JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 10, No. 1, March 2016, pp. 21-31

Data-Compression-Based Resource Management in Cloud Computing for Biology and Medicine

Changming Zhu*
College of Information Engineering, Shanghai Maritime University, Shanghai, China
cmzhu@shmtu.edu.cn

Abstract

With the application and development of biomedical techniques such as next-generation sequencing, mass spectrometry, and medical imaging, the amount of biomedical data have been growing explosively. In terms of processing such data, we face the problems surrounding big data, highly intensive computation, and high dimensionality data. Fortunately, cloud computing represents significant advantages of resource allocation, data storage, computation, and sharing and offers a solution to solve big data problems of biomedical research. In order to improve the efficiency of resource management in cloud computing, this paper proposes a clustering method and adopts Radial Basis Function in order to compress comprehensive data sets found in biology and medicine in high quality, and stores these data with resource management in cloud computing. Experiments have validated that with such a data-compression-based resource management in cloud computing, one can store large data sets from biology and medicine in fewer capacities. Furthermore, with reverse operation of the Radial Basis Function, these compressed data can be reconstructed with high accuracy.

Category: Smart and intelligent computing

Keywords: Biomedical data; Cloud computing; Data compression; Data reconstruction

I. INTRODUCTION

Specially, in the last four decades, with the application and development of biomedical techniques such as next-generation sequencing, mass spectrometry, and medical imaging, biomedical data has experienced explosive growth. In 2009, the total data volume of public health of United States was 434 PB (where 1 PB = 1,024 TB). Having a 35% growth rate [1], this value is now larger than 2,000 PB. Most of this data consists of medical video and medical health archives which are required to be stored for an extended period of time—which presents a significant data management challenge [2-5].

In order to process and store the large data sets of biology and medicine, parallel computing is constantly adopted. Parallel computing is a form of computation in which many calculations are carried out simultaneously. With parallel computing, large problems can often be divided into smaller ones, and these smaller problems can be solved concurrently with multiple processors. Furthermore, with parallel computing, data can be stored in multiple computers. Traditional solutions in parallel computing include MPI and grid computing amongst others [6, 7]. In the last ten years, based on the concept of parallel computing, the proposal of cloud computing allows data to be processed in cloud clients. Cloud computing simplifies the usage of some information technology (IT) resources including computation and storage. Moreover, cloud computing provides flexible computing capabilities and efficient data analysis methods so that the elastic demands of

Open Access http://dx.doi.org/10.5626/JCSE.2016.10.1.21 http://jcse.kiise.org

Received 14 June 2015; **Revised** 16 February 2016; **Accepted** 25 February 2016
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

computing resources in biologic and medical laboratories can be solved. Such an advantage makes the scholars in biologic and medical fields pay more attention to cloud computing [8-11].

Cloud computing has many definitions, such as a widely used definition from the National Institute of Standards and Technology (NIST) in which "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models" [12]. With cloud computing, one can share and use software, platforms, and infrastructures from cloud clients in almost any location. Cloud computing has a strong ability to manage resources, which consist of 1) computers, servers, and spaces and 2) data. Alternatively, data storage and analysis is the main function of resource management in cloud computing. If cloud clients have high data process abilities, data can be transmitted and stored quickly, analyzed efficiently, and more hidden information can be mined.

Since many biologic and medical data sets are big data, general computers and servers cannot store and process these data sets well. More scholars have come to adopt resource management in cloud computing to process and store biologic and medical data sets. But with the application and development of biomedical techniques, more data will be kept in cloud clients for an extended period of time. If we do not take measures to compress data, we will encounter the problem of a capacity crisis, i.e., storage of data by cloud computing might cost more capacity than the original data set due to resource management functions. Furthermore, when data is desired, we must reconstruct the data to acquire the original information of data. Without a good reconstruction method, the authenticity of the original data will decrease. Some scholars have proposed methods for biomedical data sets, but unfortunately these methods encounter challenges and problems. For example, Grumbach and Tahi [13, 14] proposed a BioCompress method and an improved BioCompress-2 method to compress deoxyribonucleic acid (DNA) and Chen et al. [15] proposed a GenCompress method. But with the enhancement of data, traditional compression methods cannot process the compression of data. Methods classified as substitutional compressions and statistical compressions have been proposed with the main idea of substitutional compressions to have high similarities between the original data and substitutional data. Some substitutional compression methods including DNAPack [16], CTW+LZ [17], and DNADP [18] have been proposed. The main idea of statistical compressions is that the appearances of biologic and medical data have certain probabilities. A widely used statistical compression method is GeNML [19], but such substitutional and statistical compression methods still have limitations, i.e., they can compress big data while they cannot reconstruct them in a high quality.

To find a good method to compress and reconstruct data, this paper refers to the concepts of clustering and proposes a new clustering method. With the new proposed clustering method, this paper adopts Radial Basis Function (RBF) to compress the data and with the reverse operation of RBF to reconstruct the compressed data. The proposed method is called data-compression-based resource management in cloud computing (DRM) and experiments on some biologic and medical data sets will validate the effectiveness.

II. RELATED WORK

Since DRM is proposed on the base of clustering, this section will review clustering below.

Clustering is a method to divide a data set into several subsets which have high intra-subset similarities and low inter-subset similarities, with each subset called a cluster. *K*-means clustering (*K*-means) [20], agglomerative hierarchical clustering (AHC) [21], and kernel clustering (KC) [22] are three classical clustering approaches. With clustering, one can mine local structures and information of data in an optimized fashion.

As for these three clustering approaches, there are some differences among them. First, they have different frameworks. For *K*-means, a value *K* predefines the number of clusters. For AHC, each datum is made to start in its own cluster, and then pairs of clusters are merged as each moves up the hierarchy until a feasible number of clusters is attained. For KC, one datum is selected as an initial cluster, and then the cluster grows up to cover object datum one by one until this cluster will cover any outlier if it grows continually. Second, KC is a supervised clustering approach, but *K*-means and AHC are not. This means that if the labels of data are known, the clusters derived from KC will be better than those from *K*-means and AHC. Third, *K*-means has a computational complexity which is $O(KNI)$ where N is total number of data, K is the number of clusters, and I is the number of iterations. AHC has a high computational complexity which is at least $O(N^2)$ [21], and generally $O(N^3)$. For KC, the computational complexity is $O(N^2)$. Fourth, the complexity of *K*-means is sensitive to the setting of K with different Ks bringing different results. While for AHC and KC, the results are decided by the structural relationship between data which is unique for a data set.

III. DATA-COMPRESSION-BASED RESOURCE MANAGEMENT IN CLOUD COMPUTING

In order to improve the efficiency of resource manage-

ment in cloud computing when large data sets are stored and processed, especially for biology and medicine, we propose a new clustering method. RBF was adopted so as to reduce the dimensionalities of data as soon as possible. With the reduction of dimensionalities, fewer capacities can be used to store data. This section consists of four parts. First and second parts will give the framework of DRM, namely clustering and compression. The third part will show how to acquire the original information from these compressed data. The fourth part will show the differences between the proposed DRM and other compression technologies including BioCompress, BioCompress-2, GenCompress, DNAPack, DNADP, and GeNML.

A. The Proposed Clustering Method

Data in a biomedical data set should be quantized. The data set will consist of N data from p classes. Each datum is composed of d components. Namely, the data set is $A = \{x_i, \varphi_i\}$, $i = 1, 2, ..., N$, where $x_i \in R^d$ and $\varphi_i \in \{1, 2, ..., p\}$ is class label. We propose a new clustering method to divide this data set into several clusters. Definitions are provided in order to describe the process clearly. First, when we carry out the clustering for one class, this class is called object class. Data from this class are called object data. The clusters of this class are called object clusters. Otherwise, we call them as non-object classes, non-object data (or outliers), and non-object clusters. The procedure of this new proposed clustering method is given below.

1) Transform a p-class problem into p two-class problems. For one two-class problem, one class is treated as object class, and other classes are treated as a non-object class. Data in this non-object class are called outliers.

2) In each two-class problem, for the object class, ω_A, we treated the data which have not been covered by any object clusters as $x_1, x_2, ... x_m$. Then we compute the center of the clusters and regard this center, $\bar{x}$, as the initial center of one cluster.

3) Compute all distances between datum from $x_1, x_2, ... x_m$ and $\bar{x}$.

4) Sort these distances from small to large. Denote the corresponding object data as $\{x_{d(1)}, x_{d(2)}, ..., x_{d(i)}, x_{d(j)}, ..., x_{d(m)}, x_{d(m+1)}\}$ where $1 < 2 < ... < i < ... < j < ... < m$, and the distance between $x_{d(i)}$ and $\bar{x}$ is not larger than the one between $x_{d(j)}$ and $\bar{x}$.

5) Compute the nearest distance d_{N1b} between outliers and $\bar{x}$.

6) If d_{N1b} is smaller than the distance between $x_{d(j+1)}$ and $\bar{x}$ while larger than the one between $x_{d(j)}$ and $\bar{x}$, we put the object data $\{x_{d(1)}, x_{d(2)}, ..., x_{d(i)}, ..., x_{d(j)}\}$ into the cluster and this cluster has finished the task of generation. Then the distance between $\bar{x}$ and $x_{d(j)}$ is treated as the radius of this cluster, i.e., σ, and $\bar{x}$ is treated as the center of this cluster.

7) Use steps 2 to 6 to cover class ω_A by K_A kernels, until all data in this object class are covered by different clusters. Finish all p two-class problems. In that way, p group of clusters can be generated.

B. Compression of Data

Suppose after the previous step, for a data set with N data from p classes, we have M clusters, $C_1, ..., C_j, ..., C_M$ where $j = 1, 2, ..., M$. Center of a cluster C_j is treated as $\mu_j = (\mu_{j1}, \mu_{j2}, ..., \mu_{jd})$ while radius of it is treated as σ_j. Then we adopt RBF to compress these data.

We know that the expression of RBF is $\ker(x_i, x_j) = \exp\left(-\frac{\|x_i - x\|_2^2}{\sigma^2}\right)$. So for each datum of this data set, i.e., x_i, we introduce it into each cluster and get the computational result. Concretely speaking, for C_j and x_i, we denote the result as y_{ij} and $y_{ij} = \exp\left(-\frac{\|x_i - \mu_j\|_2^2}{\sigma_j^2}\right)$. Then for x_i and M clusters, we can get $y_{i1}, y_{i2}, ..., y_{iM}$ where $j = 1, 2, ..., M$. Now for each original datum $x_i = (x_{i1}, x_{i2}, ... x_{id})$, we can adopt $y_i = (y_{i1}, y_{i2}, ..., y_{iM})$ as the replacement. In this way, each d-dimensional datum can be represented as an M-dimensional datum.

Now, in terms of resource management in cloud computing, if the proposed method is not used, $N \times d$ capacities should be used to store these data. If the proposed clustering and compression method is adopted, only $N \times M + M \times d + M$ capacities are used. For $N \times M$, it denotes the demanded capacities for the replacements of the original data. For $M \times d$, it denotes M centers. For M, it denotes M radius. Since $M < d$, in general with the proposed method, we can use fewer capacities to store data.

C. Reconstruction of Data

In the previous two steps, large data sets could be compressed to save capacities for resource management in cloud computing. If the compressed data is reconstructed so as to get the original information from these data, the following method can be adopted. Indeed, this method is the reverse operation of RBF. Now we use an instance to describe that how to get the original information from a compressed datum.

For a compressed datum y_i, we know each component y_{ij} where $j = 1, 2, ..., M$ is computed with $y_{ij} = \exp\left(-\frac{\|x_i - \mu_j\|_2^2}{\sigma_j^2}\right)$. So

$$-\frac{\|x_i - \mu_j\|_2^2}{\sigma_j^2} = \ln(y_{ij}) \tag{1}$$

Then

$$\|x_i - \mu_j\|_2^2 = -\sigma_j^2 \ln(y_{ij}) \tag{2}$$

Since y_i is a M-dimensional datum, we have equations as below.

$$\begin{cases} \|x_i-\mu_1\|_2^2 = -\sigma_1^2 \ln(y_{i1}) \\ \|x_i-\mu_2\|_2^2 = -\sigma_2^2 \ln(y_{i2}) \\ \cdots \\ \|x_i-\mu_M\|_2^2 = -\sigma_M^2 \ln(y_{iM}) \end{cases} \tag{3}$$

For the solution of Eq. (3), we expand it as follows.

$$\begin{cases} (x_{i1}-\mu_{11})^2 + (x_{i2}-\mu_{12})^2 + \ldots + (x_{id}-\mu_{1d})^2 = -\sigma_1^2 \ln(y_{i1}) \\ (x_{i1}-\mu_{21})^2 + (x_{i2}-\mu_{22})^2 + \ldots + (x_{id}-\mu_{2d})^2 = -\sigma_2^2 \ln(y_{i2}) \\ \cdots \\ (x_{i1}-\mu_{M1})^2 + (x_{i2}-\mu_{M2})^2 + \ldots + (x_{id}-\mu_{Md})^2 = -\sigma_M^2 \ln(y_{iM}) \end{cases} \tag{4}$$

i.e.,

$$\begin{cases} x_{i1}^2 - 2x_{i1}\mu_{11} + \mu_{11}^2 + \ldots + x_{id}^2 - 2x_{id}\mu_{1d} + \mu_{1d}^2 = -\sigma_1^2 \ln(y_{i1}) \\ x_{i1}^2 - 2x_{i1}\mu_{21} + \mu_{21}^2 + \ldots + x_{id}^2 - 2x_{id}\mu_{2d} + \mu_{2d}^2 = -\sigma_2^2 \ln(y_{i2}) \\ \cdots \\ x_{i1}^2 - 2x_{i1}\mu_{M1} + \mu_{M1}^2 + \ldots + x_{id}^2 - 2x_{id}\mu_{Md} + \mu_{Md}^2 = -\sigma_1^2 \ln(y_{iM}) \end{cases} \tag{5}$$

For any two equations in Eq. (5), we adopt one to minus another one and get the following result.

$$\begin{aligned} &2(-x_{i1}\mu_{k1} + x_{i1}\mu_{j1} - \ldots - x_{id}\mu_{kd} + x_{id}\mu_{jd}) + \mu_{k1}^2 - \mu_{j1}^2 + \\ &\ldots + \mu_{kd}^2 - \mu_{jd}^2 = -\sigma_k^2 \ln(y_{ik}) + \sigma_j^2 \ln(y_{ij}) \end{aligned} \tag{6}$$

Then according to the Eq. (5), we can get a new equation set with $M(M-1)/2$ equations as below.

$$\begin{cases} 2(-x_{i1}\mu_{11} + x_{i1}\mu_{21} - \ldots - x_{id}\mu_{1d} + x_{id}\mu_{2d}) + \mu_{11}^2 - \mu_{21}^2 + \\ \ldots + \mu_{1d}^2 - \mu_{2d}^2 = -\sigma_1^2 \ln(y_{i1}) + \sigma_2^2 \ln(y_{i2}) \\ 2(-x_{i1}\mu_{11} + x_{i1}\mu_{31} - \ldots - x_{id}\mu_{1d} + x_{id}\mu_{3d}) + \mu_{11}^2 - \mu_{31}^2 + \\ \ldots + \mu_{1d}^2 - \mu_{3d}^2 = -\sigma_1^2 \ln(y_{i1}) + \sigma_3^2 \ln(y_{i3}) \\ \cdots \\ 2(-x_{i1}\mu_{11} + x_{i1}\mu_{M1} - \ldots - x_{id}\mu_{1d} + x_{id}\mu_{Md}) + \mu_{11}^2 - \mu_{M1}^2 + \\ \ldots + \mu_{1d}^2 - \mu_{Md}^2 = -\sigma_1^2 \ln(y_{i1}) + \sigma_M^2 \ln(y_{iM}) \\ \cdots \\ 2(-x_{i1}\mu_{k1} + x_{i1}\mu_{j1} - \ldots - x_{id}\mu_{kd} + x_{id}\mu_{jd}) + \mu_{k1}^2 - \mu_{j1}^2 + \\ \ldots + \mu_{kd}^2 - \mu_{jd}^2 = -\sigma_k^2 \ln(y_{ik}) + \sigma_j^2 \ln(y_{ij}) \\ \cdots \\ 2(-x_{i1}\mu_{(M-1)1} + x_{i1}\mu_{M1} - \ldots - x_{id}\mu_{(M-1)d} + x_{id}\mu_{Md}) + \mu_{(M-1)1}^2 \\ -\mu_{M1}^2 + \ldots + \mu_{(M-1)d}^2 - \mu_{Md}^2 = -\sigma_{(M-1)}^2 \ln(y_{i(M-1)}) + \sigma_M^2 \ln(y_{iM}) \end{cases} \tag{7}$$

where $k, j = 1, 2, 3.,,, M$ and $k \neq j$. For Eq. (7), we can rewrite it as a matrix form, i.e.,

$$AX = B \tag{8}$$

where

$$A_{M(M-1)/2\times d} = \begin{pmatrix} -2\mu_{11} + 2\mu_{21} & \ldots & -2\mu_{1d} + 2\mu_{2d} \\ -2\mu_{11} + 2\mu_{31} & & -2\mu_{1d} + 2\mu_{3d} \\ & \ldots & \\ -2\mu_{(M-1)1} + 2\mu_{M1} & & -2\mu_{(M-1)d} + 2\mu_{Md} \end{pmatrix} \tag{9}$$

$$X_{d\times 1} = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \cdots \\ x_{id} \end{pmatrix} \tag{10}$$

$$B_{M(M-1)/2\times 1} = \begin{pmatrix} -\sigma_1^2 \ln(y_{i1}) + \sigma_2^2 \ln(y_{i2}) - (\mu_{11}^2 - \mu_{21}^2 + \ldots + \mu_{1d}^2 - \mu_{2d}^2) \\ -\sigma_1^2 \ln(y_{i1}) + \sigma_3^2 \ln(y_{i3}) - (\mu_{11}^2 - \mu_{31}^2 + \ldots + \mu_{1d}^2 - \mu_{3d}^2) \\ \\ -\sigma_{(M-1)}^2 \ln(y_{i(M-1)}) + \sigma_M^2 \ln(y_{iM}) - (\mu_{(M-1)1}^2 - \mu_{M1}^2 + \ldots + \mu_{(M-1)d}^2 - \mu_{Md}^2) \end{pmatrix} \tag{11}$$

In terms of Eq. (8), if *rank(A) = rank(A,B) = d*, the solution of Eq. (8) is unique and we can get the original information of x_i with a high quality. If *rank(A) = rank(A,B)*, we can get multiple solutions of x_i. Under such a case, we can reconstruct the compressed datum x_i but we cannot promise the primitiveness of information. Moreover, if $rank(A) \neq rank(A,B)$, we cannot reconstruct the compressed datum x_i. In terms of solving the Eq. (8), the method can be found in any book about matrix theory. Here, *rank(A)* is the rank of a matrix A.

In order to show the framework of the proposed method, with the help of a physician, we adopt lung images as an example shown in Fig. 1. To protect the confidentiality of these images, the sources of these images are not opened. The file contains 309,126 images in total, and some denote lungs with pulmonary cystic lesions, some denote the lungs with cancers, and others denote healthy lungs. From Fig. 1, we can see some images from the 309,126 images, the compression results of all 309,126 images, and the related reconstruction results. In terms of this instance, all images were found to be compressed into 3-dimensional numeric data rather than 3-dimensional images. In order to store images, more capacities must be used for storing values. From this figure, the reconstructed images are found to keep most of information of original images.

D. The Differences between DRM and Other Traditional Compression Technologies

There are many traditional compression technologies such as BioCompress, BioCompress-2, GenCompress, DNAPack, DNADP, and GeNML that can be used to compare

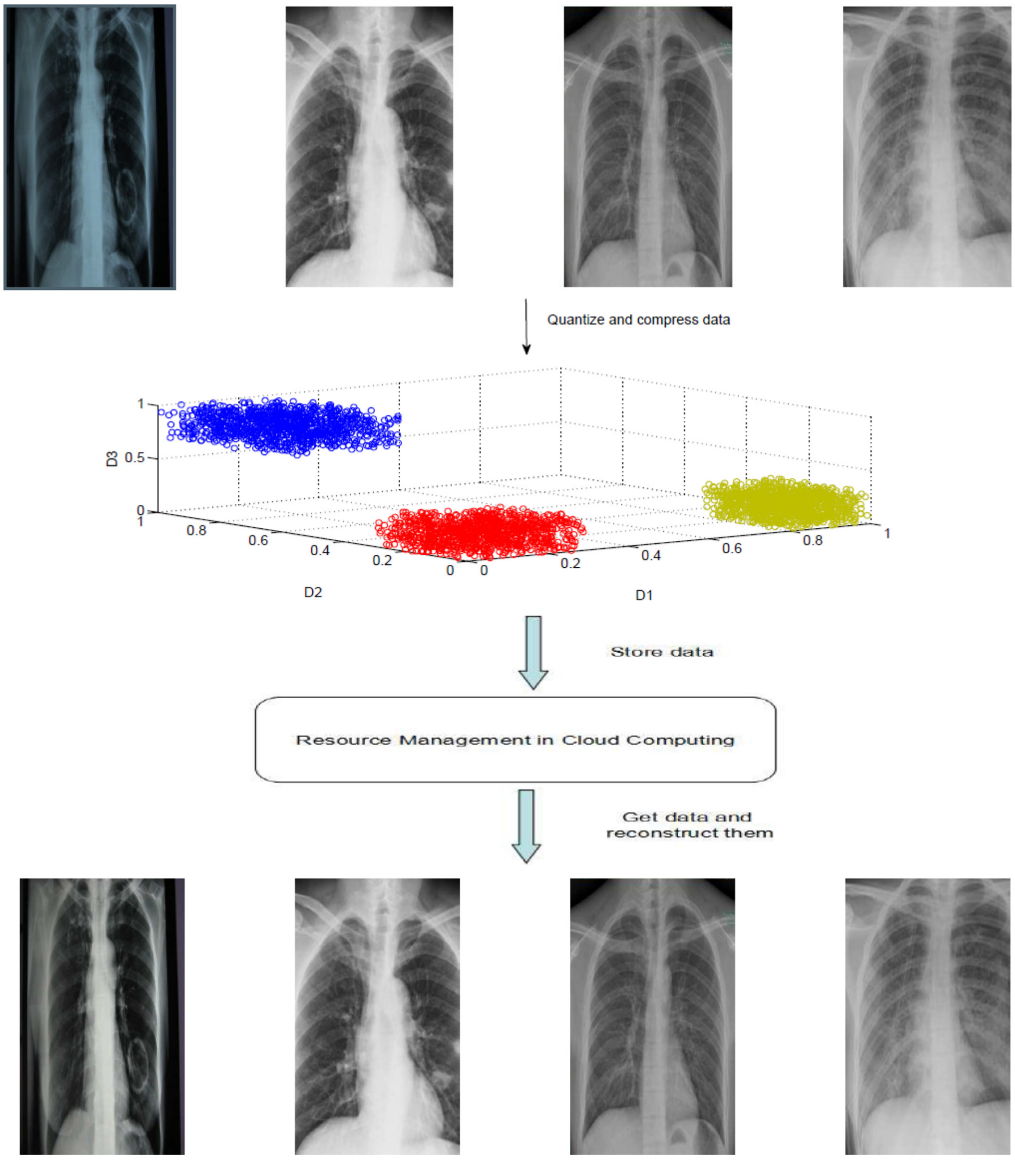

Fig. 1. The framework of DRM (data-compression-based resource management in cloud computing). The first row shows some images from those 309,126 ones. Second row shows the compression results of the original data, here blue circles denote the lungs with pulmonary cystic lesions, red circles denote the lungs with cancers, other circles denote healthy lungs. Third row shows that these compression data are stored in the cloud. Fourth row shows the results of data reconstruction for the images.

performance with DRM. These compression technologies aim to compress a DNA sequence or genome, while the proposed DRM aims to compress multiple data from one dimensional space to another dimensional space with clustering and RBF. In order to show the difference between DRM and other traditional compression technologies, the following instance was adopted.

Here, we have many DNA sequences, each DNA sequence consists of four alphabet letters {A, C, G, T}, for example 'AACTGTTGTTGTTAGAACTGTTGTT'. For BioCompress, BioCompress-2, GenCompress, DNAPack, DNADP, and GeNML the original sequence is encoded with 1 and 0. For instance, 'AACTGTTGTTGTTAGAAC TGTTGTT' is encoded in binary notation of '10011 00 00 01 11 10 11 11 00000 100 011 00 10 11 00011 110'. These traditional compression technologies do other operations on the binary sequence so a shorter sequence can be used to replace the original binary sequence, compressing the DNA sequences or genomes to save capacities.

But for the proposed DRM, we compress these DNA sequences with a different method. First, we quantize the sequences, for example using the four DNA sequences, 'AACTGTT', 'TTGTTAC', 'TGTTACA', and 'GTTGAAC' with the quantized results being $x_1 = (10,00,00,01,11,10,11)$, $x_2 = (10,11,10,01,11,01,11)$, $x_3 = (10,11,01,00,11,00,10)$, and $x_4 = (00,01,11,11,00,01,10)$, respectively. These quan-

tized DNA sequences are clustered and if there are two clusters, one may have a center μ_1 = (01,10,01,01,10,01,10) and a radius σ_1 = 0.9, and the other may have a center μ_2 = (10,01,00,10,11,10,01) and a radius σ_2 = 0.85. After forming two clusters, we use RBF and these two clusters to get the replacements of x_is where i = 1, 2, 3, 4; for example, y_1 = (0.31,0.93), y_2 = (0.09,0.82), y_3 = (0.75,0.23), and y_4 = (0.83,0.21), and then y_1, y_2, y_3, and y_4 can be used to replace the original x_1, x_2, x_3, and x_4, respectively. In order to store x_1, x_2, x_3, and x_4, the original capacities are 4 × 7 bytes, i.e., 28 bytes, but now only 4 × 2 + 2 × 7 + 2 = 24 bytes are needed.

Although DRM and other traditional compression technologies are both used to compress data, the compression methods are different. Furthermore, DRM can be used to compress multiple kinds of data while these traditional compression technologies can only be used to compress DNA sequences or genomes.

The proposed DRM may be used for the human genome and such application will be discussed here. The human genome includes 3 billion DNA base pairs and each pair consists of two nitrogenous bases. Because each pair has a complementary pairing with A-T or C-G, one letter can be used to store one pair. A, T, C, and G are always represented by '00', '01', '10', and '11', and thus one byte can store four pairs. We can use 750 M to store the human genome by using 0–1 sequences and the memory needed is not very large.

IV. EXPERIMENTS

Biology and medicine always generate large data issues and any cloud computing solutions will encounter a capacity crisis, so this paper proposes a DRM in cloud computing with a new clustering method. In order to show the effectiveness of the proposed DRM, we used other compression methods (BioCompress, BioCompress-2, GenCompress, DNAPack, DNADP, and GeNML) as the comparisons. These compared methods are aimed at compressing DNA or genomes, so we have only adapted DRM for such experiments. Furthermore, we adopt *K*-means, AHC, and KC for comparisons so as to validate the effectiveness of the proposed clustering method.

For the experiments biologic and medical data sets were selected, with some data sets including DNA sequences and genomes while others did not (Table 1). The data sets were selected from four databases: European Molecular Biology Laboratory (EMBL), DNA Data Bank of Japan (DDBJ), GenBank created by National Center for Biotechnology Information (NCBI), and the Simulated Brain Database (BrainWeb). Since the first three databases synchronize data daily, the information of data sets are the same in each database, with the difference being that the formats of data sets are different. Moreover, we also adopt the lung images which are used in the previous section for experiments.

SmartCloud Foundation of IBM, which includes SmartCloud Entry (SCE) solutions, IBM SmartCloud Provisioning (SCP) softwares, and IBM SmartCloud Monitoring, was used as the cloud platform.

The related experiments should include three comparisons: capacities, reconstruction ability, and time. If we adopt compression technologies, we can deal with the capacity crisis to some extent. If we hope to get original data from the platform of resource management in cloud

Table 1. Description of the used data sets

Order	Data set	Type	Size (TB)	Dimensionality	Number
1	HG02026	DNA & RNA	0.0086	1,000	202
2	P02533	DNA & RNA	0.0047	472	51,561
3	CHEMBL2113921	Chemical biology genomes	0.00521	1,000,000	5,200
4	Skin segmentation	Facts	0.00171	4	245,057
5	HIV-1 protease cleavage	Disease	0.00123	1	6,590
6	Diabetes 130-US hospitals for years 1999-2008	Medicine	0.00421	55	100,000
7	Lung	Disease image	1.06	3	309,126
8	Arabidopsis	Biology	1.40	5,021	370,293
9	Vectorized MRI	Medicine	4.50	1,203	5,000,000
10	Simulated brain MRI data	Medicine	2.21	251,002	11,788
11	Kazachstania	Medical biology genomes	3.38	421,466	10,716
12	Naumovozyma	Medical biology genomes	4.58	524,733	11,676
13	*Schizosaccharomyces*	Medical biology genomes	15.97	5,579,133	3,830
14	*Cryptococcus*	Medical biology genomes	23.66	679,007	46,606

MRI: magnetic resonance imaging.

computing, we should discuss the reconstruction ability, i.e., how much original information can be kept or reconstructed after the process of reconstruction. We also know that the cloud always stores many large data sets, so in order to compress, store, get, and reconstruct these large data sets, time spent is discussed.

A. Capacities Comparisons

We know that, in biology and medicine, large data sets must be stored for an extended period of time. If we do not adopt compression technologies, the resource management in cloud computing will encounter a capacity crisis in the future. So many compression technologies have been proposed, and here, we hope to show the effectiveness of the proposed method. The experimental content is to show the capacities needed to store these big data with different methods. In Table 2, the demanded capacities for different methods are given while Table 3 shows the reduced dimensionalities of the used data sets when they are compressed with these methods. From these tables, the proposed DRM is found to be able to compress these large data sets in a high rate. This also means that with DRM, a capacity crisis can be avoided in the future with a great probability. Furthermore, when *K*-means is adopted as the clustering method, the data sets can be compressed with higher rates, but according to further experiments, the *K*-means used could not reconstruct data with a high authenticity.

B. Reconstruction Ability

Compression of data is a major task of resource management in cloud computing as well as reconstruction of data. If the cloud cannot reconstruct data with a high authenticity, the storage and compression of data will not be useful. Since our proposed DRM has a function to reconstruct data, we discuss the reconstruction ability. The reconstructed data and the original data are compared to determine how much data has been reconstructed completely. Table 4 shows the results of this comparison such that the proposed method can reconstruct data in a high authenticity and accuracy and that the proposed DRM can save capacities without the damage and loss of those big data, while for other clustering methods used, data cannot be reconstructed with a high authenticity, especially for *K*-means high.

C. Time Comparisons

Time is an important index to assess the effectiveness of a method. In our work, if our proposed method can save capacities at the cost of time, then our proposal will lose value. So here, we carry out some experiments regarding time and make comparisons about time for the used methods on those used large data sets. The considered time should include time for storing data, time for compressing data (if available), time for reconstructing data (if available), and time for getting data. In terms of

Table 2. Capacities comparisons for the used methods on large data sets (unit: TB)

Data order	Original data	BioCompress	BioCompress-2	GenCompress	DNAPack	DNADP	GeNML	DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)
1	0.0086	0.00592	0.00572	0.00561	0.00501	0.00421	0.00318	0.00101	0.00033	0.00131	0.00119
2	0.0047	0.00303	0.00301	0.00291	0.00283	0.00281	0.00278	0.00072	0.00021	0.00089	0.00079
3	0.00521	0.00391	0.00385	0.00376	0.00365	0.00345	0.00332	0.00067	0.00002	0.00058	0.00068
4	0.00171	\	\	\	\	\	\	0.00021	0.00042	0.00032	0.00021
5	0.00123	\	\	\	\	\	\	0.00019	0.00057	0.00038	0.00019
6	0.00421	\	\	\	\	\	\	0.00051	0.00087	0.00058	0.00066
7	1.06	\	\	\	\	\	\	0.13	0.25	0.21	0.21
8	1.40	\	\	\	\	\	\	0.17	0.03	0.22	0.19
9	4.50	\	\	\	\	\	\	0.54	0.12	0.66	0.59
10	2.21	\	\	\	\	\	\	0.27	0.01	0.28	0.27
11	3.38	2.33	2.25	2.20	1.97	1.65	1.25	0.41	0.02	0.24	0.40
12	4.58	3.15	3.05	2.99	2.67	2.24	2.03	0.55	0.04	0.27	0.56
13	15.97	10.99	10.62	10.42	9.30	7.82	5.91	1.93	0.35	2.24	0.89
14	23.66	16.29	15.74	15.43	13.78	11.58	8.75	2.87	0.04	0.67	2.72

DRM: data-compression-based resource management in cloud computing; AHC: agglomerative hierarchical clustering, KC: kernel clustering.
Here \ denotes the corresponding method cannot compress the related data set and we have no related experimental results. DRM denotes the method we proposed in this paper, DRM (*K*-means) denotes the DRM adopts *K*-means as the clustering method, DRM (AHC) denotes the DRM adopts AHC as the clustering method, and DRM (KC) denotes the DRM adopts KC as the clustering method.

Table 3. Reduced dimensionalities of the used data sets when they are compressed

Data order	Original data	BioCompress	BioCompress-2	GenCompress	DNAPack	DNADP	GeNML	DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)
1	1,000	621	610	576	531	491	398	117	38	152	138
2	472	401	391	351	321	318	310	72	21	89	79
3	1,000,000	761,021	710,311	703,016	673,911	629,396	603,825	128,599	3,872	110,921	130,112
4	4	\	\	\	\	\	\	2	4	3	2
5	1	\	\	\	\	\	\	1	3	2	1
6	55	\	\	\	\	\	\	7	12	8	9
7	3	\	\	\	\	\	\	3	6	5	5
8	5,021	\	\	\	\	\	\	608	120	792	682
9	1,203	\	\	\	\	\	\	146	32	178	159
10	251,002	\	\	\	\	\	\	30,406	1,092	32,031	31,021
11	421,466	273,450	261,211	247,447	249,477	238,142	186,585	51,056	3,021	30,123	50,123
12	524,733	314,452	290,922	291,322	264,826	235,436	199,464	63,566	4,022	30,491	64,032
13	5,579,133	3,467,433	3,264,343	2,974,484	2,752,020	2,322,728	1,827,929	675,857	123091	782,123	309,213
14	679,007	420,952	416,255	397,418	371,035	350,346	286,678	82,255	1,093	19,321	78,212

DRM: data-compression-based resource management in cloud computing; AHC: agglomerative hierarchical clustering, KC: kernel clustering.
Here \ denotes the corresponding method cannot compress the related data set and we have no related experimental results. DRM denotes the method we proposed in this paper, DRM (*K*-means) denotes the DRM adopts *K*-means as the clustering method, DRM (AHC) denotes the DRM adopts AHC as the clustering method, and DRM (KC) denotes the DRM adopts KC as the clustering method.

Table 4. Reconstruction ability of the proposed method on those used big data sets

Data order	Original data (TB)	Authentic reconstructed data (TB)				Authentic rate (%)			
		DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)	DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)
1	0.00860	0.00858	0.00318	0.00680	0.00759	99.77	36.98	79.08	88.23
2	0.00470	0.00467	0.00183	0.00382	0.00442	99.36	38.91	81.20	94.12
3	0.00521	0.00520	0.00141	0.00411	0.00469	99.81	27.04	78.79	89.99
4	0.00171	0.00170	0.00079	0.00144	0.00161	99.42	46.04	84.43	93.92
5	0.00123	0.00122	0.00055	0.00098	0.00119	99.19	44.64	79.84	97.05
6	0.00421	0.00419	0.00169	0.00331	0.00382	99.52	40.04	78.66	90.73
7	1.06	1.05	0.52	0.89	0.90	99.06	49.33	83.81	84.66
8	1.40	1.38	0.90	1.12	1.34	98.57	64.19	79.97	95.55
9	4.50	4.47	1.82	3.77	4.04	99.33	40.50	83.87	89.82
10	2.21	2.16	0.71	1.68	2.04	97.74	32.02	76.02	92.41
11	3.38	3.31	1.00	2.66	2.95	97.93	29.68	78.68	87.30
12	4.58	4.23	1.34	3.44	3.87	92.36	29.32	75.07	84.51
13	15.97	15.23	6.30	11.69	14.25	95.37	39.47	73.17	89.22
14	23.66	23.23	5.63	18.80	19.91	98.18	23.79	79.46	84.13

DRM: data-compression-based resource management in cloud computing; AHC: agglomerative hierarchical clustering, KC: kernel clustering.

time for compressing data, if we carry out clustering, time for clustering will also be included in time for compressing data. The time comparisons for the used methods on those used large data sets are presented below in Tables 5–8. From these tables, the proposed method always brings a shorter time at higher speed for storing and getting large data sets. Specially, although DRM should compress and reconstruct data, the total time from storing data to getting data of DRM is still less than the time when we do not adopt any compression technology. Furthermore, compared with the proposed method when we adopt AHC or KC as the clustering method, the average time for compressing, getting, storing, and reconstructing data is longer.

Table 5. Time (hour) for storing large data sets

Data order	Original data	BioCompress	BioCompress-2	GenCompress	DNAPack	DNADP	GeNML	DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)
1	1.01	0.56	0.52	0.51	0.48	0.39	0.32	0.11	0.04	0.14	0.13
2	0.56	0.28	0.27	0.24	0.23	0.21	0.19	0.09	0.03	0.14	0.13
3	0.71	0.36	0.34	0.33	0.31	0.28	0.26	0.11	0.01	0.09	0.11
4	0.29	\	\	\	\	\	\	0.03	0.04	0.03	0.02
5	0.27	\	\	\	\	\	\	0.03	0.06	0.04	0.02
6	0.43	\	\	\	\	\	\	0.04	0.08	0.05	0.06
7	103.11	\	\	\	\	\	\	13.82	34.61	28.84	28.85
8	135.28	\	\	\	\	\	\	18.12	4.56	30.10	25.92
9	437.21	\	\	\	\	\	\	58.63	16.38	91.14	81.41
10	215.19	\	\	\	\	\	\	28.84	1.32	38.73	37.52
11	328.41	179.33	170.60	153.91	161.93	121.09	97.46	44.02	3.32	33.12	55.11
12	445.52	241.92	229.99	215.07	219.85	168.48	140.51	59.71	4.82	36.52	76.71
13	1553.75	880.14	776.48	758.17	745.62	634.74	523.89	208.01	48.37	307.33	121.50
14	2301.21	1279.56	1250.04	1211.13	1167.80	938.88	695.62	308.42	5.23	92.38	373.97

DRM: data-compression-based resource management in cloud computing; AHC: agglomerative hierarchical clustering, KC: kernel clustering.
Here \ denotes the corresponding method cannot compress the related data set and thus we need not have time for storing data when the corresponding method is used. Because the time should be same as the one in the second column. Namely, for the data sets which cannot be compressed by BioCompress, BioCompress-2, GenCompress, DNAPack, DNADP, and GeNML, whether to adopt these compression technologies, the time for storing data is same. DRM denotes the method we proposed in this paper, DRM (*K*-means) denotes the DRM adopts *K*-means as the clustering method, DRM (AHC) denotes the DRM adopts AHC as the clustering method, and DRM (KC) denotes the DRM adopts KC as the clustering method.

Table 6. Time (hour) for compressing large data sets

Data order	Original data	BioCompress	BioCompress-2	GenCompress	DNAPack	DNADP	GeNML	DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)
1	1.21	1.32	2.31	2.03	2.33	1.32	0.47	0.15	0.61	0.55	1
2	0.59	0.61	1.32	1.21	1.36	0.59	0.22	0.08	0.35	0.31	2
3	0.73	0.79	1.32	1.63	0.65	0.56	0.32	0.01	0.26	0.31	3
4	\	\	\	\	\	\	0.17	0.25	0.18	0.12	4
5	\	\	\	\	\	\	0.14	0.30	0.20	0.10	5
6	\	\	\	\	\	\	0.27	0.55	0.37	0.41	6
7	\	\	\	\	\	\	21.32	53.40	44.50	44.51	7
8	\	\	\	\	\	\	27.95	7.04	46.44	39.99	8
9	\	\	\	\	\	\	90.43	25.28	140.60	125.60	9
10	\	\	\	\	\	\	44.49	2.04	59.76	57.88	10
11	178.45	197.85	339.98	317.75	353.89	206.52	67.90	5.12	51.09	85.01	11
12	226.20	242.01	449.29	428.29	461.46	254.39	92.12	7.43	56.35	118.33	12
13	776.01	961.53	1441.05	1324.54	1624.60	878.32	321.26	74.62	474.11	187.44	13
14	1215.79	1428.83	2505.53	1964.86	2419.25	1325.28	475.79	8.06	542.53	576.93	14

DRM: data-compression-based resource management in cloud computing; AHC: agglomerative hierarchical clustering, KC: kernel clustering.
Here \ denotes the corresponding method cannot compress the related data set and we have no related experimental results. DRM denotes the method we proposed in this paper, DRM (*K*-means) denotes the DRM adopts *K*-means as the clustering method, DRM (AHC) denotes the DRM adopts AHC as the clustering method, and DRM (KC) denotes the DRM adopts KC as the clustering method.

Table 7. Time (hour) for getting large data sets

Data order	Original data	BioCompress	BioCompress-2	GenCompress	DNAPack	DNADP	GeNML	DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)
1	1.02	0.57	0.59	0.61	0.68	0.43	0.41	0.13	0.04	0.17	0.15
2	0.55	0.45	0.54	0.29	0.26	0.24	0.21	0.11	0.04	0.17	0.16
3	0.68	0.43	0.51	0.38	0.34	0.31	0.29	0.14	0.01	0.11	0.13
4	0.31	\	\	\	\	\	\	0.04	0.06	0.04	0.03
5	0.29	\	\	\	\	\	\	0.05	0.11	0.07	0.04
6	0.39	\	\	\	\	\	\	0.04	0.08	0.05	0.06
7	113.21	\	\	\	\	\	\	9.03	22.62	18.85	18.85
8	21.51	\	\	\	\	\	\	2.74	0.84	3.59	3.16
9	69.60	\	\	\	\	\	\	13.92	5.06	21.51	20.25
10	34.24	\	\	\	\	\	\	7.05	0.50	5.54	6.55
11	52.26	29.42	28.15	32.28	34.69	21.45	20.64	6.33	2.06	8.34	7.70
12	70.89	38.22	39.15	42.28	51.70	30.54	28.28	9.53	2.78	12.07	10.14
13	247.25	132.85	145.57	151.37	168.04	98.74	100.02	33.60	9.90	42.55	36.93
14	366.17	197.77	200.85	230.36	241.61	157.71	148.10	45.12	14.53	61.55	54.42

DRM: data-compression-based resource management in cloud computing; AHC: agglomerative hierarchical clustering, KC: kernel clustering.
Here \ denotes the corresponding method cannot compress the related data set and thus we need not have time for getting data when the corresponding method is used. Because the time should be same as the one in the second column. Namely, for the data sets which cannot be compressed by BioCompress, BioCompress-2, GenCompress, DNAPack, DNADP, and GeNML, whether to adopt these compression technologies, the time for getting data is same. DRM denotes the method we proposed in this paper, DRM (*K*-means) denotes the DRM adopts *K*-means as the clustering method, DRM (AHC) denotes the DRM adopts AHC as the clustering method, and DRM (KC) denotes the DRM adopts KC as the clustering method.

Table 8. Time (hour) for reconstructing large data sets

Data order	DRM	DRM (*K*-means)	DRM (AHC)	DRM (KC)
1	0.12	0.04	0.15	0.14
2	0.09	0.03	0.11	0.10
3	0.13	0.01	0.11	0.14
4	0.03	0.06	0.05	0.03
5	0.02	0.06	0.04	0.02
6	0.06	0.11	0.06	0.08
7	16.32	30.93	24.77	22.61
8	20.48	4.24	26.20	23.71
9	69.72	15.63	82.50	68.55
10	53.23	1.92	54.69	51.76
11	59.83	3.57	34.46	61.34
12	59.80	3.49	26.62	59.35
13	233.11	41.42	267.87	108.38
14	370.89	5.14	91.61	372.43

DRM: data-compression-based resource management in cloud computing; AHC: agglomerative hierarchical clustering, KC: kernel clustering.
DRM denotes the method we proposed in this paper, DRM (*K*-means) denotes the DRM adopts *K*-means as the clustering method, DRM (AHC) denotes the DRM adopts AHC as the clustering method, and DRM (KC) denotes the DRM adopts KC as the clustering method.

V. CONCLUSION AND FUTURE WORK

With the application and development of new biomedical techniques, we face the data problem of large data sets. Cloud computing represents excellent advantages regarding resource allocation, data storage, computation, and sharing in order to solve problems of big data sets from biomedical research. If compression technologies are not adopted, the capacity crisis will be encountered in the future. Since the traditional compression methods including BioCompress, BioCompress-2, GenCompress, DNAPack, DNADP, and GeNML possess some disadvantages, for example, they can only be used in a few fields, this paper proposes a clustering method and adopts RBF to compress large biomedical data sets with high quality, and then stores these data with resource management in cloud computing. This proposed method is called data-compression-based resource management (DRM) in cloud computing. Experiments have validated that DRM can help to store large data sets in fewer capacities. Furthermore, with reverse operation of RBF, these compressed data can be reconstructed with high accuracy and result in shorter times for storing and recalling.

Although the proposed method can save capacities, time, and reconstructing data with a high authenticity, there is another question that should be solved in the future. In order to compress large data sets, they must first be quan-

tized. Yet, some data sets cannot be quantized; for example, some data sets have lost information or are locked by special departments and cannot be changed. Regarding these data sets, our future work is aimed at finding new compression methods.

REFERENCES

1. J. Qin, Y. Li, Z. Cai, S. Li, J. Zhu, F. Zhang, Y. Peng, S. Liang, W. Zhang, Y. Guan, et al., "A metagenome-wide association study of gut microbiota in type 2 diabetes," *Nature*, vol. 490, no. 7418, pp. 55-60, 2012.
2. M. C. Schatz, B. Langmead, and S. L. Salzberg, "Cloud computing and the DNA data race," *Nature Biotechnology*, vol. 28, no. 7, pp. 691-693, 2010.
3. "Gathering clouds and a sequencing storm: why cloud computing could broaden community access to next-generation sequencing," *Nature Biotechnology*, vol. 28, no. 1, 2010. http://dx.doi.org/10.1038/nbt0110-1.
4. A. Rosenthal, P. Mork, M. H. Li, J. Stanford, D. Koester, and P. Reynolds, "Cloud computing: a new business paradigm for biomedical information sharing," *Journal of Biomedical Informatics*, vol. 43, no. 2, pp. 342-353, 2010.
5. E. Pennisi, "Human genome 10th anniversary. Will computers crash genomics?," *Science*, vol. 11, no. 6018, pp. 666-668, 2011.
6. A. Darling, L. Carey, and W. C. Feng, "The design, implementation, and evaluation of mpiBLAST," in *Proceedings of ClusterWorld Conference & Expo*, San Jose, CA, 2003.
7. E. E. Schadt, M. D. Linderman, J. Sorenson, L. Lee, and G. P. Nolan, "Computational solutions to large-scale data management and analysis," *Nature Reviews Genetics*, vol. 11, no. 9, pp. 647-657, 2010.
8. D. P. Wall, P. Kudtarkar, V. A. Fusaro, R. Pivovarov, P. Patil, and P. J. Tonellato, "Cloud computing for comparative genomics," *BMC Bioinformatics*, vol. 11, pp. 1-12, 2010.
9. L. D. Stein, "The case for cloud computing in genome informatics," *Genome Biology*, vol. 11, pp. 1-7, 2010.
10. J. T. Dudley, Y. Pouliot, R. Chen, A. A. Morgan, and A. J. Butte, "Translational bioinformatics in the cloud: an affordable alternative," *Genome Medicine*, vol. 2, pp. 1-6, 2010.
11. J. Wilkening, A. Wilke, N. Desai, and F. Meyer, "Using clouds for metagenomics: a case study," in *Proceedings of IEEE International Conference on Cluster Computing & Workshops*, New Orleans, LA, 2009, pp. 1-6.
12. National Institute of Standards and Technology, "The NIST definition of cloud computing," Sep. 2011; http://dx.doi.org/10.6028/NIST.SP.800-145.
13. S. Grumbach and F. Tahi, "Compression of DNA sequences," in *Proceedings of Data Compression Conference (DCC'93)*, Snowbird, UT, 1993, pp. 340-350.
14. S. Grumbach and F. Tahi, "A new challenge for compression algorithms: genetic sequences," *Information Processing & Management*, vol. 30, no. 6, pp. 875-886, 1994.
15. X. Chen, S. Kwong, and M. Li, "A compression algorithm for DNA sequences and its applications in genome comparison," *Genome Informatics*, vol. 10, pp. 51-61, 1999.
16. T. Matsumoto, K. Sadakane, and H. Imai, "Biological sequence compression algorithms," *Genome Informatics*, vol. 11, pp. 43-52, 2000.
17. B. Behzadi and F. L. Fessant, "DNA compression challenge revisited: a dynamic programming approach," in *Combinatorial Pattern Matching*, Heidelberg: Springer, pp. 190-200, 2005.
18. K. G. Srinivasa, M. Jagadish, K. R. Venugopal, and L. M. Patnaik, "Efficient compression of nonrepetitive DNA sequences using dynamic programming," in *Proceedings of International Conference on Advanced Computing & Communications*, Surathkal, India, 2006, pp. 569-574.
19. G. Korodi and I. Tabus, "An efficient normalized maximum likelihood algorithm for DNA sequence compression," *ACM Transactions on Information Systems*, vol. 23, no. 1, pp. 3-34, 2005.
20. W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of Classification*, vol. 1, no. 1, pp. 7-24, 1984.
21. J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a k-means clustering algorithm," *Journal of the Royal Statistical Society Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100-108, 1979.
22. D. Gao and J. Li, "Kernel fisher discriminants and kernel nearest neighbor classifiers: a comparative study for large-scale learning problems," in *Proceedings of International Joint Conference on Neural Networks*, Vancouver, BC, 2006, pp. 1333-1338.

Changming Zhu

Changming Zhu received the B.Sc. degree in Department of Computer Science and Engineering, East China University of Science and Technology, China, 2010 and the Ph.D. degree in Department of Computer Science and Engineering, East China University of Science and Technology, China, 2015. Now he is a teacher in College of Information Engineering, Shanghai Maritime University, Shanghai, China. His research interests focus on neural computing and pattern recognition.