusters), I_E (number of iterations required when using squared Euclidean distance), I_C (number of iterations required when using cosine similarity measure), Td_E (distance using squared Euclidean), Td_C (distance using cosine measure), Sh_E (silhouette mean under Euclidean distance) and Sh_C (silhouette mean under cosine measure). The data in columns under 'Total distance' are the total intra cluster distance from the centroid of formed



(a) (b)

Figure 2. The silhouette plots for clusters derived for reduced dimension e.g. for D_r=2 and without preprocessor: K = 10, N = 117, [(a) Euclidean, (b) Cosine].



(a) (b)

(c) (d)

Figure 3. The silhouette plots for clusters of 100 data points (data set 4)[(a) K_O= 4 (Cosine measure) and (b) K_O=4 (squared Euclidean measure)] and 88 data points (data set 10)[(c) K_O=3 (cosine measure), and (d) K_O=3 (squared Euclidean measure)]

clusters[1]. That is if there are three clusters and d_{ij} is the distance the i^{th} element (of the j^{th} cluster S_j with n_j elements) and its centroid C^j , then the total is calculated over the three centroids of clusters of the data set, generated during the iteration process. This means that the total sum of distance is the value realized at the last iteration when the algorithm reaches a minimum, and the total is calculated over the set

$$Q_i = \left\{ \sum_{i=1}^{n_1} d_{1i}, \sum_{i=1}^{n_2} d_{2i}, \sum_{i=1}^{n_3} d_{3i} \right\}.$$

The data in columns under 'Silhouette mean' are the average of the silhouette values. For example, the average silhouette index values for the m^{th} cluster is given by

[1] The total sum of distances decreases at each iteration as *K*-means reassigns points between clusters and recomputes cluster centroids.

$$Q(m) = \frac{1}{n(m)} \sum_{i=1}^{n(m)} q_i(m),$$

where $q_i(m)$ is the silhouette value for the i^{th} member of the m^{th} cluster, and $n(m)$ is the total number of elements in the m^{th} cluster. The values presented in the last two columns in Table 1 are therefore the values $\frac{1}{K}\sum_{m=1}^{K} Q(m)$,where K is the total number of clusters.

To see the effect of K, the 4th and 5th major columns (Total sum of distances and Silhouette mean) in Table 1are studied. The total sum of distances should be as low as possible – a better clustering should give a lower value of the total sum of distances. However, these values seem quite high. The next was to study the silhouette means in Table 1. These values determine how far apart the clusters are. With a high value, the cluster quality is near optimal. Again these values seem non-optimal. To visualize the cluster-wise silhouette index values, the silhouette plot of generated clusters is present in Figure 1. For this, data sets 1 and 3 in Table 1 are used. Figures 1(a) and 1(c) are, respectively, for K =5 and 10 using squared Euclidean measure, data set 1. Figures 1(b) and 1(d) are, respectively, for K =5 and 10 using cosine measure, data set 1. Figures 1(e) and 1(f) are for the data set 3 using cosine and Euclidean measures respectively. Figure 1 clearly shows that the many silhouette values are negative and the overall results are unsatisfactory. Observe from Table 1 and Figure 1 that as the dimension decreases, the result of the clustering becomes better. It is clear from the figures that the value of K greatly determines the cluster quality.

5.2 The introduction of preprocessor to the K - means algorithm

To deal with the initialization problem of K, an automatic initialization scheme is proposed. The silhouette mean under Euclidean distance measure, Sh_E, presented in Table 1 plays the most important role in the scheme. For an initial value of K, provided by the user, the Algorithm 1 (the K-means algorithm) is run for a small number of iterations (e.g. typically 3) three times, respectively using $K-2$, K and $K+2$. Three corresponding Sh_E values corresponding to $K-2$, K and $K+2$ are found (hereafter denoted as $Sh_E(K-2)$, $Sh_E(K)$ and $Sh_E(K+2)$, respectively). The initial value, K_O, of K is then assigned using the following procedure:

1. If $Sh_E(K-2) < Sh_E(K)$ and $Sh_E(K) > Sh_E(K+2)$ then the Algorithm 1 is run again twice (each time for 3 iterations) using $K+1$ and $K-1$ and the corresponding $Sh_E(K-1)$ and $Sh_E(K+1)$ are found. The maximum value of three $\{Sh_E(K-1), Sh_E(K)$ and $Sh_E(K+1)\}$ then determines K_O. For example if $Sh_E(K-1)$ is the maximum then $K_O = K-1$.
2. If $Sh_E(K+2) > Sh_E(K)$ and $Sh_E(K+2) > Sh_E(K-2)$ then the Algorithm 1 is run again using $K+1$, $K+3$ and $K+4$. The K value corresponding to the maximum in $\{Sh_E(K+1), Sh_E(K+2), Sh_E(K+3), Sh_E(K+4)\}$ is then assigned to K_O.
3. If $Sh_E(K+2) < Sh_E(K-2)$ and $Sh_E(K) < Sh_E(K-2)$ then the value corresponding to the maximum in $\{Sh_E(K-1), Sh_E(K-2), Sh_E(K-3), Sh_E(K-4)\}$ is then assigned to K_O.

The initial value, K_O , of K found using the above procedure is then used to find K_O clusters using K-means algorithm, i.e. the Algorithm 1. To test the effectiveness of the above procedure two data sets from Table 1 are used namely, the first and the third data sets. The initial K in the preprocessor is used as given in Table 1. Results obtained are presented in Table 2. Table 2 clearly shows that the results have been improved for both data sets, although the problem dimension is very high.

5.3 Application to reduced dimensional data without preprocessor

The K-means algorithm is further tested with preprocessor scheme using the data sets presented in Table 1, but with reduced dimensions (D_r). Data sets of dimension two obtained by equation (5) are first tested with the data sets presented in Table 1 and present the results obtained in Table 3. To see the effect of reduced dimensionality the preprocessor is not incorporated in this experiment. The same K values as in Table 1 were used as this will allow a direct comparison of Tables 1 and 3. The results in Table 3 show significant improvement in all data sets with high silhouette means than those in Table 1. Two figures are presented, both for the data set 1 (K =10), corresponding to two different measures. Figures 2(a) and (b) correspond to the corresponding Figures 1(a) and (c). This comparison also establishes positive effect of dimension reduction.

5.4 Application to reduced dimensional data with preprocessor

The next is to study the effect of both reduced dimension and preprocessor on 10 data sets. Firstly, with Dr =2 the results obtained are presented in Table 4, where the first 6 data sets considered before are used.

To see the effect of the dimension reduction the same data sets in Tables 2 and 4 are compared, i.e. the data set 1 in Tables 2 and 4. Results shows that K_o corresponding to this data set in both tables are very close. This proves the effect of preprocessor as well as the dimension reduction in the K-means for categorical biological data sets. Notice that for data set 8 K and K_o are the same. This means that the initial K assigned to preprocessor remained the same.

Furthermore, the silhouette values is presented in Figure 3 for two data sets of 109 and 88 data points, respectively, with reduced dimensions. These are respectively the 4th and 10th data sets presented in Table 4. These figures clearly show well separated clusters. The usefulness of the silhouette value in the clustering task as well as the incorporation of the preprocessor is now evident.

An obvious question that one may rise is how to identify an appropriate value for the reduced dimension, D_r. To address this question, the values of dataset 1 in Table 3 are reproduced using D_r =3. Results obtained are very similar. For example, for K =10 the following values are obtained: Td_E =4448.01, Td_C =0.1872, Sh_E =0.4610, and Sh_C =0.6762. The corresponding graph for K =10 is presented in Figure 4. In addition, a graph for dataset

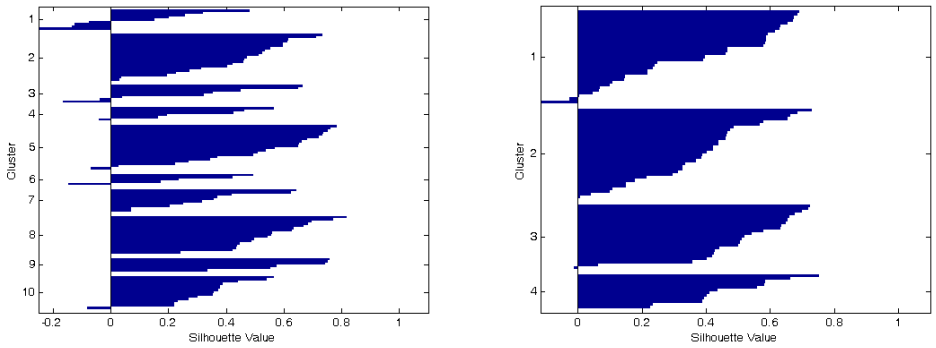
3 is also presented. Although preprocessor has not been used for this experiment, the graphs produced show that silhouette values are fairly acceptable. These results can be further improved by the use of preprocessor. Our experiments have shown that the optimized values are not exactly the same, for D_r =2 and 3, but they are within an acceptable level of

Table 2. Effects of preprocessor in the K-means algorithm applied to the high dimensional data

					Iterations		Total distance		Silhouette mean	
i	N	D	K	K_o	I_E	I_C	Td_E	Td_C	Sh_E	Sh_C
1	117	128	6	8	102	7	14576.1	7.8258	0.0437	0.0415
3	100	50	4	5	6	5	3052.75	3.05384	0.1163	0.1143

Table 3. Effects of reducing D in the K-means clustering, D_r=2

				Iterations		Total distance		Silhouette mean	
i	N	Previous D	K	I_E	I_C	Td_E	Td_C	Sh_E	Sh_C
1	117	128	5	8	12	7021.1	0.0110784	0.5377	0.7014
	117	128	10	16	6	3240.82	0.00461887	0.5184	0.6808
2	117	198	5	12	9	10019.5	0.00600523	0.5879	0.7000
	117	198	10	9	12	5678.06	0.00242334	0.5118	0.6507
3	100	50	4	15	5	2461.83	0.0221821	0.5103	0.7254
	100	50	6	9	13	1697.71	0.0106729	0.4941	0.7253
4	50	50	4	9	4	1018.37	0.0074717	0.5014	0.7700
	50	50	5	6	4	815.016	0.00648176	0.5038	0.7492
5	50	20	5	4	4	291.551	0.0144597	0.4524	0.6780
	50	20	4	13	5	350.375	0.0275186	0.4736	0.6206
6	20	50	4	3	3	184.571	0.00327066	0.6619	0.5042
	20	50	3	2	6	329.19	0.00360131	0.5612	0.7446



(a) (b)

Figure 4. The silhouette plots for clusters of 117 data points (data set 1)[(a) K= 10 (Euclidean measure) and (b) K=4 (Euclidean measure)] and 100 data points (data set 3)

Table 4. Optimal K_o in the K-means algorithm

				Iterations		Total distance		Silhouette mean	
i	N	K_o	K	I_E	I_C	Td_E	Td_C	S_E	S_C
1	117	7	4	6	7	3940.84	0.146286	0.6587	**0.8110**
2	117	6	8	5	5	5767.98	0.608631	0.6837	0.7358
3	100	5	7	5	7	6142.24	0.453146	0.6687	0.7923
4	50	4	6	9	6	7338.64	0.625917	**0.7508**	0.7859
5	50	6	3	10	7	12943.3	0.967989	0.7246	0.8021
6	20	7	3	13	7	3322.21	0.137556	**0.7537**	0.7898
7	88	6	7	5	9	4017.84	0.148084	0.6869	0.8072
8	88	5	5	5	4	4210.39	0.319518	0.7433	**0.8143**
9	88	4	3	6	5	8646.92	0.46525	0.6207	0.7772
10	88	3	6	6	3	9514.41	0.64087	**0.7546**	**0.8348**

closeness. Hence, it is suggested that D_r =2 is a good value to choose.

6. CONCLUSION AND FURTHER RESEARCH

The usefulness of the *K*-means algorithm clustering algorithm in the clustering of high- and multi-dimensional data have been studied. The clustering algorithm was applied to the categorical biological sequence data. These sequences consist of alphabets and are high- and multi- dimensional in nature. The study conducted in this paper regarding the dimension of data involved in a clustering task is, to the best of our knowledge, the first conducted on *K*-means clustering algorithm since it was developed.

Series of procedures were followed. The introduction of a numerical equivalence of sequences of the categorical data was done. To reduce the effect of initial *K* in *K*-means a preprocessor scheme was introduced. It has been shown that significant gains in optimality can be achieved by using the preprocessor. In addition, a dimension reduction technique was introduced which when applied with the preprocessor produces well separated clusters. Results presented in the paper further establish that the *K*-means algorithm is not suitable in the clustering of biological sequences without con-

ducting a character-to-numeric conversion to transform the nucleic/amino acids symbols to numeric values. Although the silhouette index is not new, the clustering technique introduced in the paper is new and thus can be applied to many practical problems. A further work in this direction would be the application of preprocessor with the *K*-means in the clustering of complex data structures.

7. REFERENCES

[1] P. Berkhin, 2002. "Survey of clustering data mining techniques," Accrue Software, Inc., San Jose, California, Tech. Rep. 4, available online: www.citeseer.nj.nec.com/berkhin02survey.html.

[2] D. A. Binder, 1977. "Cluster analysis under parametric models," PhD thesis, University of London.

[3] P. Hansen and B. Jaumard, 1997. "Cluster analysis and mathematical programming," in 16th International Symposiium on Mathematical Programming, vol. 79, pp. 191–215.

[4] J. B. MacQueen, 1967. "Some methods for classification and analysis of multivariate observations," in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, pp. 281–297.

[5] F. D. Smet, J. Mathys, K. Marchal, G. Thijs, B. D. Moor, and Y. Moreau, 2002. "Adaptive quality-based clustering of gene expression profiles," Bioinformatics, vol. 18, no. 6, pp. 735–748.

[6] J. Tou and R. Gonzalez, 1974. Pattern Recognition Principles. Massachusetts, USA: Addison-Wesley,.

[7] K. Huang, 2002. "A synergistic automatic clustering technique (SYN- ERACT) for multispectral image analysis," Photogrammetric Engineering and Remote Sensing, vol. 1, no. 1, pp. 33–40.

[8] J. Tou, 1979. "DYNOC - a dynamic optimal cluster-seeking technique," International Journal of Computer and Information Sciences, vol. 8, no. 6, pp. 541–547.

[9] C. Rosenberger and K. Chehdi, 2000. "Unsupervised clustering method with optimal estimation of the number of clusters: Application to image segmentation," in Proceedings of the International Conference on Pattern Recognition (ICPR'00), vol. 1, pp. 1656–1659.

[10] P. Rousseuw, 1987. "Silhouettes: a practical aid to the interpretation and validation of cluster analysis," Computational and applied mathematics, vol. 20.

[11] J. C. Bezdek, 1980. "A convergence theorem for the fuzzy ISODATA clustering algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, pp. 1–8.

[12] R. H. Turi, 2001. "Clustering-based colour image segmentation," PhD thesis, Monash University.

[13] M. G. H. Omran, 2004. "Particle swarm optimization methods for pattern recognition and image processing," PhD thesis, Univer- sity of Pretoria, Faculty of Engineering, Built Environment and Information Technology, Department of Computer Science, Nov.

[14] R. Xu and D. Wunsch II, 2005. "Survey of clustering algorithms," IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 601–614

[15] S. K. Gupta, K. S. Rao, and V. Bhatnagar, 1999. "K-means clustering algorithm for categorical attributes," in Proceedings of 1st Inter- national Conference on Data Warehousing and Knowledge Discovery, Florence, Italy, pp. 203–208.

[16] W. Z. Altun, G. Harrison, R. Tai, and P. C. Yi Pan, 2005. "Improved k- means clustering algorithm for exploring local protein sequence motifs representing common structural property," IEEE Trans. on Nanobioscience, vol. 4, no. 3, pp. 255–265.

[17] A. P. Gasch and M. B. Eisen, 2002. "Exploring the conditional correlation of yeast gene expression through fuzzy *k*-means clustering," Genome Biology, vol. 3, no. 11.

[18] K. F. Han and D. Baker, 1995. "Recurring local sequence motifs in proteins," Journal of Molecular Biology, vol. 251, pp. 176–187.

[19] MATLAB, 2004. The Language of Technical Computing. The Mathworks, Inc., May vol. version 7.0.

[20] L. Kaufman and P. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley.

[21] F. Azuaje, 2002. "Cluster validity framework for genome expression data," Bioinformatics, vol. 18, no. 2.

[22] M. D. G. Teledo, 2005. "A comparison in cluster validation techniques," Master of Science thesis, University of Puerto Rico, Department of Mathematics (Statistics).

[23] N. Bolshakova and F. Azuaje, 2003. "Cluster validation techniques for genome expression data," Signal Processing, vol. 83, pp. 825–833.

[24] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender