Elsevier Editorial System(tm) for Applied Soft Computing

Manuscript Draft

Manuscript Number: ASOC-D-16-01038

Title: Parallel Swarm Intelligence Strategies for Large-scale Clustering based on MapReduce with Application to Epigenetics of Aging

Article Type: SI: EMO-BD

Keywords: Big data\sep Swarm intelligence \sep Large-scale clustering\sep Map-Reduce \sep Epigenetics \sep Aging

Corresponding Author: Mr. Zakaria Benmounah, Ph.D student

Corresponding Author's Institution: Constantine 2 university

First Author: Zakaria Benmounah, Ph.D student

Order of Authors: Zakaria Benmounah, Ph.D student; Souham Meshoul, Prof; Mohamed Batouche, Prof; Pietro Lio', Dr

Abstract: The high-throughput sequencing technologies have produced a wealth of epigenetics data. These datasets require stand-alone techniques to extract useful insights which can be used for further analysis. One tailored technique is data clustering; it is a primary method to extract the first layer of information from unlabeled data sets. However, epigenetics data sets are very large making conventional data clustering techniques inappropriate. By another way, Swarm Intelligence (SI) algorithms such as Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO) have shown promising results when applied to data of moderate size. They exhibit different capabilities making their cooperation a promising alternative to achieve good quality clustering. In this paper, a parallel and distributed generalized island model (GIM) based on these SI algorithms is developed according to MapReduce framework. The proposed framework (MRC-GIM) allows cooperation between the three SI algorithms to achieve largely scalable data partitioning. MRC-GIM has been validated on Amazon Elastic MapReduce service (EMR) deploying up to 192 computer nodes and 30 gigabytes of data. The experiments show that MRC-GIM competes and often outperforms existing methods. The developed model has been applied to study the epigenetics impact on aging; experimental results reveal that DNA-methylation changes slightly with aging, confirming previous studies.

***Highlights (for review)**

This piece of the submission is being sent via mail.

***Graphical abstract (for review)**

This piece of the submission is being sent via mail.

LaTeX Source Files

Click here to download LaTeX Source Files: manuscript.rar

Parallel Swarm Intelligence Strategies for Large-scale Clustering based on MapReduce with Application to Epigenetics of Aging

Zakaria Benmounah[1,2], Souham Meshoul[1], Mohamed Batouche[1], Pietro Lio'[3]

Abstract

The high-throughput sequencing technologies have produced a wealth of epigenetics data. These datasets require stand-alone techniques to extract useful insights which can be used for further analysis. One tailored technique is data clustering; it is a primary method to extract the first layer of information from unlabeled data sets. However, epigenetics data sets are very large making conventional data clustering techniques inappropriate. By another way, Swarm Intelligence (SI) algorithms such as Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO) have shown promising results when applied to data of moderate size. They exhibit different capabilities making their cooperation a promising alternative to achieve good quality clustering. In this paper, a parallel and distributed generalized island model (GIM) based on these SI algorithms is developed according to MapReduce framework. The proposed framework (MRC-GIM) allows cooperation between the three SI algorithms to achieve largely scalable data partitioning. MRC-GIM has been validated on Amazon Elastic MapReduce service (EMR) deploying up to 192 computer nodes and 30 gigabytes of data. The experiments show that MRC-GIM competes and often outperforms existing methods. The developed model has been applied to study the epigenetics impact on aging; experimen-

[☆]Fully documented templates are available in the elsarticle package on CTAN.

[1]Computer Science Department, Constantine 2 University, Constantine, Algeria.
[2]MISC laboratory.
[3]Computer Science Department, Cambridge University, Cambridge, UK.

tal results reveal that DNA-methylation changes slightly with aging, confirming previous studies.

Keywords: Big data, Swarm intelligence, Large-scale clustering, Map-Reduce, Epigenetics, Aging

2010 MSC: 00-01, 99-00

1. Introduction

Over the last few decades, developed countries in Europe, Asia, and North America have experienced a significant change in the age profile of their demographic structure, with a steady increase in the number of adults aged 65 years or older. Although longer lifespan is a clear sign of progress and improved quality of life, the considerable growth of the elderly population poses far-reaching social and economic challenges regarding the fiscal sustainability of the welfare state. As such, health-care system and related services have to be rethought to treat aging as a manipulable long-term biological process whose detrimental effects can be limited or procrastinated rather than passively accepted as inevitable.

Key research challenges need to be effectively and efficiently addressed to cope with the ever-growing amount of epigenetics data that is being exponentially produced. Traditional techniques and tools for data analytics and autonomous learning are no longer suitable and even unusable to extract human-interpretable knowledge and information from the enormous complex amount of data. Therefore, new revolutionary approaches and tools are more than required, among these tools we highlight clustering.

Almost 60 years beyond have passed from the first proposed clustering algorithm [1]. Cluster analysis aims at grouping data points into separate groups called clusters. It plays a versatile role in knowledge discovery, and it is used in a myriad of fields to extract hidden relationships among data. An up-to-date review of the application of clustering analysis can be found in [2]. A plethora of clustering algorithms has been designed to deal with different types and dis-

tributions of data. The exponential increase of data makes cluster analysis even more challenging than before. Two broad approaches have emerged to alleviate the issue, either by reducing the dimensionality of data (dimensionality reduction)[3] or by reducing the number of samples within a dataset (sampling) [4]. However both approaches have been proved to be ineffective when a single machine is used as the prohibitively large amount of data cannot be kept on a single computer. Therefore, multiple machines are needed, and parallel processing of data is undeniably necessary.

Hardware accelerators such as Field Programmable Gate Array (FPGA) and Graphics Processing Unit (GPU) have emerged recently as promising technology drivers [5]. On the other hand, Application Programming Interfaces (APIs) such as Message Passing Interface (MPI) and OpenMP have traditionally provided a software-oriented approach. However while dealing with parallel programming languages, additional concerns have to be considered, such as the load balancing, the communication flow, the topology choice, the split of data, etc. This makes designing a parallel algorithm a very tedious task. To deal with these concerns, a new open source framework called Apache Hadoop consisting of a storage part namely Hadoop Distributed File System (HDFS) and a processing part namely MapReduce (MR) has emerged lately. HDFS is a distributed file system that provides high-performance access to data across Hadoop clusters by managing pools of big data and handling big data analytics applications. MapReduce, designed by Google [6], provides a new methodology of thinking and developing a parallel algorithm suitable for large scale systems without being concerned about scalability as MapReduce is auto-scalable. The idea was inspired by the map and reduce primitives characterizing the functional programming LISP. Hadoop encapsulates the details of parallelization, fault-tolerance, data distribution and load balancing. It Demonstrates a great performance in big data scenarios, especially for challenging tasks such as clustering.

Nevertheless, data Clustering is an NP-hard problem. If the number of clusters exceeds three, the alternative ways to group the data is $\frac{k^n}{k!}$, where k is the number of groups, and n is the number of data points to be clustered.

However, data clustering can be easily cast as a global optimization problem of finding the partition that maximizes/minimizes an objective function with only partial search space than exhaustive one. This can be appropriately tackled using metaheuristics, such as Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO) and so forth. These metaheuristics exhibit different dynamics leading to distinct strategies that can be effectively combined to handle hard optimization problems such as Clustering large datasets.

In this paper, first, we present and discuss a review of big data clustering algorithms based upon MapReduce. Furthermore, we propose a MapReduce design of the Ant Colony Optimization and an Artificial bee colony to cluster large data sets. Additionally, we merge the proposed algorithms in a collaborative way based on Generalized island model (MRC-GIM). MRC-GIM enables the cooperation between three SI metaheuristics to explore the search space more efficiently. We validate the proposed algorithms on many computers (192 computers) connected with each other and large real datasets (more than 30GB). The comparative study reveals that MRC-GIM outperforms novel developed clustering algorithms dedicated to big data clustering. Subsequently, we use MRC-GIM to investigate the correlation between Epigenetics and Aging. As a result of this application, we found that epigenetics changes slightly and not aberrantly with aging, this latter confirms previous evidence shown in [59, 60].

The remainder of the paper is organized as follows: in Sec. 2 a brief description of the background material is given. In Sec, 3. a review of clustering large data sets algorithms is provided. Section 4 is devoted to the presentation of the proposed frameworks. In Sec. 5, the performance of MRC-GIM is evaluated using large datasets and parallel metrics. In Sec. 6 we study the correlation between epigenetics and aging and finally in Sec. 7 conclusions are drawn.

2. Background and materials

2.1. Clustering Analysis

Clustering analysis is a central topic in computer science and statistics, whose purpose is to find groups in a given data set, according to a notion of similarity or homogeneity that is maximized/minimized for elements in the same group (or cluster). Formally, given a set V, a clustering is a partition $\tau(V) = \{c_1, \ldots, c_K\}$ of V into non-empty and pairwise disjoint subsets c_i, $i = 1, \ldots, K$, whose union is V.

As a major field of research in exploratory data analysis, over the years, a large number of clustering algorithms have been proposed to find "good" partitioning that can uncover latent structures in the data. Clustering approaches can be divided into two major classes: partitional clustering and hierarchical clustering.

Partitional clustering algorithms split the dataset into groups based on a two-step iterative process. Given an initial set of cluster representatives centroid locations as in k-means [7] or centroid data points as in k-medoids [8] the procedure alternates an assignment step where each data point is assigned to the cluster with the closest representative and an update step where cluster representatives are recomputed.

Hierarchical clustering algorithms build a hierarchy of clusters by either merging smaller clusters into larger clusters (agglomerative clustering) or by splitting larger clusters into smaller clusters (divisive clustering). As a result, a hierarchical clustering algorithm produces a tree of clusters, called dendrogram, which shows how clusters and data points are related.

2.2. MapReduce Model

A MapReduce program takes place generally in three successive jobs, the *Map*, the *Auxiliary* and the *Reduce* jobs as depicted in figure 1.

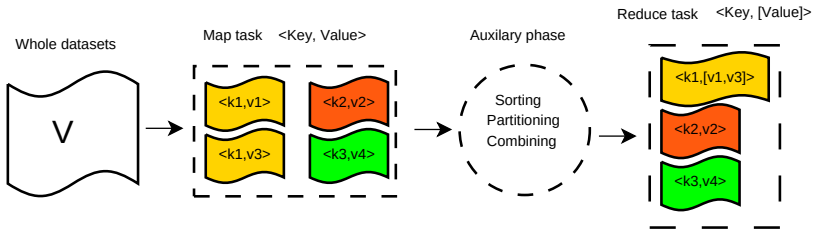


Figure 1: MapReduce framework, generally composed of the map job followed by the reducer job

The map job is a preprocessing of the split data where each split part is considered as a *value* and attributed a *key*, all values with the same key are submitted to the same reducer. More formally let's denote V as the whole data, after splitting the data we get $V = v_1 \cup v_2 \cup \ldots \cup v_p$ where p is the number of split elements. The mapper creates a function $V \rightarrow key$ where $key = key_1, key_2, \ldots key_r$ and r denotes the number of keys ($r << p$). A key key_i can be attributed to more than one value. The *Auxiliary* is an optional function that takes place between the mappers and the reducers when some additional preprocessing tasks are required. The reducer receives a set of pairs in the form of $< key, value >$ the values with the same key are grouped together in the same reducer thus it results in one key per each reducer.

3. Related works

Since the inception of MapReduce (MR), several MR based clustering algorithms have been proposed. Most of these methods are based on different MR schemes of k-means. Among them, Pk-means [12] which follows the classical k-means procedure and runs in one iterative MR job. In the map phase, centroids are computed as the weighted average of all points within a cluster, then in the reducer phase, the algorithm updates the new centers. An additional auxiliary phase is set up to combine the intermediate data of the same map task. Further, k-means++ has been designed to optimize the selection of initial centers, starting by choosing them uniformly at random, then adding potential

centers one by one in a controlled way until reaching k centers. k-means‖ [18] has come as an improvement of k-means++, rather of sampling a single point in each pass like kmeans++, k-means‖ samples O(k) points in each round and repeats the process for approximately O(log n) rounds to get better accuracy. Unlike the aforementioned proposed k-means based methods which perform a series of k-means tasks serially, Mux-Kmeans [24] performs a multiple k-means tasks concurrently by taking various centroids together. The algorithm starts by evaluating the clusters, selecting the best task and finally integrating the new tasks. Efficient k-means++ [25] is another improvement of k-means++ which uses only one MR job to obtain the k centers. The map task consists in the k-means++ initialization, followed by the weighted k-means++ which is performed in the reducer phase. The algorithm integrates a pruning strategy that computes the distances without redundant time as advantage selection of k (the number of the clusters) is automatic.

By using k-means with sampling, X. Cui et al. [27] proposed an algorithm that takes place in three MR jobs. Data summarization has been used as a baseline for BigK-Clustering [21]. The key idea behind the algorithm is to divide the data into sub-data and group them separately which results in micro-clusters, then merge the closest micro-clusters using the equivalence relation, and finally, calculate the centers of the final groups. DBCURE-MR [26] and MR-DBSCAN [15] are two density based algorithms that identify clusters with highly dense areas separated by sparsely dense areas. Both methods employed a sampling approach to reducing the size of the large data and carry out the processing of the reduced data to obtain centers, which will be used to cluster the original data. Abhinandan Das et al. redesign the MinHach algorithm [10] in a more scalable way using map-reduce. This algorithm can capture multiple interests of user shared within a cluster. DisCo [11] is based on co-clustering which unlike clustering attempts to cluster both samples and items at once.

Table 1: Summary of clustering algorithms based on MapReduce (sz: data size, nd: number of computer node, pt: data point, dim: dimension and syn: Synthetic)

Algorithm	Based on	Data	platform
Abhinandan Das et al. [10]	Probabilistics based (MinHash)	Real (943, 5000, 500000) users	non
Spiros Papadimitriou et al. (DisCo) [11]	Co-Clustering	Raw dataset, Sz(100-135GB) Graph dataset, sz(170MB-4.3GB)	39 nd with 2 IntelXeon (2.7-3GHz), 8GB of ram.
Weizhong Zhao et al. (PKMeans) [12]	K-means	sz (1GB- 8GB).	(1-4) -nd with 2 cores (2.8GHz), 4GB of ram.
Robson L. F. Cordeiro et al. (BOW) [13]	Subspace Clustering	-Real:(62millions-1.4 billion)pt, sz: (0.014-0.2)TB. -syn (100000-100 million) pt*15 dim	-Yahoo Cluster M45: 400 nd, 3.5 TB ram. -DISC:(64 machines=512 cores +1TB of ram)
Hai-Guang Li et al. [14]	Bagging, Kmeans	Small datasets	-1 nd IntelCore Duo, (2.10GHZ), 2.00GB ram.
Yaobin He et al.(MR-DBSCAN) [15]	Density based	Real(GPS location records): 0.32 billion-1.92 billion pt, sz((8.4GB-50.4GB).	-13 nd Intel Core i7 950 (3.0GHz), 8GB DRAM
Alina Ene et al. (Parallel Lloyds) [16]	K-center, K-median	syn (Zipf distribution): 10,000-10,000,000 pt.	-1 nd IntelCore i7 (2.93GHz) with, 8GB ram (100nd simulation).
Ibrahim Aljarah et al.(MR-CPSO) [17]	Particle swarm optimization	Real and syn: (2, 000-32, 000, 000)pt (2-54) dim, sz(0.14- 1320.8)MB	-Longhorn Hadoop cluster: 48 nd, 8 Intel Nehalem cores (2.5GHz), 48GB ram. -NDSU Hadoop cluster: 18 nd 4 Intel cores (2.67GHz each), 6GB of ram.
Bahman Bahmani et al.(k-means ‖) [18]	K-means	-syn: GaussMixture 10;000 points -real(4601-4.8)M pt*(42-58) D.	1968 nd, 2 quad-core (2.5GHz), 16GB ram.
Fei Gao et al. [19]	Approximate Spectral Clustering	-syn: 1024 to 4 million pt*64 dim -Real (wikipedia): sz (1M-2G)	Local Cluster: 5 Core2 Duo E6550, (2.33 GHz) with 1 GB DRAM.
Trilce Estrada et al. [20]	Tree-based	real (molecular geometries), sz (48MB-1TBytes).	16 nd Gordon ION (192 cores), (2.7 GHz) with 3GB of ram.
Yuqing Miao et al.(BigKClustering) [21]	K-means, micro-cluster structure	syn:(normal dist), Sz(1GB-3GB)	3 nd with 2 IntelCore (3.1GHz), 4GB of ram.
Chen Jin et al. (DiSC) [22]	Single-Linkage, MST (Hierarchical)	syn: 500,000 data pt * 10 dim.	JESUP hadoop cluster
Pelle Jakovits et al. (CLARA) [23]	K-medoid	real(handwritten digits), 25 000- 10 000 000 pt.	17 nd with CPU (2.2GHz), 500MB RAM.
Chen Li et al. (Mux-Kmeans) [24]	K-means	real (17770-359330) pt* (40-1000) dim.	EC2: 16nd, 2 ECUs and 1 CPU, 3.7 GiB ram
Yujie Xu et al. (Efficient K-means++) [25]	K-means	-real sz (5.67 GB). -Syn: 20 million pt * 128 dim, sz(15GB).	12 nd with 2 AMD Opteron 2212, (2.00GHz) and 8GB of ram.
Younghoon Kim et al. (DBCURE-MR) [26]	Density-based	-Real: (164,860-164,860*50)pt* 3 dim sz(21mb-1.05GB). -Syn:(5,000,000-25,000,000)pt* (2 to 8 dim), sz(1.7GB).	20 nd with 2 Duo (2.66GHz), 2GB of ram.
Xiaoli Cui et al. [27]	K-means and Sampling	-syn(Gauss Distribution) 10,000 pt* 3 dim. -Real 2,351,710,420 pt* 3 dim -Real 4,296,075,259 pt * 9 dim.	16 nd 2 Core AMD, Opteron (2.00GHz), 2GB of ram.

Fei Gao et al. designed an approximate spectral clustering which enables kernel-based machine learning algorithms to efficiently process very large-scale

datasets [19] with the objective to optimize the computation and memory overhead required to compute the kernel matrix without affecting the accuracy of the result. The approach is independent of the employed kernel-based machine learning and was tested on Amazon Elastic MapReduce. Trilce Estrada et al. [20] presented a scalable method dedicated to the molecular conformation such as ligand and peptides to identify conformations that are an accurate representative of the native conformations. The algorithm starts by reducing the conformation space and then combines the geometrical knowledge of conformation similarities based on a tree clustering, a considerable decrease of parallel runtime has been reported, from 5 hours to 20 minutes while tweaking the processing nodes from 12 to 196. Hierarchical clustering based has also witnessed the scaling up using MapReduce. As an example, DiSC is a hierarchical clustering algorithm [22] designed according to MapReduce framework which runs on a fairly small number of MR rounds, while reducing the hierarchical clustering single-linkage problem to the minimum spanning tree (MST). The CLARA is a medoid-based clustering algorithm [23] which unlike centroid-based chooses real data points as centers. The algorithm has been defined regarding MapReduce jobs, showing a way of how to adapt a non-embarrassingly parallel algorithm to a platform that is dedicated to embarrassingly parallel methods. In contrast, the efficiency and scalability have been barely affected. Finally, MR-CPSO [17] is based on particle swarm optimization (PSO) algorithm originally developed by taking inspiration from the flocking behavior of birds. The algorithm is implemented in three MR stages each of which consists in a PSO operation. In the first stage the centroids of individual particle are updated using the classical equation of PSO that updates the velocity and the position, afterwards, an evaluation of the fitness generated in the first level is computed, then, the last stage is devoted for merging results of the previous stages and identify the personal best and global best centroids. In table 1, a summary of all these algorithms is provided including the basic clustering features used as described above with extra information related to the type and volume of data sets and the platform employed per each algorithm.

From this brief review, one can notice that attempts to develop MapReduce methods of clustering are at their debut and in particular, swarm-based algorithms which still need to be rethought in this context despite their success to solve NP-hard problems and particularly data clustering. By another side, cooperating among metaheuristics has been shown to be a very promising way to solve efficiently complex problems. Therefore, in our work we investigate the potential of such cooperation to handle clustering of large data sets.

4. Highly scalable clustering based on MapReduce

A generalized island model is a parallel distributed scheme which is amenable to large-scale clustering. To perform clustering of large data sets using a GIM we need first to redesign each of the used metaheuristics according to the MapReduce model. In the proposed GIM, the three swarm-based algorithms PSO, ACO and ABC cooperate to find the best partition that optimizes a given cluster quality measure. In our study, we consider the total within variance or cohesion measure as the objective function. This later describes how close the data points within each cluster are to each other. It is defined as follows:

$$\frac{1}{n}\sum_{j=1}^{k}\sum_{x_i \in z_j} \|x_i - z_j\|^2 \tag{1}$$

Where n is the number of items or data points in the datasets, k the number of clusters, $x_i, \{i = 1, \ldots, n\}$ the location of the ith item to be clustered and z_i the center of cluster c_i. Therefore, clustering is cast as an optimization task that aims to minimize the cohesion total within variance.

Our algorithms can run even if:

1. The data is huge and cannot be stored on a single computer,
2. The clustering solution could be distributed among different computers,
3. The clustering solution is constructed gradually.

In the rest of this section, we consider the following notations:

1. $v = \{item_1, item_2, ..., item_n\}$, the set of n items or data points to be grouped into clusters,
2. m: The number of artificial ants or bees.
3. k: The number of clusters.
4. $p_i = \{(item_j, c_l)\}$, where $\{i = 1, \ldots, m\}$, $\{j = 1, \ldots, n\}$ and $\{l = 1, \ldots, k\}$, the clustering partition is encoded as a set of pairs, each is composed of an item and the assigned cluster.
5. f_i is the cohesion value of the clustering solution p_i.
6. $c = \{z_1, z_2, \ldots, z_k\}$,the set of centroids.

In the following, we describe the proposed ACO and ABC based MapReduce. Afterward, we describe the overall architecture through which cooperation between ACO, ABC and PSO introduced in [17] is done.

4.1. Ant colony optimization based MapReduce

ACO is a swarm based metaheuristics developed by Dorigo et al. [28] to solve NP-hard optimization problems. It is inspired by the foraging behavior of a colony of ants. A detailed description of this metaheuristic and its biological metaphor can be found in [29]. Given an optimization problem to be solved, a typical dynamics of an ACO algorithm can be defined as follows. A set of cooperating artificial ants is used to construct solutions incrementally. Each ant constructs a solution and deposits an amount of pheromone on each solution component according to the overall quality of the solution found. During solution construction step, an ant selects the next solution component according to a probabilistic rule which depends on the corresponding amount of pheromone and the value of a problem-specific heuristic. Many variants of ACO algorithms exist depending on the policies used to update pheromone trails and to select solutions components.

In our work, we adopted ACO algorithm for data clustering inspired from ant k-means described in [30]. The algorithm aims to find the partitions that minimize the cohesion measure defined above. A solution component is viewed as an assignment of a data point to a cluster. Let's adopt the following notations:

1. τ, the threshold value.
2. $\phi(i,j)$, where $\{i = 1,\ldots,n\}$, $\{j = 1,\ldots,k\}$, is n by k matrix represents the pheromone value of each item in each cluster.

The proposed ACO algorithm MapReduce (MRC-ACO) composed of three stages, both the first and second are parallel and based upon MapReduce; the last stage is sequential as shown in figure 2.

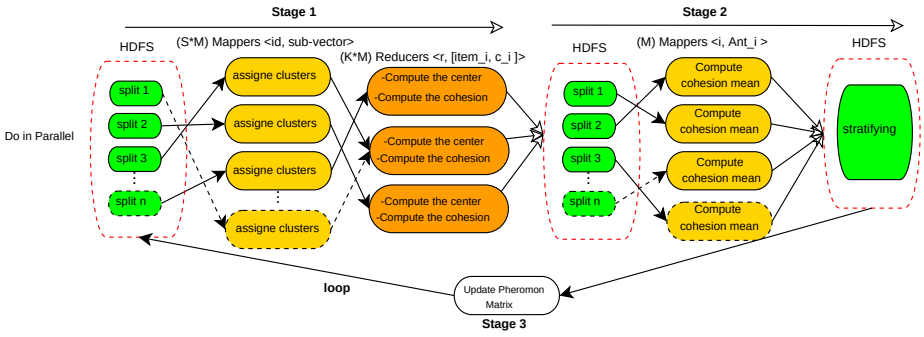


Figure 2: MRC-ACO: Ant Colony Optimization for clustering large datasets based on MapReduce framework

The ACO algorithm for data clustering can be outlined as follows:

1. During an initialization phase, each ant builds a solution by assigning items to clusters randomly. The same amount of pheromone is assigned to each of all possible pairs (item, cluster).
2. During an iterative phase each ant updates the amount of pheromone to each of its solution components namely a pair $(item, cluster)$ and constructs another solution by assigning items to clusters according to a probability that depends on the amount of pheromone and heuristic value related to the cohesion measure value. During iterations, the best partition i.e. with the smallest value of cohesion measure found so far is kept.
3. At the end of the iterative process, the best partition is returned.

To handle the big data, we split each p_i vector to s sub-vectors where s is an integer tweaked with regards to the size of data points and the memory

capacity available. The sub-vectors are loaded on the mappers, at this end, the mappers assign in parallel for each $item_i$ the cluster number c_i following a guided probabilistic research described in algorithm 1. Afterward, the mappers attribute the same key to all sub-vector pairs $(item_i, c_i)$ that satisfy the following two conditions:

1. The pairs should belong to the same p_i parent vector.
2. Second the c_i of each pair has been assigned to the same cluster.

This latter is done through the use of MD5 [34] function which is a cryptographic hash function that provides the digital fingerprint. Subsequently, the mappers transmit the pairs with their keys to the reducers, where the same key is transmitted to the same reducer. Increasing s leads to the decrease of the granularity (the size of the parallel task) per each mapper, giving rise to a shorter processing time. The number of the parallel task at this stage is m multiplied by s.

At the reducers job the transmitted pairs merged to form one cluster of a p_i vector which is the purpose of applying both above conditions. In parallel, each reducer computes the center and evaluates the clustering quality by computing the cohesion as described in equation 1.

The granularity of the system increases compared to the first stage as the number of parallel tasks equals to k multiplied by m. The purpose of splitting the data per cluster is to contribute to the decrease of the granularity that yields to more tasks in parallel thus a rapid system.

In the second stage, another MapReduce round is launched to evaluate the quality of each p_i vector and to update the pheromone matrix. Firstly, a light map stage receives per each mapper the cohesion values of a p_i vector and computes for them the mean. The mean describes how good the overall clustering solution is. In this round, the number of tasks in parallel is m.

At this end, all p_i possess a complete clustering solution with its quality calculated. An auxiliary phase is employed to stratify in an ascending way the p_i vectors with regards to their cohesion values preparing them for the

pheromone update stage.

Algorithm 1: Do in parallel for each $mapper_i$

Input: $< key : id, value :$ sub-vector$>$

extract ϕ; **for** *each* $item_i$ *in sub-vector* **do**

generate random number r ;

if $r \leq q$ **then**

$pMax = max(\phi(item_i, \forall))$; `// the highest pheromone value`

$c_i = find(\phi(item_i, \forall), pMax)$; `// where the highest pheromone laid`

else

$pSom = \sum_{h=1}^{h=k} \phi(item_i, h)$;`// the sum of the pheromone recorded`

for *each* $c_l > 1$ *in* k **do**

$\phi(item_i, c_l) = \frac{(\phi(item_i, c_(l-1)) + \phi(item_i, c_l))}{pSom}$; `// the shared pheromone`

end

generate random number rr ;

for *each* c_l *in* k **do**

if $rr \leq \phi(item_i, c_l)$ **then**

$c_i = c_l$;

end

end

end

end

for *each pair* $(item_i, c_i)$ *in sub-vector* **do**

generate a key_i;

transmit $(item_i, c_i)$; `// via the use of MD5`

end

The last stage is sequential to update the pheromone matrix. For each corresponding value of the assigned cluster, c_i in the best solution found so far; the pheromone value is updated ϕ as follows:

$$\phi(i,j) = \frac{(1-\rho)\phi(i,j)+1}{f_{best}} \tag{2}$$

Where ρ is the evaporation rate and f_{best} is the best cohesion value found so far.

4.2. Artificial bee colony clustering based MapReduce

We designed an artificial bee colony clustering based on MapReduce (MRC-ABC), MRC-ABC is composed of 4 stages described in figure 3.

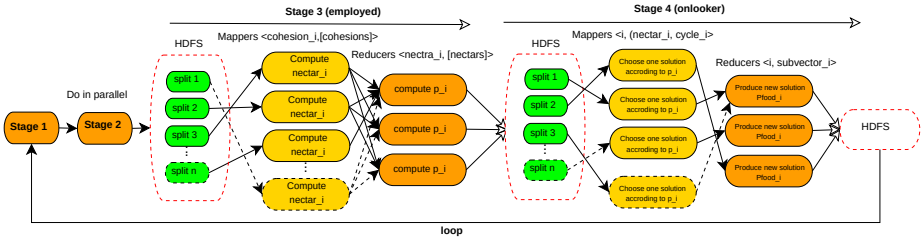


Figure 3: MRC-ABC: an Artificial Bee Colony Clustering for large datasets based on MapReduce Framework

The following assumptions are considered [31]:

1. Three groups of bees: employed bees, onlookers, and scouts artificial bees. The employed bees produce a food source (solutions) and share it with the onlookers in the dance area. The onlookers choose one food source with a probability related to its nectar amount and produce for it the distinct new solutions. The job of the scout is to check whether a solution is not chosen by the onlookers for many cycles if so the abandoned solution is replaced with a newly generated one.
2. $nectar_i$ refers to the nectar amount of the food source, it evaluates how much a solution is good and calculated as follows:

$$nectar_i = \frac{cohesion(Food_i)}{\sum_{s=1}^{M} cohesion(f_s)} \tag{3}$$

3. pb_i refers to the probability of a p_i solution to be chosen by the onlookers, it is computed as follows.

$$pb_i = \frac{nectar_i}{\sum_{s=1}^{M} nectar_s} \tag{4}$$

4. To avoid getting stuck in the local minimum, ABC employs the variable $cycle_i$ for each p_i, that will be incremented when a food source is not chosen by any onlooker bee. When a $cycle_i$ reaches the maximum cycle number mcn, the scouts eliminate the food and replace it with a new food source.

5. To produce a new potential food position from an old source food, the artificial bees employ the following expression:

$$pfood_i = p_i + \varphi(p_i - p_s) \tag{5}$$

In the MRC-ABC, we exploit the same technique used in MRC-ACO, all p_i are split to sub-vectors, and the same stage one and two of MRC-ACO are adopted in MRC-ABC, however, to imitate the dynamics of ABC the first assignment of clusters is done randomly. After the accomplishment of stage one and two, all vectors possess a cohesion value of the clusters assignment.

The stage three is devoted to computing $nectar_i$ and pb_i for each food source. To compute the $nectar_i$ each mapper receives all the computed cohesion values; then they transmit the results to the reducers as shown in algorithm 2.

Afterward, each single reducer receives all the $nectar_i$ computed in the mappers step and calculate for each p_i the value pb_i, as stated in algorithm 3. The number of mappers/reducers required in this round equals to m. By loading only the cohesion values in each mapper/the nectar amounts in each reducer, we drastically reduce the size of the loaded values into the mapper's/reducer's memory compared to the first and the second stage.

Algorithm 2: MRC-ABC: stage three

Mapper: Do in parallel for each $mapper_i$

Input: $< key : cohesion_i, Listvalues : [cohesions] >$

compute $nectar_i$ for the p_i using eq. 3;

emit(i, $nectart_i$) to all reducers;

Reducer: Do in parallel for each $reducer_i$

Input: $< key : nectar_i, Listvalues : [nectars] >$

compute pb_i for the p_i using eq. 4;

After achieving stage three, all p_i have a completed clustering solution with its $nectar_i$ and pb_i assessed in the memory. In the last round, the onlooker bees choose one solution according to its p_i value and send only the chosen nectar to the reducers, if p_i is not chosen, the variable $cycle_i$ is incremented, see algorithm 3.

Algorithm 3: MRC-ABC: stage four

Mapper: Do in parallel for each $mapper_i$

Input: $< key : i, value : (pb_i, cycle_i) >$

if *pb_i is chosen* **then**

emit(i,pb_i);

else

$cycle_i = cycle_i + 1$;

end

Reducer: Do in parallel for each $reducer_i$

Input: $< key : i, value : 0 >$

if *pb_i is chosen* **then**
produce new solution $Pfood_i$ using expression 5;

else

if *$cycle_i$ equals to mcn* **then**
produce a random solution and replace p_i ;

end

end

The reducers of the last stage imitate the scouts, they receive all the chosen pb_i and produce for them new clustering solution $pfood_i$ using equation 5. The reducers also check if any $cycle_i$ has reached the maximum cycle number mcn. If so, the solution is considered abandoned and replaced by a randomly generated one, see algorithm 3.

4.3. The Generalized island model, based MapReduce

GIM (Generalized Island Model) is a parallel model which can be applied to a large class of optimization problems. Based on metaheuristics cooperation, it allows an effective parallel execution of multiple algorithms across multiple islands. This cooperation is maintained by the exchange of solutions between these islands. The exchange operator (migration operator) aims at improving the overall performance of the different algorithms used [8].

To let cooperation, we set up in parallel our proposed MRC-ACO and MRC-ABC and from the literature MR-CPSO proposed in [17]. Then integrate an exchange operator to enable solution migration from one algorithm to another. Each algorithm act as an island and islands cooperate between them by selecting a subset of solution (selection strategy S i.e. elitist), exchanging selected solutions using the migrant operator and finally recombine them with local solutions following a recombination strategy R (i.e. replacing worst solutions). We conceive this model in Hadoop by managing the HDFS, which will be logically divided using Centralized Cache Management (CCM). We create four logical memories. Three of them represent a distributed memory associated with each algorithm. One shared memory between the algorithms serving as a migrant solution deposit which acts as a buffer to exchange solutions in an asynchronous way (figure 4).

Without deploying a barrier operations, no processor will exhibit idle waiting for another one. Thus, this technique will increase the load balancing yielding to

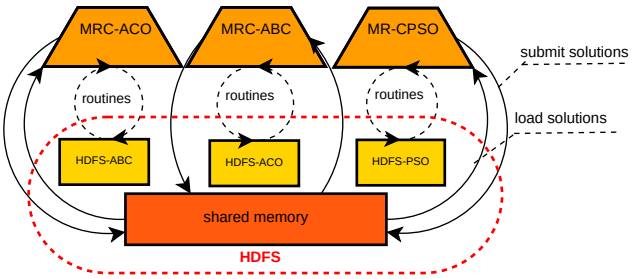


Figure 4: HDFS management for Generalized Island Model in Hadoop architecture (MRC-GIM)

the speediest system. The dynamics of MRC-GIM is described in algorithm 4.

Algorithm 4: Do in parallel for MRC-ACO, MRC-ABC, and MR-CPSO

P :initial population;

S :selection strategy;

R :recombination policy;

A_i : MRC-ACO, MRC-ABC and MR-CPSO;

initalize P; `// initialize population`

while *non stop criteria* **do**

$P' = A_i(P)$; `// P' is the new population`

$M = S(P')$; `// select M solutions from P'`

submit (M) in the Shared HDFS meomory; `// Sharing the results with others SI`

load M' from the HDFS; `// extract the shared solutions`

$P'' = R(P', M')$; `// combine M' with P' to generate new population`

$P = P''$;

end

The recombination policy consists of combining the migrant populations with the local ones to get more diversity for the next iteration which is the key task of cooperation.

5. Experiments

In the experiment design, we analyze the efficiency of the proposed algorithms. First, we study the amount of resource such as time, the number of processing elements and iterations required to perform clustering. The algorithms are designed to be executed on different platforms with different size of data. Therefore, we study the scalability, speedup, and the sizeup of the algorithms. Finally, we compare the meta model MRC-GIM to different state-of-the-art large scale clustering algorithms. The clustered data are two sets with different size to test the ability of the algorithms to group both large and small sets of data. The following subsection explains briefly these sets of data.

5.1. Datasets

We validate the proposed algorithms by clustering available large datasets. The data set used have been downloaded from the Stanford Network Analysis Project (SNAP) [32]. To assess the quality of clustering both small and large data sets. We employed two sets of data Friendster and DBLP as shown in table 2.

Table 2: The datasets used in our experiments

Dataset	Records	Cluster	Size
Friendster	65,608,366	287,512	30.1 GB
DBLP	317,080	13,477	10 MB

Friendster is a gaming website, which was a social networking website that enabled users to connect with their friends. The data was released at the end of June 2011, which contains 65,608,366 users. Each line of the data files represents the friends list of one user, in the following format: (id of user: separated list of user's friends), for example, $1 : 2, 3$ means that the user with $id = 1$ has a connection with both the users two and three. We preprocessed the data to be suitable for partitioning clustering by constructing for each player an ascending

order of all players following their *id* and then, check weather a connection is found between this player and the *ith* player if so the number one is affected elsewhere 0. A player is considered to have a connection with himself.

DBLP is a computer science bibliography that provides a comprehensive list of research papers in computer science field. Two authors are connected if they publish at least one paper together. We followed the same preprocessing to prepare the DBLP data for the partitioning clustering.

The number of potential clustering solutions within the search area for the Friendster data is $\frac{287{,}512^{65{,}608{,}366}}{287{,}512!}$, which renders the clustering more challenging task. We conduct our experiments on the EMR cluster of Amazon with 192 nodes where each node has 2.5 GHZ processor and 1.7 Go memory. The data was moved from local disk to Amazon using the Jets3t Java library. The size of the memory limits the number of map jobs to be used per node. We split each node memory to 20 blocks where each block is a 64 MB (HDFS block). Hadoop uses HDFS data blocks and assigns a single HDFS block to each map task, for the Friendster data the minimal configuration is to use 24 nodes with 20 map jobs per each node.

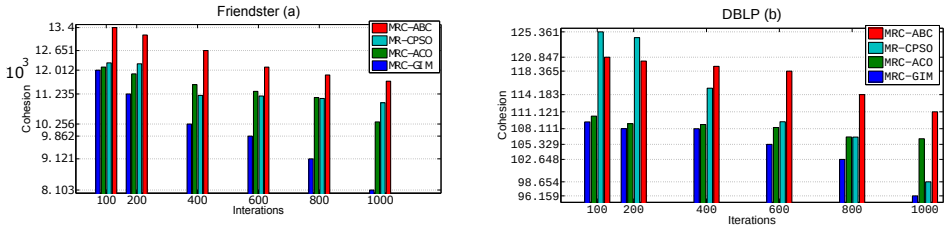
5.2. Evaluating each swarm intelligence algorithm individually

To evaluate the convergence of the proposed MRC-ABC, MRC-ACO, MRC-GIM, we recorded the cohesion by tweaking the number of iterations from 100 to 1000 at each time (we stooped at 1000 iterations following a limitation of the platform used). A bar chart of the cohesion against the number of iterations is shown in figure 5(a).

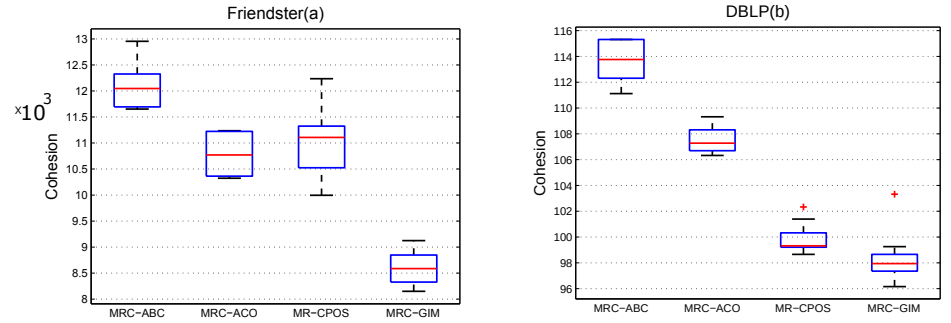
Regarding Friendster data, MRC-GIM returned the higher clustering quality. As the number of iterations increases the gap between MRC-GIM and the rest of algorithms increases as well (figure 5(a)). Increasing the number of iterations involves enhancing the chance of solution migration thus exploring the search space more efficiently using three different swarm strategies at the same time. Moreover, the cooperative scheme of MRC-GIM improves its quick convergence while avoiding local minima, e.g., the results reached by MRC-GIM in 200

iterations, can be achieved by MR-CPSO after around 800 iterations.

(a)



(b)



(c)

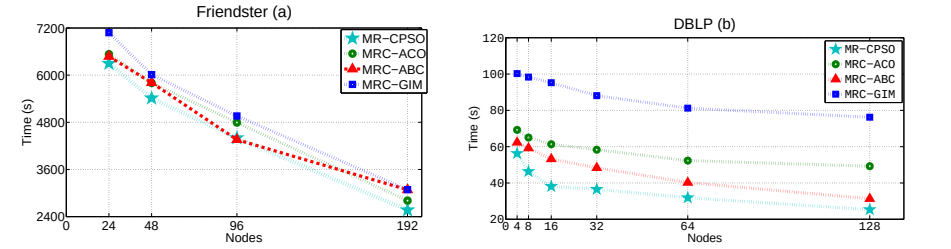


Figure 5: (a) A Bar chart showing the cohesion value against the number of iteration, (b) Boxplots displays the distribution of results over 18 runs using both Friendster and DBLP data, (c) The Parallel runtime recorded by seconds while increasing the processing elements at each step.

Regarding DBLP data, the algorithms are more competitive as the search space is restricted compared to Friendster data. MRC-ACO and MR-CPSO are competitive; MRC-ABC achieved the highest (worst) cohesion value. Subse-

quent, we set up MRC-GIM with only MR-CPSO and MRC-ACO to evaluate whether MRC-GIM converges better with isolating MRC-ABC. The results for 100, 200, and 400 iterations are 12.352, 12.110 and 11.461 respectively, indicating that even MRC-ABC, which returned the worst result, improves the convergence of the overall MRC-GIM alongside MR-CPSO and MRC-ACO.

Furthermore, a statistical analysis is conducted to test the randomness of the algorithm outputs. The test illustrates the distributions of cohesion in terms of box-plots over 18 runs and 1000 iterations on 192 computers as shown in figure 5(b). MRC-GIM boxes indicate that the returned results are symmetric (roughly the same on each side when cutting down the middle) therefore, it returns more robust and stable median value compared to other skewed boxes.

Afterwards, the parallel runtime is recorded in figure 5(c), it depends on both the input size of the data and the number of processing elements used. The recorded parallel runtime is for 600 iterations. MRC-GIM requires slightly more computational time compared to the rest of algorithms since it performs additional tasks such as selection strategy, recombination policy, transmission and integration of migrant solutions. However as the number of processing elements increases the gap between the plots shrinks, this difference becomes very narrowed within the nodes 96 and 192. In DBLP data sets, the processing elements surplus the requirement of data, therefore, a small gain in seconds has been recorded.

In the rest of this section, we evaluate, the speed up, scaleup and sizeup. The speedup is defined as the ratio of time recorded to solve a parallel task to the time required to solve the same problem on a double set of processing elements with the same capacity of processing nodes [33], following the expression 6.

$$Speedup = \frac{T_N}{T_{2N}} \tag{6}$$

Where T_N is the running time using N nodes and T_{2N} is the running time using 2-fold of N. The speedup measures the acceleration of the parallel algorithms from small computational power to a larger one while maintaining the same size of the dataset. We performed the speed up by doubling the number

of nodes used at each step, starting from 24 nodes for Friendster and four nodes in DBLP as the minimum required configurations.

While clustering DBLP data sets, all algorithms achieved a speedup larger than one (figure 6). Therefore, they are all scalable. In Friendster data, MRC-GIM reached the best speedup imitating the diagonal (figure 6). The MapReduce implementation has helped the algorithms to achieve a good speedup as MapReduce paradigm is proved to be auto-scalable.

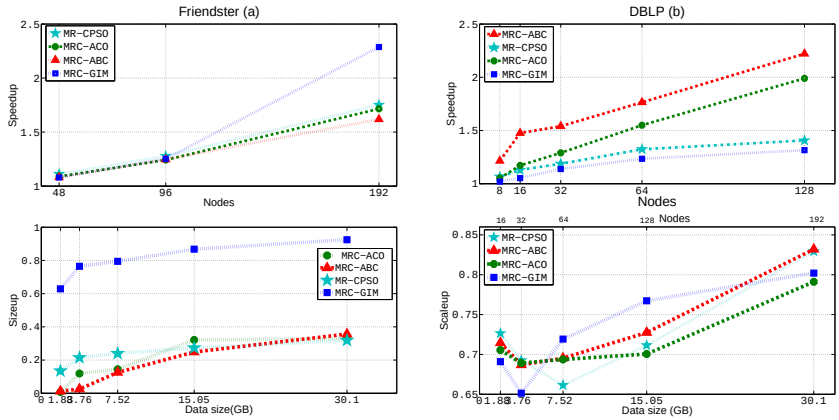


Figure 6: Plots showing the speed up, size up and scale up metrics. The speed up is computed for both data sets

Secondly, we computed the size up by fixing the number of nodes and increasing the size of datasets by 2-folds at each step [33].

$$Sizeup = \frac{T_S}{T_{2S}} \tag{7}$$

Where TS is the parallel runtime for the dataset with size S using N nodes and T_{2S} is the parallel runtime using 2-fold of S and N nodes. MRC-GIM deals well with the increase of data, whereas the rest of algorithms recorded very close result compared to each other. Finally, We compute the scaleup which evaluates the ability of a parallel algorithm to solve a significant problem using larger datasets and more processing elements at once [33].

$$Scaleup = \frac{T_{SN}}{T_{2SN}} \quad (8)$$

Where T_{SN} is the parallel runtime for the dataset with the size S using N nodes and T_{2SN} is the parallel runtime using 2-fold of S and 2-folds of N nodes. To carry out the experiment, we decreased the size of Friendster data. We first cluster the data using kmeans. Afterward, we balance between the number of elements present in each group until reaching 0.94 GB of data (1/32 of the original size). Then adding a proportion of the deleted elements to reach 1.88GB (1/16), 3.76 GB (1/8), 7.52GB (1/4), 15.05 GB (1/2), and 30.1 GB (1/1). This data is processed in 8, 16, 32, 64, 128, and 192 nodes respectively (since the limitation of the platform we used 192 nodes rather than 256 nodes).

MRC-ABC recorded the best values between the ratio (0.69-0.79), MRC-GIM also shows a good scaleup between (0.65-0.80) as shown in figure 6.

5.3. Comparative study

We compare MRC-GIM to Lloyds algorithm with the partitioning based scheme proposed in [16], PKmeans [12], BigKclustering [21] and Kmeans ++ with the improved version proposed in [25]. We compare the clustering quality by gathering the cohesion values and the recorded time after 50 runs for 100, 200, 400, 600, 800, and 1000 iterations. The results are shown in table 3. The number of processing elements used in this section is 124.

Table 3: Table shows the recorded cohesion/time required in second. All the cohesion values are divided by 10^3.

Iterations	**MRC-GIM**	**Lloyd**	**BigKClustering**	**Pkmeans**	**Kmeans++**
100	12,012/375s	12,376/398s	12,509/328s	12,835/287s	13,115/264s
200	11.235/789s	11,773/811s	12,287/720s	12,482/649s	12,788/631s
400	10,259/1665s	11,109/1793s	11,587/1603s	12,095/1587s	12,546/1466s
600	9,862/2783s	10,832/2833s	10, 865/2667s	11,311/2583s	11,840/409s
800	9,121/3597s	9,563/3786s	9, 745/3422s	9,976/ 3389s	10.248/3266s
1000	8,213/4357s	8,482/4610s	8,69/4531s	8,712/4387s	8,911/4198s

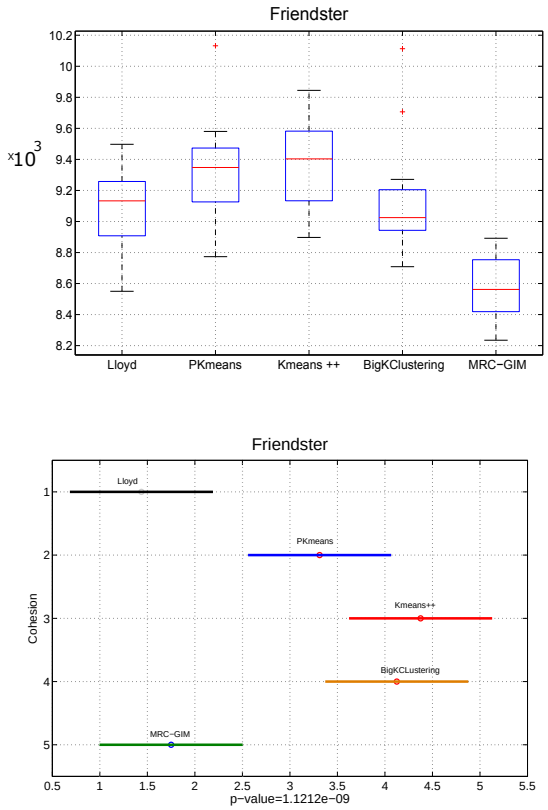


Figure 7: Boxplots showing distribution of results over 50 runs and 1000 iterations and the corresponding multi-comparison graph for MRC-GIM, BigKCLustering, PKmeans, Lloyd and Kmeans++ based MapReduce

The same initial clustering is used (also the same number k of clusters). Regarding clustering quality, MRC-GIM outperforms the rest of algorithms reaching good partition in a reasonable time. We were unable to run the algorithms for more than 1000 iterations as a result of the platform limitation. Kmeans++ returned many outliers; the best solutions returned in 50 runs are selected in table 3. Kmeans ++ and PKmeans recorded the shortest time as they follow the classical Kmeans algorithm and run in one iterative MapReduce job.

In figure 7, the distribution of results for 1000 iterations and 50 runs on 124 computers are recorded. Regarding median value, the proposed model outper-

forms the four algorithms and shows more stability and robustness. To study how significant is the difference between the algorithms, a Friedman test followed by a multi-comparison test were conducted. The p-values given by the Friedman indicates a significant difference between the proposed model and the other algorithms at significance level. The multi-comparison graphs clearly show that our algorithm is significantly different from the rest algorithms. This gain shows the advantage of using the cooperation and parallelism between metaheuristics on a MapReduce platform.

5.4. MRC-GIM reveals slight changes in DNA-Methylation across age

In this section, we use MRC-GIM to cluster real epigenetics data to investigate the correlation between aging and epigenetics. Formally, aging is a time-dependent degenerative process characterized by cellular senescence that leads to progressive functional decline, reduced stress response, homeostatic imbalance, and increased susceptibility to disease [35]. In the last decade, a growing body of research has revealed a tight connection between epigenetic factors, defined as the set of mitotically and meiotically heritable changes in gene expression that do not depend on alteration in the DNA sequence [36, 37], and aging-related phenotypes and diseases [38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. For example DNA methylation, the most widely studied epigenetic mark consisting in the addition of a methyl group to the fifth position of cytosine in CpG dinucleotides shows a progressive drift with age in humans, characterized by global hypomethylation and site-specific hypermethylation in promoter regions associated with tumor suppressor genes [51, 52, 53, 54, 55]. Nevertheless, the causal relationship between epigenetic alterations in normal individuals and aging remains poorly understood, as changes are bidirectional, not uniform across the genome and highly dependent on individual lifestyle and environment [56].

In the effort to assess the dynamics of DNA methylation as a function of age and underlying pathological events, we extended our cluster analysis to the methylation profiles from CD34 primary cells assayed by bisulfite sequencing in six normal individuals ranging in age from 27 to 50 years (Table 4).

Table 4: DNA methylation data.

Sample Code	Age	Sex	Ethnicity	Methylated BPs	Data Size (Mb)
GSM706858	27	female	Caucasian	2,729,614	118.3
GSM772729	27	female	Hispanic	2,464,116	106.7
GSM706859	36	male	Hispanic	3,047,608	132.2
GSM772730	42	male	Caucasian	2,690,026	116.6
GSM706839	43	male	Caucasian	2,920,722	126.6
GSM772731	50	female	na	2,726,708	118.1

In particular, since age-associated alterations of the epigenome are strictly related to inflammatory processes [44, 57, 58], we analyzed the methylation behavior of 139 genes from chemokines, interleukins, tumor necrosis factor, and tgf-β families, to assess how methylation levels in pro-inflammatory mediators vary with age. Due to the high decomposability of a DNA methylation analysis within each individual, we were able to parallelize the recognition of methylation features in CpG islands, gene promoters, and gene bodies for the entire dataset, optimizing the computational burden across the nodes.

We first examined the average value of methylation[4] in CpG islands, gene body, and gene promoter for the 139 target genes in each individual (Figure 8, a). We found a global low level of methylation in gene bodies and CpG islands and a nearly unmethylated state in promoters. Even though methylation in the gene body increases with age, we did not identify any significant trend or change related to aging, as values remain consistently near to zero. Therefore, we can argue that pro-inflammatory mediators in normal individuals exhibit constant low methylation, especially in promoter regions and CpG islands, confirming previous general evidence [59, 60].

[4]Methylation values range from 0 to 1.

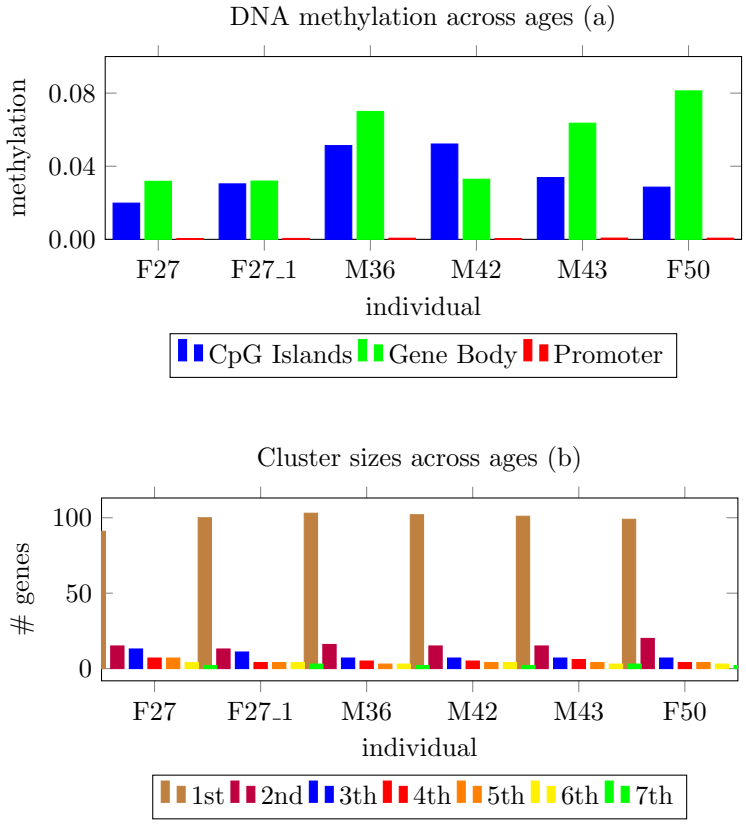


Figure 8: (a): DNA methylation levels in CpG islands, gene body, and promoter regions across ages. (b): Cluster sizes as function of DNA methylation features and age.

We then performed cluster analysis for the 139 target genes in each individual to investigate how the formation of groups is guided by DNA methylation features (corresponding to values in CpG islands, gene body, and gene promoter) over time and whether epigenetic signatures emerge in specific clusters as a byproduct of aging. From the examination of the size of clusters (Figure 8,b), we noticed the emergence of a giant cluster (containing 65% to 74% of all genes) for any individual, suggesting the presence of a general methylation pattern at each age. In fact, elements in the largest cluster in each individual correspond to nearly unmethylated genes in CpG islands, gene body, and gene promoter and surprisingly these genes remain in the largest cluster over time, as

showed by measuring the Jaccard index[5] between the largest clusters for each pair of individuals (Table 5).

Table 5: Jaccard index for the largest clusters across ages.

	F27	F27-1	M36	M42	M43	F50
F27	1.00	0.91	0.88	0.89	0.90	0.92
F27-1	0.91	1.00	0.92	0.96	0.95	0.91
M36	0.88	0.92	1.00	0.92	0.92	0.96
M42	0.89	0.96	0.92	1.00	0.93	0.93
M43	0.90	0.95	0.92	0.93	1.00	0.92
F50	0.92	0.91	0.96	0.93	0.92	1.00

Finally, we studied how methylation features (corresponding to values in CpG islands, gene body, and gene promoter) vary across clusters and ages (Figure 9). While the largest cluster in every individual shows nearly unmethylated features, clusters tend to exhibit changes in methylation in CpG islands and gene body as size decreases, with a significant peak in one of the two features. In particular, we found two recurrent clusters of genes across ages: the first cluster, characterized by an increased level of methylation in the gene body, is composed of three genes belonging to tnf family (TNFRSF14, TNFRSF6B, TNFRSF25) and two genes in tgf-β family (AMH, PSPN), while the second cluster, characterized by an increased level of methylation in CpG islands, is composed of two genes in tnf family (TNFRSF4, TNFRSF11A) and four genes in tgf-β family (INHBB, TGFB1, GDF7, GDF10). A gene enrichment analysis confirmed the strict relationships between genes in each of the two clusters in a wide number of biological processes. Specifically, genes in the first cluster are involved in tumor necrosis factor-mediated signaling pathway, urogenital sys-

[5]The Jaccard index measures the similarity between finite sets. Formally, given two sets A and B, the Jaccard index is defined as: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$.

tem development, and more generally in the response to chemical, while genes in the second cluster are involved, among others, in the regulation of protein phosphorylation and apoptotic process.

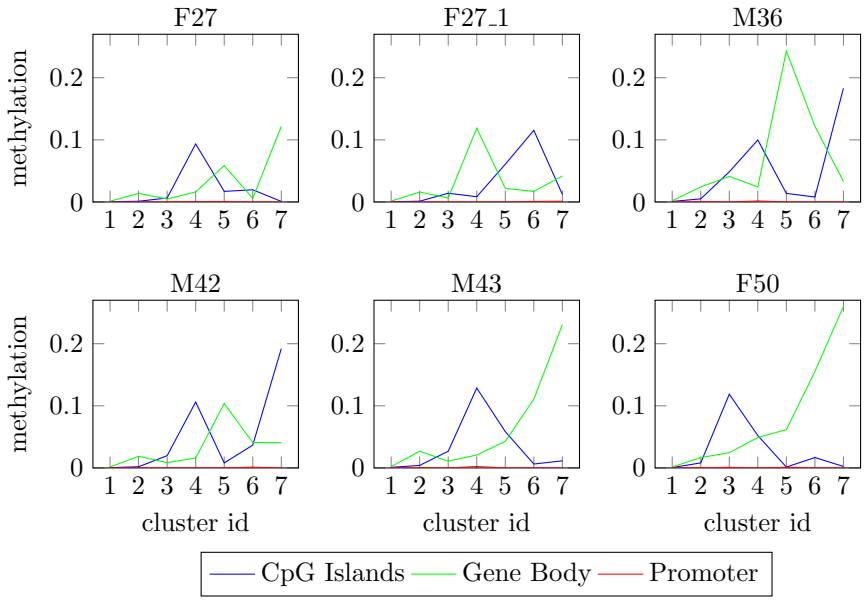


Figure 9: DNA methylation levels in CpG islands, gene body, and promoter regions in different clusters and different individuals.

6. Conclusions

The amount of data is growing further in size and complexity. The recent advances in collecting data have triggered an extreme need for novel data analysis methods to decipher the complexity of these data. Undoubtedly, parallelization frameworks and distributed computation will play a primary role in the future of interpreting big data in general and biological data in particular. For example, the cloud computing will help to scale up the analysis of epigenetics profiles to larger samples of individuals, providing a more detailed characterization of epigenetic mechanisms in fast and cost-efficient ways.

In this paper, we deal with the issue of clustering big sets of data. Data

clustering itself is an NP-hard problem, and it becomes more challenging when the clustered data is large. Moreover, we present MRC-GIM, which is a scalable, time-efficient, and robust clustering approach to group very large datasets on a different commodity of computer architectures. The proposed method aggregates three swarm intelligence strategies, helping to avoid local convergence issues and making the algorithm reaches an optimum solution in a shorter time. MRC-GIM is designed and implemented with Hadoop MapReduce, which makes MRC-GIM auto-scalability and fault-tolerance. The comparative study reveals that MRC-GIM outperforms novel developed big-scale clustering in a reasonable time. Although Hadoop exists for a half decade, a noticeable shortage of tools dealing with big data analysis has been witnessed.

We extended our parallelization strategy to the study of DNA methylation in pro-inflammatory mediators from a population of normal individuals ranging in age from 27 to 50 years. Exploiting the high decomposability of the dataset, we were able to analyze simultaneously millions of nucleotides, retrieve methylated cytosines, and cluster genes according to methylation features in CpG islands, gene body, and gene promoter. We found a global low level of methylation in CpG islands and gene body, and a nearly unmethylated status in promoter regions for each, suggesting that the epigenetic landscape in normal condition should not change aberrantly with aging.

7. References

[1] Steinhaus, H.: Sur la division des corps materiels en parties, Bulletin de l'Academie Polonaise des Sciences, Volume IV, Pages 801-804, Issue 12 (1956)

[2] Nanda, S.J., Panda, G.: A survey on nature inspired metaheuristic algorithms for partitional clustering, Swarm and Evolutionary Computation, Volume 16, Pages 1-18 (2014)

[3] Nir, A., Bernard, C.: Faster Dimension Reduction, communications of the acm, Volume 53, Pages 97-104, Issue 2 (2010)

[4] George, K., Dimitrios, G.: Nick Koudas, Stefan Berchtold, Efficient Biased Sampling for Approximate Clustering and Outlier Detection in Large Data Sets, IEEE Transactions on Knowledge and Data Engineering, Volume 15, Pages 1170-1187, Issue 5 (2003)

[5] Sarkar, S., Majumder, T., Kalyanaraman, A., Pande, P.: Hardware accelerators for biocomputing: A survey, Proceedings of IEEE International Symposium on circuits and systems (ISCS), Pages 3789-3792 (2010)

[6] Dean, J.,Ghemawat, S.: MapReduce: simplified data processing on large clusters, Communications of the ACM, Volume 51 Issue 1,Pages 107-113 (2008)

[7] MacQueen, J. B.:Some Methods for Classiffication and Analysis of MultiVariate Observations, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press vol. 1, Pages 281-297 (1967)

[8] Kaufman, L., and Rousseeuw, P.:Clustering by means of medoids, in Statistical Data Analysis based on the L1-norm and Related Methods ,North-Holland,Pages 405-416 (1987)

[9] Dario, I., Marek, R., Francesco, B.: The Generalized Island Model. Parallel Architectures and Bioinspired Algorithms 415, 151169 (2012)

[10] Das, A. S., Datar, M., Garg, A., and Rajaram, S.: Google News Personalization: Scalable Online Collaborative Filtering. Proceedings of the 16th international conference on World Wide Web ACM, Pages 271-280 (2007)

[11] Papadimitriou, S., Sun, J.:DisCo: Distributed Co-clustering with Map-Reduce , Eighth IEEE International Conference on Data Mining (ICDM'08), Pages 512-521 (2008).

[12] Zhao, W., Ma, H., He, Q.: Parallel K-Means Clustering Based on MapReduce, Cloud Computing, LNCS 5931, Pages 674-679 (2009)

[13] Ferreira Cordeiro, R. L., Traina Junior, C., Machado Traina, A. J., Lpez, J., Kang, U., Faloutsos, C.:Clustering very large multi-dimensional datasets with mapreduce, Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, Pages 690-698 (2011)

[14] Li, H. G., Wu, G. Q., Hu, X. G., Zhang, J., Li, L., Wu, X.:K-Means Clustering with Bagging and MapReduce. 44th Hawaii International Conference on System Sciences (HICSS),Pages 1-8 (2011)

[15] He, Y., Tan, H., Luo, W., Mao, H., Ma, D., Feng, S., and Fan, J.: MR-DBSCAN: An Efficient Parallel Density-based Clustering Algorithm using MapReduce. 17th International Conference on Parallel and Distributed Systems (ICPADS),Pages 473-480 (2011)

[16] Ene, A., Im, S., Moseley, B.: Fast Clustering using MapReduce. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, Pages 681-689 (2011)

[17] Aljarah, I., Ludwig, S.:Parallel Particle Swarm Optimization Clustering Algorithm based on MapReduce Methodology, Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC), Pages 104-111 (2012)

[18] Bahmani, B., Moseley, B., Vattani, A., Kumar, R.,:Scalable KMeans++, Proceedings of the VLDB Endowment, Volume 5, Issue 7, Pages 622-633 (2012)

[19] Hefeeda, M., Gao, F., Abd-Almageed, W.:Distributed Approximate Spectral Clustering for Large-Scale Datasets, Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing ACM, Pages 223-234 (2012)

[20] Estrada, T., Zhang, B., Taufer, M., Cicotti, P., Armen, R.:Reengineering high-throughput molecular datasets for scalable clustering using MapReduce, IEEE 14th International Conference on High Performance Computing and Communication,Pages 351-359 (2012)

[21] Miao, Y., Zhang, J., Feng, H., Qiu, L., Wen, Y.: A Fast Algorithm for Clustering with MapReduce, Advances in Neural Networks, Pages 532-538, (2013)

[22] Jin, C., Patwary, M. M. A., Agrawal, A., Hendrix, W., Liao, W. K., Choudhary, A.: DiSC: A Distributed Single-Linkage Hierarchical Clustering Algorithm using MapReduce, ACM 4th International SC Workshop on Data Intensive Computing in the Clouds (2013)

[23] Jakovits, P., Srirama, S. N.: Clustering on the Cloud: Reducing CLARA to MapReduce, ACM Proceedings of the Second Nordic Symposium on Cloud Computing, Pages 64-71 (2013).

[24] Herwig, R., Poustka, A. J., Mller, C., Bull, C., Lehrach, H., O'Brien, J. :Multiplex Kmeans for Clustering Large-scale Data Set, The journal of Genome research,Volume 9 Issue 11, Pages 1093-1105 (2014)

[25] Xu, Y., Qu, W., Li, Z., Min, G., Li, K., Liu, Z.:Efficient K-means++ Approximation with MapReduce, IEEE Transactions on Parallel and Distributed Systems, Volume 25 Issue 12, Pages 3135-3144 (2014)

[26] Kim, Y., Shim, K., Kim, M. S., Lee, J. S.: DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce, The journal of Information Systems, Volume 42, Pages 15-35 (2014)

[27] Cui, X., Zhu, P., Yang, X., Li, K., and Ji, C.: Optimized big data K-means clustering using MapReduce, The journal of Supercomputing, Volume 70 Issue 3, Pages 1249-1259 (2014)

[28] Dorigo, M.:Optimization, learning and natural algorithms. Ph.D. thesis Politecnico di Milano (1992)

[29] Dorigo, M., Birattari, M., Stutzle, Thomas.: Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique, IRIDIA (2006)

[30] Kuo, R. J., Wang, H. S., Hu, T. L., Chou, S. H.:Application of ant K-means on clustering analysis, Journal of Computers and Mathematics with Applications, Volume 50 Issue 10,Pages 1709-1724 (2005)

[31] Karaboga, D., Ozturk, C.:A novel clustering approach: Artificial Bee Colony (ABC) algorithm, journal of Applied soft computing, Volume 11 Issue 1, Pages 652-657 (2011)

[32] Leskovec, J., Krevl, A.:SNAP Datasets: Stanford Large Network Dataset Collection, $http://snap.stanford.edu/data$ (2014)

[33] Grama, A., Gupta, A., Karypis, G., Kumar, V.: Introduction to Parallel Computing. Addison-Wesley (2003)

[34] Rivest, R,: The MD5 message-digest algorithm (1992)

[35] Lpez-Otn, C., Blasco, M. A., Partridge, L., Serrano, M., and Kroemer, G.:The Hallmarks of Aging, Cells, vol.153, Pages 1194-1217 (2013).

[36] Holliday, R.:The inheritance of epigenetic defects, Science, vol. 238, Pages 163-170 (1987)

[37] Feinberg, A.P.:Phenotypic plasticity and the epigenetics of human disease, Nature , vol. 447, Pages 433-440 (2007)

[38] Ahuja, N., Li, Q., Mohan, A. L., Baylin, S. B., and Issa, J. P. J.:Aging and DNA Methylation in Colorectal Mucosa and Cancer, Cancer Research, vol.58, Pages 5489-5494 (1998)

[39] Issa, J.P.:Epigenetic Variation and Human Disease, The Journal of Nutrition, vol. 132, Pages. 2388-2392 (2002)

[40] Issa, J. P.:Age-related epigenetic changes and the immune system, Clinical Immunology, vol. 109, Pages. 103-108 (2003)

[41] Bollati, V., Schwartz, J., Wright, R., Litonjua, A., Tarantini, L., Suh, H., Sparrow, D., Vokonas, P., and Baccarelli, A.:Decline in genomic DNA

methylation through aging in a cohort of elderly subjects, Mechanisms of Ageing and Development, vol. 130, Pages 234-239 (2009)

[42] Christensen, B. C., Houseman, E. A., Marsit, C. J., Zheng, S., Wrensch, M. R., Wiemels, J. L., Nelson, H. H., Karagas, M. R., Padbury, J. F., Bueno, R., Sugarbaker, D. J.,Yeh, R. F., Wiencke, J. K., and Kelsey, K. T.:Aging and Environmental Exposures Alter Tissue-Specific DNA Methylation Dependent upon CpG Island Context, PLoS Genetics, vol. 5 (2009)

[43] El Mezayen, R., El Gazzar, M., Myer, R., and High K. P.:Aging-dependent upregulation of il-23p19 gene expression in dendritic cells is associated with differential transcription factor binding and histone modifications, Aging Cell, vol. 8, pp. 553-565 (2009)

[44] Agrawal, A., Tay, J., Yang, G.-E., Agrawal, S., and Gupta, S.:Age-associated epigenetic modifications in human DNA increase its immunogenicity, Aging, vol. 2, Pages 93-100 (2010)

[45] Rakyan, V.K., Down, T.A., Maslau, S., Andrew, T., Yang, T.-P., Beyan H., Whittaker, P., McCann, O.T., Finer, S., Valdes, A. M., Leslie, R.D. Deloukas, P., and Spector, T.D.:Human aging-associated DNA hypermethylation occurs preferentially at bivalent chromatin domains, Genome Research, vol.20, Pages 434-439 (2010)

[46] Muoz-Najar, U., and Sedivy, J. M.: Epigenetic Control of Aging, Antioxidants & Redox Signaling, vol. 14, Pages 241-259 (2011)

[47] Ben-Avraham, D., Muzumdar, R. H., and Atzmon, G.:Epigenetic genome-wide association methylation in aging and longevity, Epigenomics, vol. 4, Pages 503–509 (2012).

[48] Berdasco, M., and Esteller, M.:Hot topics in epigenetic mechanisms of aging: 2011, Aging Cell, vol. 11, Pages 181-186 (2012)

[49] Florath, I. , Butterbach, K., Mller, H., Bewerunge-Hudler, M., and Brenner, H.:Cross-sectional and longitudinal changes in dna methylation with age:

an epigenome-wide analysis revealing over 60 novel age-associated cpg sites, Human Molecular Genetics, vol. 23, no. 5, Pages 1186-1201 (2014)

[50] Jung, M., and Pfeifer, G. P.:Aging and DNA methylation, BMC Biology, vol. 13, no. 7 (2015)

[51] Wilson, V. L., Smith, R. A., Ma, S., and Cutler, R. G.: Genomic 5-methyldeoxycytidine decreases with age, Journal of Biological Chemistry , vol. 262, Pages 9948–9951, Jul (1987)

[52] Issa, J.-P. J. , Ottaviano, Y. L., Celano, P., Hamilton, S. R., Davidson, N. E., and Baylin, S. B.:Methylation of the oestrogen receptor CpG island links ageing and neoplasia in human colon,Nature Genetics, vol. 7, Pages 536-540 (1994)

[53] Bjornsson, H., Sigurdsson, M., Fallin, M., Irizarry, R., Aspelund, T., Cui, H., Yu, W., Rongione, M., Ekstrm, T., Harris, T., Launer, L., Eiriksdottir, G., Leppert, M., Sapienza, C., Gudnason, V., and Feinberg, A.:Intra-individual change over time in DNA methylation with familial clustering, JAMA, vol. 299, Pages 2877-2883 (2008)

[54] Grnniger, E., Weber, B., Heil, O., Peters, N., Stb, F., Wenck, H., Korn, B., Winnefeld, M., and Lyko, F.: Aging and chronic sun exposure cause distinct epigenetic changes in human skin, PLoS Genetics, vol. 6 (2010)

[55] Hernandez, D. G., Nalls, M. A., Gibbs, J. R., Arepalli, S., van der Brug, M., Chong, S., Moore, M., Longo, D. L., Cookson, M. R., Traynor, B. J., and Singleton, A. B.:Distinct DNA methylation changes highly correlated with chronological age in the human brain, Human Molecular Genetics, vol. 20, no. 6, Pages . 1164–1172 (2011)

[56] Issa, J.-P.: Aging and epigenetic drift: a vicious cycle, The Journal of Clinical Investigation, vol. 124, Pages 24–29 (2014)

[57] Bayarsaihan, D.: Epigenetic Mechanisms in Inflammation,Journal of Dental Research, vol. 90, Pages. 9–17 (2011)

[58] Shanmugam, M., and Sethi, G.:Role of Epigenetics in Inflammation-Associated Diseases,in Epigenetics: Development and Disease, vol. 61 of *Subcellular Biochemistry*, Pages. 627-657, Springer Netherlands (2013)

[59] P. A. Jones and P. W. Laird, "Cancer-epigenetics comes of age," *Nature Genetics*, vol. 21, pp. 163–169, (1999).

[60] Jones, P. A., and Baylin, S. B.:The fundamental role of epigenetic events in cancer, Nature Reviews Genetics , vol. 3, Pages 415-428 (2002)

Cover Letter

25th of April 2016

Dear Editor-in-Chief of the International Journal "Engineering Applications of Artificial Intelligence"

Please find attached for your kind review our manuscript entitled "Parallel Swarm Intelligence Strategy for Large-scale Clustering based on MapReduce with Application to Epigenetics of Aging" for the special issue on Evolutionary Multi-objective Optimization and Applications in Big Data (**SI:EMO-BD**).

Authors: Zakaria Benmounah [1,3], Souham Meshoul[1], Mohamed Batouche[1], and Pietro Lio'[2]
[1]Computer science department, Constantine 2 University, Constantine, Algeria.
[2]Computer Science Department, Cambridge University, Cambridge, UK.
[3]MISC laboratory.

The unprecedented wealth of nucleic data has generated an enormous demand for stand-alone tools and methods to analyze and decipher the complexity of these large data. One of these methods is data clustering. Data clustering is an NP-hard problem, and it becomes more challenging as the size of data rises.

There is a large literature on data mining in general and data clustering in particular. However, the lack of big data clustering algorithms is one of the deficiencies in the state-of-the-art nowadays. In this paper, we present a parallel, scalable methods devoted to cluster large data sets based on MapReduce paradigm. The methods combine between the swarm intelligence strategies such as Ant Colony Optimization, Artificial Bee Colony, and Particle Swarm Intelligence to explore the vast clustering research space more efficiently. The proposed methods are tested on real big data platform with up to 192 computers to cluster large data sets.

The major contributions of the submitted paper are:

1- We present and discuss a review of big data clustering algorithms based upon MapReduce.
2-A highly scalable clustering algorithms designed for large-scale data. The methods were tested on a real big platform with big datasets.
3- An application to real epigenetics data, investigating the correlation between DNA-methylation and aging. The results reveal that epigenetics landscape in general and DNA methylation in particular change slightly with age.

Sincerely Yours,