Penguins Search Optimisation Algorithm for Association Rules Mining

Youcef Gheraibia[a,b,*], Sohag Kabir[a], Abdelouahab Moussaoui[c], Youcef Djenouri[d], Peng Yeng Yin[e], Smaine Mazouzi[b]

[a] *Department of Computer Science, University of Hull, Hull HU6 7RX, UK*
[b] *Badji Mokhtar-Annaba University, P.O. Box 12, 23000 Annaba, Algeria*
[c] *Laboratoire de Recherche en Informatique Applique (LRIA), Computer Science Department, University of Stif, 19000, Algeria*
[d] *DTISI, CERIST Research Center, Algiers, Algeria*
[e] *Department of Information Management, National Chi Nan University, Taiwan*

Abstract

Association Rules Mining (ARM) is one of the most popular and well-known approaches for the decision-making process. All existing ARM algorithms generate a very large number of association rules with high overlapping. To deal with this issue, we propose a new ARM approach based on penguins search optimisation algorithm (Pe-ARM for short). Moreover, an efficient measure is incorporated into the main process to evaluate the amount of overlapping among the generated rules. Indeed, before the evaluation of the rules, each penguin calculates the amount of overlapping between its rule and all accepted rules. Then, the given rule is rejected without fitness evaluation if at least the amount of overlapping for this rule and one rule from the set of accepted rule is greater than the maximum accepted overlap. The proposed approach also ensures a good diversification over the whole solutions space thanks to the multi-groups behaviour of penguins. Furthermore, the search process is based on the oxygen reserve of each penguin, which permit to accelerate the search of a good solutions in a reasonable time and to save energy in the disagreeable regions. To demonstrate the effectiveness of the suggested approach, several experiments have been carried out on different data-sets and specifically on the biological ones. The results reveal that the

*Corresponding author. Tel.: +44-7479-560709 ;
Email address: Y.Gheraibia@hull.ac.uk (Youcef Gheraibia)

proposed approach outperforms the well known ARM algorithms in both execution time and solution quality.

Keywords: Association Rule Mining; Penguin Search Optimisation Algorithm; Overlap measure; Biological Data-set; ARM.

1. Introduction

Association Rules Mining is one of the most challenging and important tasks in data mining [1]. ARM problem was first introduced by Agrawal and Shafer in [2] and can be formalised as : Let $I = \{i_1, i_2, i_3....i_n\}$ be a set of items and $T = \{t_1, t_2, t_3....t_m\}$ be a set of transactions, an association rule is an implication of the form $X \rightarrow Y$ such as $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \phi$. The sets of items X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the given rule respectively. The ARM problem consists of extracting among the transactional database T, all pertinent rules respecting minimum support and minimum confidence constraints. Association rules are the core of many applications like data warehousing for indexing problem [3], information retrieval for request processing and educational data mining for improving education system [4]. Describing hidden patterns and different dependencies between the set of data has the potential to efficiently improve the problem solving process. In the literature, ARM algorithms are divided into two main categories: exact approaches and meta-heuristics based approaches. The exact approaches such as Apriori [5], FPGrowth [6], DIC [7], DHP [8] and Eclat [9] aim at extracting all possible association rules. However, these algorithms are high time and space consuming when dealing with very large databases with many items and transactions. To reduce the computation time of such algorithms, a second category, metaheuristics-based approaches have been proposed. Examples of such approaches include ARMGA [10] and G3PARM [11] for evolutionary algorithms, PSO-ARM [12], ACO_R [13] and BSO-ARM [14] for swarm intelligence. These algorithms have the potential to **give only one part of rules and not all relevant rules**. Therefore, two challenges can be derived, the computation time of exact algorithms by exploring all rules space in one hand and the quality of the extracted rules using meta-heuristics based approaches by taking into account the similar rules in the other hand.
The penguins search optimisation algorithm (PeSOA) [15] is a recent nature inspired approach based on the collaborative hunting strategies of penguins.

The penguins synchronise their dives to reduce the expenditure of energy in the hunting process. This strategy is used in the algorithm to help converging quickly towards an optimal solution. The diversification strategy of the penguins search algorithm allows penguins to explore the whole solution space efficiently. The search process of penguins is based on the reserve of oxygen which allows penguins to decide whether to accelerate or decelerate while exploring the search space and also to decide whether to search or not in a given region. This has motivated the use of the PeSOA for the association rules mining problem to facilitate efficient exploration of the solution space. We incorporate a new measure to the main process of the penguins search to compute the amount of overlapping between rules to help penguins search process to generate only the non redundant rule with low amount of overlapping among them.
In this paper, a new algorithm called Pe-Arm (penguins search optimisation algorithm for association rules mining) is proposed to find a set of consistent rules with superior quality and a low amount of overlapping among the rules. These set of rules must cover the maximum number of transaction in the database.
The remainder of the paper is organised as follows: In the next section, relevant works on association rule mining with diverse applications are presented. Then, the penguins search optimisation algorithm is introduced in Section 3, followed by the proposed Pe-ARM approach in section 4. The experimental results with both standard and biological data sets are reported in Section 5. Conclusion and future perspectives of the the present work are provided in Section 6.

2. Related Works

ARM approaches can be divided into two main categories, exact and meta-heuristic based methods. This section reports some existing ARM approaches from both categories.

2.1. ARM with exact methods

The well known ARM exact algorithms are Apriori [5], AIS (Agrawal, Imielinski, Swami) [1], Eclat [9] and FP-Growth [6]. The Apriori algorithm is the most used exact algorithm for association rule mining, starts by finding all item sets that satisfy minimum support (frequent item sets) also called large item sets. After that Apriori uses these frequent item sets to generate

association rules. AIS is the first proposed algorithm for mining association rules. The main drawbacks of this algorithm are that it requires multiple scanning of database, i.e., it is time consuming and requires more storage space. FP-growth uses FP-tree structure to compress the database, where a divide-and-conquer strategy is performed to decompose the mining tasks and the database as well. When dealing with large transactional database, these algorithms become high time and memory consuming. Thereafter, different approaches have been proposed to ameliorate the exact methods such as reducing the number of passes over the database, sampling the database, using parallelism, and adding constraints on the structure of rules.

2.2. ARM with meta-heuristics

Meta-heuristics have been used to reduce the run time of the existing ARM algorithms. Genetic algorithm is the first evolutionary algorithm used to solve the ARM problem, such as GENAR [16], GAR [17]. Those two methods used the standard version of the genetic algorithm with poor representation of the **individuals**. Afterward, several methods for improving genetic algorithm for association rule mining have been proposed to ameliorate genetic operators and the representation of **individuals**, such as ARMGA [10]. Two major differences between the classical GENAR and ARMGA are the mutation and crossover methods. Often a hybrid method is used to reduce cost, as well as to improve the original methods. A novel hybrid genetic based algorithm called PQGMA has been applied for association rules mining with the use of simulated annealing for the mutation and the crossover operations respectively. Quantum computing based method [18] uses an adaptive mutation rate, and provides a diversified population. G3APRM [11] is a novel generation on association rule mining algorithm with the use of genetic programming based on the Grammar Guided Genetic Programming to avoid invalid individuals found by genetic programming process. In [19], the authors reviewed all association rule mining algorithms based on genetic algorithm and its hybridisation.

Other methods based on swarm intelligence have been designed for ARM problem. Particle swarm optimisation algorithm has been applied for association rule mining [12], in this algorithm particle moves randomly on different neighbourhood and optimise the selection process with the best neighbour. The main difference between this method and the AGA is the use of neighbourhood points which outperforms the intensification search. Other swarm intelligence algorithms have also been applied for ARM problem. Firstly,

an Ant colony optimisation [20] was employed in health application to deal with the health insurance database. ARMBGSA [21] is an association rule mining algorithm based on Newton law of universal gravitation namely a gravitational emulation local search algorithm, which is a nature inspired optimisation algorithm. Each rule is modelled as a mass, all the masses attract each other using the law of motion. Each iteration is based on the previous one, and take only the k-heaviest masses in order to influence the new masses. The algorithm generates few rules because the search space is reduced at each iteration. Authors on [22] proposed a new hybrid algorithm called (HBSO-TS) for association rule mining based on hybrid method based on Bees Swarm Optimisation (BSO) and Tabu Search (TS). BSO is used to explore the search space so that it can cover most of its neighbours.

2.3. ARM applications for genomic

Algorithms for Association Rule Mining have been extensively developed in market basket analysis, further studies concerning biological data sets are already available. In [23], the authors apply association rule mining process for genomic. Ant based Association Rule Mining (Ant-ARM) is employed to discover classification of rules for one particular class only, each ant is used to construct one Classification Association Rule and change one item set at a time. An improved method for integrated analysis of gene expression has been proposed using additional data [24]. Temporal association rules have been used to represent dependencies between the different factors on gene regulatory network [25]. Recent works of biological data analysis based on association rules mining are reviewed in [26].

3. Penguins Search Optimisation Algorithm

Penguin search optimisation algorithm (PeSOA) is a new swarm based meta-heuristic algorithm which was proposed in [15]. The dietary behaviour of penguins may be explained by the economic reasoning: it comes to a profitable food search activity when the gain of energy is greater than the expenditure required to obtain this gain. Penguins, behaving along the line of foraging predators, must extract information about the time and cost to get food and the energy content of prey in order to choose the course for making their next dive. PeSOA is inspired by the penguin's hunting behaviour and it generally works as follows:

The population of penguins locates initial position **(solution space)**, then the population is divided into a set of groups, and each group is assigned to a region in the whole solution space. Each penguin then dives and swims under the water for hunting fish while consuming its oxygen reserve. Different forms of communication between penguins are occasionally **taken place** and the quantities of eaten fish increase. The process is repeated until the specified amount of fish is obtained or the maximum number of iterations is reached. The authors of [15] have shown that the PeSOA outruns genetic algorithms and particle swarm optimisation in obtaining better values for benchmark optimisation functions. After a number of dives, the penguin returns to surface and share with its group affiliates the position and quantity of the food found. So the local best of each group continuously improves as more members report the food sources. After an entire cycle of the intra-group communication of all the penguin groups, the penguins might migrate to other group's habitat according to the probability of nurture existence of each group in terms of the quantity of food found by all its members. The oxygen reserve depends on both the gain of the food source and the swimming duration the penguin endures. If the energy gain is positive, the longer the penguin stays under the water, the more quantities of food it catches and thus becomes healthier. Otherwise, the longer the swimming duration, the more oxygen the penguin consumes. Hence, the oxygen reserve is updated according to the amelioration of the objective function. The oxygen reserve increases if the new solution is better than the previous one, and the oxygen reserve decreases in the other case.

4. Pe-ARM: PeSOA for association rules mining

4.1. Encoding

There are two well known association **reference needed** rule presentation, binary encoding and integer encoding. In binary encoding, each solution is represented by a vector S of n elements where n is the number of items. The i^{th} element of a given solution S is set to 1 if the item i is in the rule and 0 otherwise. However, in integer encoding, the solution is represented by a vector S of $k+1$ elements where k is the size of the rule. The first element is the separator index between antecedent and consequent parts of the solution. For all others elements i in S, if $S[i] = j$ then the item j appears in the i^{th} position of the rule. In Pe-ARM, both representations are combined to make the penguins operations and the fitness computing process easier. Indeed,

three values $(0, 1, 2)$ are used for interpreting the presence of a given item in the rule. The value 0 means that the item is absent in the rule. The value 1 means that the item is present in the antecedent part of the rule. The value 2 means that the item participates in the consequent part of the rule. More formally, we have :

1. $S[i] = 0$ if the item i is not in the solution S.
2. $S[i] = 1$ if the item i belongs to the antecedent part of the solution S.
3. $S[i] = 2$ if the item i belongs to the consequent part of the solution S.

This representation allows to separate the antecedent part and the consequent part where each single position of a given solution has the full interpretable information. Moreover, such representation is flexible and helps us on the calculation of the overlap measure.
Example: Let's $I = \{i_1, i_2, ..., i_{10}\}$ be a set of items. The solution $S_1 = \{0, 0, 0, 1, 2, 1, 0, 0, 0, 0\}$ represents the rule $r_1 : i_4, i_6 \Rightarrow i_5$

4.2. Overlapping measure

The optimisation of ARM aims at maximising the average of the confidence and the support of the generated rules. Optimisation algorithm gives only a set of the pertinent rules having a high confidence and support values. However, the generated rules may be redundant or similar [27]. For dealing with this problem, we propose a new measure to evaluate the correlation between the generated rules, allowing to maximise the coverage of the target data, this new measure thus gives a set of consistent rules with minimum overlap.

Definition 1. Let I=$\{i_1, ..., i_m\}$ be a set of items; D=$\{r_1, ...r_n\}$ be a set of association rules which can be defined as follow :
A= $\{(X, Y)/X \subseteq I, Y \subseteq I \; and \; X \cap Y = \phi\}$
Let μ be a function that computes the dissimilarity between two rules:

$$\mu : \left| \begin{array}{ccc} \mathbf{D} * \mathbf{D} & \longrightarrow & R^+ \\ \mu(r_i, r_j) & \longmapsto & \sum_{k=0}^{|r_i|} \sigma_k(r_i, r_j) \end{array} \right.$$

$$\sigma_k(r_1, r_2) = \begin{cases} 0 & \text{If} \quad [(i_k \in X_{r_1}) \;\&\; (i_k \in X_{r_2})] \; Or \; [(i_k \in Y_{r_1}) \;\&\; (i_k \in Y_{r_2})] \\ \frac{1}{2} & \text{If} \quad [(i_k \in X_{r_1}) \;\&\; (i_k \in Y_{r_2})] \; Or \; [(i_k \in Y_{r_1}) \;\&\; (i_k \in X_{r_2})] \\ 1 & \text{If} \quad otherwise \end{cases}$$

While X_{r_1}, X_{r_2} are the X parts of r_1 and r_2 element (rule) respectively, and Y_{r_1}, Y_{r_2} are the Y parts of r_1 and r_2 element (rule) respectively.

Proposition 1: μ satisfies the usual conditions for a distance:

i) $\mu\ (r_1, r_2) \geq 0$ and $\mu(r_1, r_2) = 0$ if and only if $r_1 = r_2$

ii) $\mu\ (r_1, r_2) = \mu(r_2, r_1)$

iii) $\mu\ (r_1, r_3) \leq \mu(r_1, r_2) + \mu\ (r_2, r_3)$ for any $r_1, r_2, r_3 \in$ E*E.

Proof:

i) $\mu(r_1, r_2) = 0$ if and only if r_1, r_2 agree in all items and this happens if and only if $r_1\{\text{x}\} = r_2\{\text{x}\}$ and $r_1\{\text{y}\} = r_2\{\text{y}\}$.

ii) The number of items in which r_1 differs from r_2 is equal to the number of items in which r_2 differs from r_1, because the distance is equal to the sum of all items $|r_1| + |r_2|$.

iii) $\mu(r_1, r_2)$ is equal to the minimal number of items which change their position or to add necessary to get from r_1 to r_2. In its turn, $\mu(r_2, r_3)$ is equal to the minimal number of items changes position or to add necessary to get from r_2 to r_3.
So $\mu(r_1, r_2) + \mu(r_2, r_3)$ changes from r_1 to r_3. Hence
$\mu(r_1, r_2) + \mu(r_2, r_3) \geq \mu(r_1, r_3)$ which is the minimal number of items that change position or to add necessary to get from r_1 to r_3.

Example: Let $I = \{A, B, C, D, E, F\}$ be a set of items and (r_1,r_2) are two rules defined as follow:
r_1 : A,B,C →D
r_2 : C,D → E
$\mu(r_1, r_2) = 3.5$

4.3. The Fitness Function

The main goal of penguins search algorithm is to optimise the expenditure of energy (run time) and to improve the quality of generated rules. The generated rules have both individual and collective quality, the first quality represents the statistical measure (confidence and support) which is calculated only from the rule and the transactional database, whereas, the second

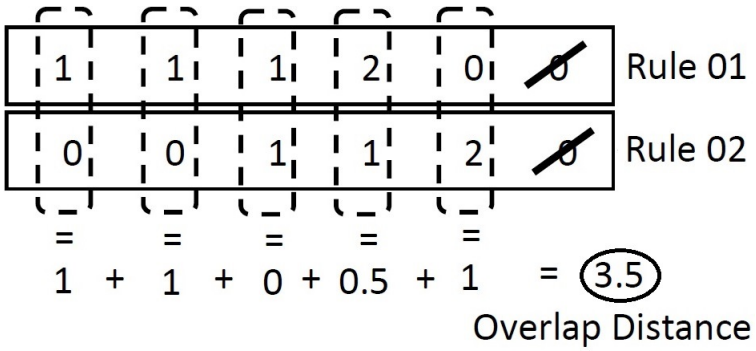

Figure 1: Measuring overlap distance

one aims to represent the correlation between the rules (overlap) which is well explained in the previous section. In the fitness computing, we focused on the first aspect by taking the rules which maximise the average of the support (Supp) and the confidence (Conf). The fitness value is computed only for the rules satisfying the maximum overlap where the maximum Overlap (Max-Overlap) is a predefined value that represents the maximum accepted distance between each pair of rules. More formally, the fitness function F for a given solution S can be formulated as:

$$F_{max}(S) = \alpha Supp(S) + \beta Conf(S) \tag{1}$$

where

$$Supp(S) = \frac{|\{t \in T | S[i] \neq 0 \Rightarrow S[i] \subseteq t, \forall i \in [1..n]\}|}{|T|}$$

$$Conf(S) = \frac{|\{t \in T | S[i] \neq 0 \Rightarrow S[i] \subseteq t, \forall i \in [1..n]\}|}{|\{t \in T | S[i] = 1 \Rightarrow S[i] \subseteq t, \forall i \in [1..n]\}|}$$

where α and β are two empirical parameters chosen between 0 and 1 and T is the transactional database.

Example:

Let us consider the transactional database (see Table 1) that contains 5 transactions T=$\{t_1, t_2, t_3, t_4, t_5\}$ and 5 items I=$\{A, B, C, D, E\}$. For instance, to compute the support and the confidence of the solution $S = (1, 2, 0, 0, 0)$ equivalent to the rule $(A \rightarrow B)$, the number of occurrences of the itemset (A) and the item set (A,B) should be first determined. We notice that (A)

t_1	A	B	C
t_2	A	B	
t_3	C	D	
t_4	E	D	
t_5	C	A	

Table 1: Illustration of transactional database for fitness computing

is repeated 3 times and (A,B) are repeated together 2 times. As a result, the support of (A) is 3/5 and the support of (A,B) is 2/5. So, the confidence of $(A \rightarrow B)$ is $\frac{2/5}{3/5}$ that equals to 2/3. Now, if we consider $\alpha = 0.5$ and $\beta = 0.5$, the fitness of S is : $F_{max}(S) = \left(\frac{1}{2} \times \frac{2}{5}\right) + \left(\frac{1}{2} \times \frac{2}{3}\right)$, which equal to $\frac{8}{15}$.

4.4. Algorithm of Pe-ARM

Pe-ARM algorithm (see Fig. 2) starts with generating a random population of penguins (each penguin represent a rule), this population is divided into groups, each group contains a variable number of penguins which is updated according to the penguins' health. The division of the initial population is based on the amount of overlapping between population rules. First a random penguin (P_r) is selected (center of the first group), and all penguins that have a distance (amount of overlapping) from P_r inferior than the min-distance are added to this group. The min-distance is equal to the average of distances between any two rules in the whole population. A new group is created if all other remaining penguins have a distance from P_r superior than the min-distance. A diversification generation strategy is used to generate K diversified groups in the initial penguin population. Pe-ARM starts with a population distributed in K groups, and each group is placed in a separate region with a minimum distance to any other. The purpose is to start the search with a set of diversified initial solutions which have contrasting features benefiting future solution improvement and to control the non visited region in the coming iterations. Our main goal is to generate a set of consistent rules that have a good fitness with small amount of overlapping between them. The objective function of a given solution $F(P_i)$ is to maximise the average of statistical measure (confidence and support).

Each penguin generates from it's rule another set of rules (neighbours). The best rule among these rules that optimises the objective function is selected. The penguin can move to another position and generate these neighbours if and only if its oxygen reserve O_i is not depleted. This oxygen

reserve is updated according to the objective function, it represents the health of the penguin. After each iteration, the fitness of the solution of the previous iteration is subtracted from the fitness of the new solution to obtain the value of the oxygen reserve. If the value of the subtraction is positive, the oxygen reserve is incremented to allow this penguin to move to other positions in the next iteration otherwise the oxygen reserve is decremented. The oxygen reserve controls the energy of the penguins in the whole search process. If the oxygen reserve is depleted (equal to zero) the penguins move to another location, either in an existing group or to a new unexplored area.

All generated rules are firstly validated with the overlap measure $\mu(P_i)$ before it is evaluated by the objective function. Any solution is validated (passed to the objective function evaluation) if it guarantee the minimum amount of overlap allowed with other accepted rules. The objective function evaluation is performed only for the valid rules because it is usually a high time consuming task for any meta-heuristics based algorithms.

The number of neighbours changes from one penguin to another and it is updated according to the penguin's health (penguin's oxygen reserve). In each iteration, if the oxygen reserve is incremented the number of neighbours is also incremented, in the other case the number of neighbours is decremented. The number of neighbours is initialised to '1', with such situation the penguin can generate only one new position by swapping between the possible values (0,1,2) for one item set. The amount of oxygen allows the penguins to decide to search or not in a given area and the number of neighbours allows penguins to decide to evaluate only a new position or a set of new positions.

After each iteration, the penguins communicate with each other the best rule, $GLbest$, founded to converge to the best group, update the oxygen reserve and the number of neighbours for each penguin. The computation of the health of each group is made to define the probabilities of improvement in each group, and finally we redistribute the penguins on the new population according the health of the penguins of each group (probabilities).

5. EXPERIMENTATION AND RESULTS

We used several evaluation criteria in the experimentation process in order to evaluate the proposed algorithm. Firstly, the statistical measure is computed which is represented as the average of the confidence and the support of the generated rules. Secondly, the run time of each approach is

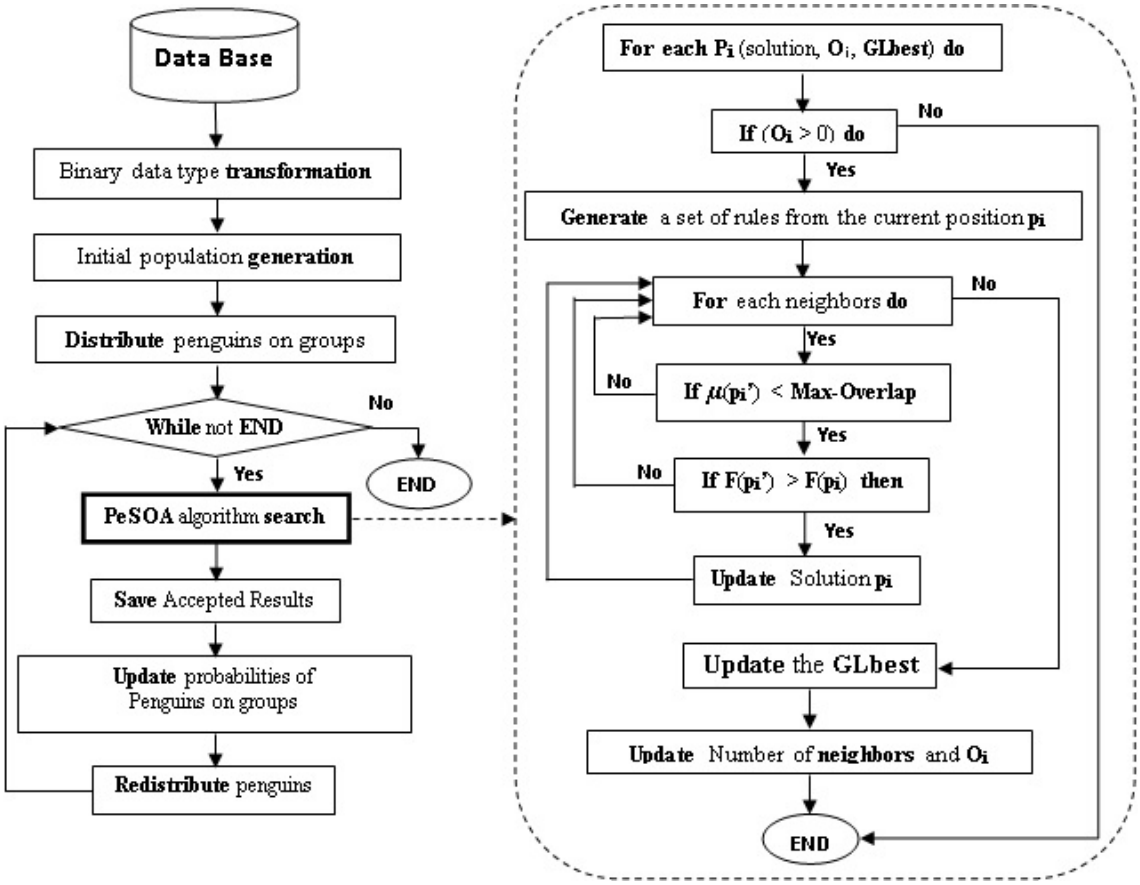

Figure 2: Penguins Search Optimisation Algorithm For Arm

determined.
The last measure is the coverage formula which represents the similarities between the generated rules according to the number of common transactions. Indeed, the rules are similar when they verify together many transactions and dissimilar where they do not verify any transaction [27]. The coverage formula is given as follows:
Let T_{r_i} be the set of transactions verified by r_i, and n is the number of generated rules.

$$Coverage = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \eta(r_i, r_j)}{n\ (n-1)}$$

$$\eta(r_i, r_j) = \begin{cases} 0 & i = j \\ \left|T_{r_i} \cup T_{r_j}\right| - \left|T_{r_i} \cap T_{r_j}\right| & \text{Otherwise} \end{cases}$$

5.1. Parameters Settings

The Penguin search algorithm needs many parameters to ensure the diversification and the intensification properties of the search process. The aim of this experiment is to find the best parameters to maximise the ratio between the average of between the fitness of generated rules and the CPU run time. In each iteration, we change the parameters in order to find the best stabilized averages values. Each new parameters are tested for a set of data sets. According to table 2, we see that when few number of penguins are used, the CPU run time is low, consequently, a small part of rules space is explored, this reduce the quality of the generated rules. Otherwise, when the number of penguins increases, we get a set of good rules but with increased CPU run time increases (We have used a 100 generation for all tests).
For the number of iterations(table 3), we aim to stabilize the average of the fitness solutions with the execution time. For few numbers of iterations we get a small execution time but also with small average of confidence and support, because of the few numbers of generated rules. while with a greater number of iterations we get a large average but also with longer execution time. According to the obtained results, the number of iterations is set to 100 and the number of penguins is set to 25.

Table 2: The performance of the PeSOA number of iterations with CPU execution time for different data sets.

Data sets	Bolts		Sleep		Pollution		Basket-Ball		IBM-Quest		Quack		Chess		Mushroom	
N Penguins	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**
10	0.81	0.03	0.88	0.24	0.79	0.21	0.91	0.14	0.81	0.19	0.81	0.54	0.80	0.61	0.78	1.21
15	0.85	0.05	0.93	0.31	0.91	0.28	0.95	0.20	0.86	0.21	0.89	0.61	0.83	0.70	0.79	1.41
20	0.98	0.08	0.97	0.39	0.97	0.35	0.98	0.29	0.90	0.25	0.90	0.69	0.87	0.76	0.81	1.58
25	0.99	0.10	0.99	0.46	1	0.37	0.99	0.31	0.92	0.28	0.90	0.76	0.88	0.80	0.84	1.67
30	1	0.13	0.99	0.52	1	0.44	0.99	0.37	0.92	0.35	0.90	0.82	0.88	0.87	0.84	1.72
35	1	0.18	0.99	0.57	1	0.49	0.99	0.44	0.92	0.39	0.89	0.87	0.88	0.93	0.84	1.79
40	1	0.21	0.97	0.62	1	0.56	0.98	0.46	0.91	0.42	0.90	0.93	0.87	0.99	0.84	1.83
50	0.99	0.25	0.99	0.75	1	0.62	0.98	0.53	0.92	0.56	0.90	1.10	0.88	1.15	0.82	1.99
60	1	0.30	0.98	0.82	1	0.81	0.99	0.61	0.92	0.63	0.90	1.25	0.88	1.32	0.84	2.18
75	1	0.41	0.99	0.91	1	0.95	0.99	0.82	0.92	0.75	0.90	1.39	0.88	1.49	0.84	2.32
100	1	0.62	0.99	1.14	1	1.24	0.99	1.02	0.92	0.92	0.90	1.52	0.88	1.81	0.84	2.61

Table 3: The performance of the PeSOA number of penguins with CPU execution time for different data sets.

Data sets	Bolts		Sleep		Pollution		Basket-Ball		IBM-Quest		Quack		Chess		Mushroom	
N Iteration	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**	**F**	**t**
50	0.94	0.04	0.81	0.21	0.91	0.19	0.89	0.17	0.79	0.15	0.82	0.44	0.75	0.52	0.77	0.85
75	0.98	0.07	0.92	0.29	0.95	0.25	0.92	0.21	0.85	0.19	0.88	0.54	0.80	0.61	0.79	1.19
100	1	0.11	0.98	0.41	0.99	0.28	0.96	0.26	0.89	0.23	0.91	0.69	0.82	0.68	0.85	1.44
125	1	0.13	0.99	0.45	1	0.34	0.99	0.30	0.92	0.27	0.90	0.75	0.88	0.77	0.84	1.59
150	1	0.15	1	0.48	1	0.37	1	0.32	0.91	0.30	0.90	0.81	0.89	0.82	0.88	1.64
175	1	0.18	1	0.50	1	0.41	1	0.36	0.92	0.32	0.91	0.92	0.89	0.88	0.87	1.71
200	1	2.1	1	0.56	0.99	0.43	0.99	0.39	0.92	0.36	0.91	1.05	0.88	0.95	0.88	1.79
225	1	2.5	1	0.57	0.99	0.46	1	0.44	0.91	0.38	0.92	1.17	0.89	1.00	0.85	1.85
250	1	2.7	1	0.62	1	0.50	1	0.49	0.92	0.43	0.90	1.26	0.89	1.14	0.84	1.93
275	1	3.0	1	0.64	0.99	0.54	1	0.51	0.92	0.45	0.91	1.32	0.87	1.21	0.86	2.05
300	0.99	3.3	1	0.68	0.99	0.59	1	0.53	0.92	0.49	0.90	1.56	0.88	1.32	0.88	2.21
325	1	3.8	0.98	0.70	1	0.62	0.99	0.56	0.92	0.52	0.91	1.62	0.89	1.38	0.87	2.33
350	1	4.2	0.99	0.71	0.98	0.66	1	0.59	0.92	0.53	0.91	1.75	0.89	1.42	0.88	2.43
375	1	4.5	1	0.73	1	0.68	0.99	0.63	0.91	0.54	0.91	1.86	0.88	1.49	0.89	2.49
400	0.99	4.7	1	0.72	1	0.73	0.99	0.67	0.92	0.61	0.92	1.96	0.89	1.55	0.87	2.58
425	1	5.1	1	0.77	1	0.75	1	0.69	0.91	0.63	0.90	2.10	0.78	1.61	0.87	2.63
450	1	5.5	0.99	0.79	1	0.78	1	0.73	0.92	0.65	0.92	2.23	0.89	1.64	0.88	2.71
475	1	5.4	1	0.81	1	0.82	1	0.75	0.92	0.69	0.91	2.32	0.88	1.70	0.87	2.88

5.2. Evaluation with standard data sets

The following datasets were prepared by [28] from the UCI datasets and PUMSB, though they have been converted to Apriori binary format. It has been widely used in the evaluating and comparing process for association rule mining problem, these data sets can be classified on three categories (small, medium and large data sets) [29]. Table 4 describes the size of the used datasets, the number of transactions and the number of items in each of the transactions.

Table 4: Standard data sets description

Dataset	**Transactions size**	**Items size**
Bolts	40	8
Sleep	56	8
Pollution	60	16
Basket-Ball	96	5
IBM-Quest	1000	40
Quack	2178	4
Chess	3196	75
Mushroom	8124	119

Table 5: Comparison of Pe-ARM different approaches with confidence and support average

Data sets	PE-ARM	BSO-ARM	ACO_R	SA	G3APARM	ARMBGSA
Bolts	1	1	0.69	0.60	0.92	0.45
Sleep	1	1	0.67	0.53	0.90	0.39
Pollution	1	1	0.66	0.50	0.92	0.56
Basket-Ball	1	1	0.61	0.66	0.93	0.45
IBM-Quest	0.92	0.89	0.45	0.30	0.88	0.40
Quack	0.91	0.89	0.73	0.52	0.90	0.39
Chess	0.89	0.86	0.3	0.15	0.86	0.38
Mushroom	0.88	0.84	0.1	0.05	0.85	0.35

Table 6: Comparison of Pe-ARM to different approaches with the run time (second)

Data sets	PE-ARM	BSO-ARM	ACO_R	SA	G3APARM	ARMBGSA
Bolts	0.12	0.22	1.23	1.04	0.59	1.17
Sleep	0.48	0.95	2.1	2.4	0.87	1.38
Pollution	0.35	0.62	1.1	1.6	0.67	1.87
Basket-Ball	0.28	0.56	1.3	1.5	0.42	2.10
IBM-Quest	0.251	0.32	0.9	1.9	0.87	1.45
Quack	0.67	0.75	1.5	2.4	1.00	1.94
Chess	0.725	0.85	2.4	3.02	0.99	2.41
Mushroom	1.474	1.5	3.6	2.8	1.84	3.98

Table 7: Comparison of Pe-ARM to different approaches with the coverage of 100 generated rules

Data sets	PE-ARM	BSO-ARM	ACO_R	SA	G3APARM	ARMBGSA
Bolts	11	7	5	4.12	8.24	6.25
Sleep	11.23	7.5	5	5.65	6.01	5.9
Pollution	11.25	4.58	6.24	5.32	5.98	6.02
Basket-Ball	12.65	6.9	4.10	7.01	6.8	5
IBM-Quest	15.2	6.14	6.98	4.28	6.87	7.82
Quack	21.51	10.25	9.24	8.24	11.08	10.01
Chess	29.41	9.27	8.21	10.36	8.52	9.88
Mushroom	35.25	12.34	10.01	9.85	12.38	12.01

We have compared PeARM with a set of well known association rule mining algorithms(BSO-ARM [22] , ACO_R [13], SA [30], G3APARM [11], ARMBGSA [21]). Table 5,6 and Table 7 summarize all the obtained results by using Pe-ARM with the various standard datasets. The aim is to maximise the average of the statistical measures (confidence and support) and to maximise the overlap between rules in order to maximise the coverage. The new Pe-ARM with the new overlap measure gives the best coverage values relative to other used algorithms, because of the set of association rules generated by the Pe-Arm (with the overlap measure) have low overlap between them. The mechanism of the penguins algorithm, ensures a good intensification technique on the way to ameliorate the execution time. In experimentation of standard data sets cases we have used 0.25, 0.30, 0.75 for minimum support, minimum confidence and maximum overlap accepted, respectively during performing experiments on standard datasets.

5.3. Evaluation with Biological data sets

One of the useful applications of association rules mining is bio-informatics [31]. In this section we have used several biological data sets for gene expression under a (sub)set of conditions [32, 29]. In the context of market basket analysis, gene expression data can be used as a single transaction, and each condition as an item. Also each condition can be validate or not in a given transaction (gene expression). Since the gene expression data belong to continuous real values, a discretization preprocessing for the gene expression data is needed [33]. Data sets values are discretized into two values, 0 if the condition are less-than or equal 0; and 1 otherwise. Table 8 presents the description of the gene expression datasets. The main motivation behind using biological datasets for evaluation is the specificity of gene expression data sets that contains comprehensive and integrated biological information, enabling the discovery of functional relationships between those information. Tables 9,10 and 11 show that the coverage of the final set of association rules are very large comparing to other standard datasets according to the strong relation between biological data, also the coverage of Pe-ARM (with overlap measure) is the best comparing with Pe-ARM (without overlap measure) and BSO-ARM, and this large coverage can be interpreted with very low correlation between rules that allow to take all transaction and items on consideration. The confidence and support average of the Pe-ARM has been ameliorated since the use of the Pe-ARM. In experimentation of biological data sets cases we have used 0.20, 0.30, 0.50 for minimum support, minimum

Table 8: Biological data sets description

Dataset	Transactions size	Items size
Leukemia	12457	72
Arabidopsis Thaliana	73	69
Saccharomyces Cerevisiae	190	170
Yeast	94	174
Alpha Factor	911	17
Cdc15	607	607
Elutriation	5632	15

Table 9: Comparison of Pe-ARM different approaches with confidence and support average

Data sets	PE-ARM	BSO-ARM	ACO_R	SA	G3APARM	ARMBGSA
Leukemia	0.78	0.68	0.45	0.41	0.65	0.51
Arabidopsis Thaliana	0.64	0.51	0.41	0.48	0.54	0.39
Saccharomyces Cerevisiae	0.59	0.38	0.34	0.31	0.37	0.35
Yeast	0.55	0.38	0.37	0.37	0.40	0.41
Alpha factor	0.64	0.46	0.43	0.39	0.44	0.50
Cdc15	0.62	0.43	0.43	0.48	0.43	0.39
Elutriation	0.69	0.43	0.50	0.40	0.41	0.38

Table 10: Comparison of Pe-ARM to different approaches with the run time (second)

Data sets	PE-ARM	BSO-ARM	ACO_R	SA	G3APARM	ARMBGSA
Leukemia	1.8	2.4	3.8	3.1	2.61	3.01
Arabidopsis Thaliana	0.867	1.875	2.1	1.9	1.12	2.10
Saccharomyces Cerevisiae	1.283	3.025	3.85	2.08	2.02	1.54
Yeast	0.571	1.457	1.98	0.86	1.42	1.04
Alpha factor	0.822	1.958	2.35	2.87	2.01	0.99
Cdc15	0.704	2.656	2.99	1.05	2.31	1.11
Elutriation	1.339	2.38	3.01	2.81	2.15	2.51

Table 11: Comparison of Pe-ARM to different approaches with the coverage of 100 rules

Data sets	PE-ARM	BSO-ARM	ACO_R	SA	G3APARM	ARMBGSA
Leukemia	1273.82	645.56	512.01	496.58	715.54	544.60
Arabidopsis Thaliana	18.46	7.45	5.14	5.27	7.14	4.98
Saccharomyces Cerevisiae	1269.02	925.46	536.27	821.14	898.74	768.24
Yeast	224.19	100.04	84.21	124.25	124.25	88.00
Alpha factor	461.41	221.15	102.46	98.57	184.54	113.56
Cdc15	620.62	252.98	232.97	201.62	412.32	199.85
Elutriation	1545.62	915.75	814.65	901.54	754.62	865.25

confidence and minimum overlap accepted, respectively during performing experiments on standard datasets.

6. Conclusion

In this work a new association rules mining algorithm based on penguins search optimisation algorithm (Pe-Arm) has been proposed. The incorporation of a new overlapping measure into the main method of Pe-Arm to evaluate the amount of overlapping between generated rules in order to generate a set of consistent rules. Pe-Arm has been compared with a set of well known meta-heuristics for association rule mining, by different criteria such as the computational time, the statistical measure (confidence and support) and finally with the coverage measure, which proved the efficiency of the proposed Pe-ARM. Different data sets have been used in the comparison, firstly is the standard data is used to evaluate the approach with those datasets that have been used by most of the association rule mining algorithm. Secondly, with a biological datasets, the use of biological data sets allows us to validate the approach with data that contain a huge number of association between item sets which is the specificity of the gene expression data. The gene expression datasets is made for that raison like extracting rules between conditions of genes. Currently, we are investigating with the improvement of the rules representation to develop new way to represent large datasets because we lose a lot of time on measure calculation.

References

[1] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, in: ACM SIGMOD Record, Vol. 22, ACM, 1993, pp. 207–216.

[2] R. Agrawal, J. C. Shafer, Parallel mining of association rules, IEEE Transactions on knowledge and Data Engineering 8 (6) (1996) 962–969.

[3] Z. Chen, Intelligent Data Warehousing: From data preparation to data mining, CRC press, 2001.

[4] C. Romero, S. Ventura, E. García, Data mining in course management systems: Moodle case study and tutorial, Computers & Education 51 (1) (2008) 368–384.

[5] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th int. conf. very large data bases, VLDB, Vol. 1215, 1994, pp. 487–499.

[6] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: ACM SIGMOD Record, Vol. 29, ACM, 2000, pp. 1–12.

[7] S. Brin, R. Motwani, J. D. Ullman, S. Tsur, Dynamic itemset counting and implication rules for market basket data, in: ACM SIGMOD Record, Vol. 26, ACM, 1997, pp. 255–264.

[8] J. S. Park, M.-S. Chen, P. S. Yu, An effective hash-based algorithm for mining association rules, Vol. 24, ACM, 1995.

[9] M. J. Zaki, Scalable algorithms for association mining, Knowledge and Data Engineering, IEEE Transactions on 12 (3) (2000) 372–390.

[10] X. Yan, C. Zhang, S. Zhang, Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support, Expert Systems with Applications 36 (2) (2009) 3066–3076.

[11] J. L. Olmo, J. M. Luna, J. R. Romero, S. Ventura, Association rule mining using a multi-objective grammar-based ant programming algorithm, in: Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on, IEEE, 2011, pp. 971–977.

[12] R. J. Kuo, C. M. Chao, Y. Chiu, Application of particle swarm optimization to association rule mining, Applied Soft Computing 11 (1) (2011) 326–336.

[13] P. Moslehi, B. M. Bidgoli, M. Nasiri, A. Salajegheh, Multi-objective numeric association rules mining via ant colony optimization for continuous domains without specifying minimum support and minimum confidence, International Journal of Computer Science Issues (IJCSI) 8 (5).

[14] Y. Djenouri, H. Drias, Z. Habbas, Bees swarm optimisation using multiple strategies for association rule mining, International Journal of Bio-Inspired Computation 6 (4) (2014) 239–249.

[15] Y. Gheraibia, A. Moussaoui, Penguins search optimization algorithm (pesoa), in: Recent Trends in Applied Artificial Intelligence, Springer, 2013, pp. 222–231.

[16] J. Mata, J. Alvarez, J. Riquelme, Mining numeric association rules with genetic algorithms, in: Artificial Neural Nets and Genetic Algorithms, Springer, 2001, pp. 264–267.

[17] J. Mata, J.-L. Alvarez, J.-C. Riquelme, An evolutionary algorithm to discover numeric association rules, in: Proceedings of the 2002 ACM symposium on Applied computing, ACM, 2002, pp. 590–594.

[18] M. Ykhlef, A quantum swarm evolutionary algorithm for mining association rules in large databases, Journal of King Saud University-Computer and Information Sciences 23 (1) (2011) 1–6.

[19] M. Zhang, C. He, Survey on association rules mining algorithms, in: Advancing Computing, Communication, Control and Management, Springer, 2010, pp. 111–118.

[20] K. Thangavel, P. Jaganathan, Rule mining algorithm with a new ant colony optimization algorithm, in: Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on, Vol. 2, IEEE, 2007, pp. 135–140.

[21] F. Khademolghorani, A. Baraani, K. Zamanifar, Efficient mining of association rules based on gravitational search algorithm, International Journal of Computer Science Issues (IJCSI) 8 (4).

[22] Y. Djenouri, H. Drias, A. Chemchem, A hybrid bees swarm optimization and tabu search algorithm for association rule mining, in: Nature and Biologically Inspired Computing (NaBIC), 2013 World Congress on, IEEE, 2013, pp. 120–125.

[23] M. Anandhavalli, M. Ghose, M. Gauthaman, Association rule mining in genomics, International journal of computer Theory and engineering 2 (5).

[24] P. Carmona-Saez, M. Chagoyen, A. Rodriguez, O. Trelles, J. M. Carazo, A. Pascual-Montano, Integrated analysis of gene expression by association rules discovery, BMC bioinformatics 7 (1) (2006) 54.

[25] E. Baralis, G. Bruno, E. Ficarra, Temporal association rules for gene regulatory networks, in: Intelligent Systems, 2008. IS'08. 4th International IEEE Conference, Vol. 2, IEEE, 2008, pp. 12–2.

[26] G. Atluri, R. Gupta, G. Fang, G. Pandey, M. Steinbach, V. Kumar, Association analysis techniques for bioinformatics problems, in: Bioinformatics and Computational Biology, Springer, 2009, pp. 1–13.

[27] Y. Djenouri, Y. Gheraibia, M. Mehdi, A. Bendjoudi, N. Nouali-Taboudjemat, An efficient measure for evaluating association rules, in: Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of, IEEE, 2014, pp. 406–410.

[28] K. Bache, M. Lichman, Uci machine learning repository. university of california, school of information and computer science, irvine, ca (2013) (2009).

[29] S. Ozel, H. Guvenir, An algorithm for mining association rules using perfect hashing and database pruning, in: 10th Turkish Symposium on Artificial Intelligence and Neural Networks, Citeseer, 2001, pp. 257–264.

[30] S. Naulaerts, P. Meysman, W. Bittremieux, T. N. Vu, W. V. Berghe, B. Goethals, K. Laukens, A primer to frequent itemset mining for bioinformatics, Briefings in bioinformatics (2013) bbt074.

[31] A. Gyenesei, U. Wagner, S. Barkow-Oesterreicher, E. Stolte, R. Schlapbach, Mining co-regulated gene profiles for the detection of functional

associations in gene expression data, Bioinformatics 23 (15) (2007) 1927–1935.

[32] D. Liu, Improved genetic algorithm based on simulated annealing and quantum computing strategy for mining association rules, Journal of Software 5 (11) (2010) 1243–1249.

[33] G. Li, Q. Ma, H. Tang, A. H. Paterson, Y. Xu, Qubic: a qualitative biclustering algorithm for analyses of gene expression data, Nucleic acids research 37 (15) (2009) e101–e101.