Kafrelsheikh Journal of Information Sciences ISSN (Online): 2535-1478, ISSN (Print): 2537-0677 Volume 4, Issue 1, 2023, PP. 1–16

# A hybrid of Information gain and Coati Optimization Algorithm for gene selection in microarray gene expression data classification

Sarah Osama[1], Abdelmgeid A. Ali[1], and Hassan Shaban[1]
1 Computer Science Department, Faculty of Computers and Information, Minia University, Minia, Egypt

**Abstract**. *Gene expression data has become an essential tool for cancer classification because it provides substantial insights into the underlying mechanisms of cancer progression. However, the high-dimensional nature of microarray gene expression data presents a significant challenge. This paper introduces a new method called IG-COA, which combines Information Gain (IG) approach and Coati Optimization Algorithm (COA), to identify the biomarkers genes. COA is a recent algorithm that has not been previously examined for feature or gene selection, to the best of our knowledge. Firstly, the IG method is used because using COA directly on microarray datasets is ineffective and can make it challenging to train a classifier accurately. Secondly, the COA algorithm is utilized to select the optimal subset of genes from the previously selected ones. The effectiveness of the suggested IG-COA method with a Support Vector Machine is tested on several microarray gene expression datasets, and it exceeds other state-of-the-art methods.*

***Keywords***: *Gene selection; Feature Selection; Microarray Gene Expression; Coati Optimization Algorithm; Cancer Classification.*

## 1 Introduction

In recent decades, bioinformatics has emerged as a prominent research field, facilitating the utilization of computer science and computational statistics technologies to comprehensively analyze an organism's genomic, transcriptomic, and proteomic data [1]. In biomedicine, disease prediction using microarray gene expression data is considered a crucial task [2]. A limited number of features effectively distinguish samples from distinct classes, while many others are deemed irrelevant, redundant, or generate noise. Moreover, it is worth mentioning that irrelevant features contribute to higher dataset dimensionality and greater computational complexity in clustering and classification tasks. Consequently, this can result in lower performance for machine learning (ML) algorithms [3]. Decreasing the dimension count in microarray datasets to a more manageable number can solve this problem [4]. Feature (gene) selection is a critical technique applied in different fields of study, such as bioinformatics, data mining, ML, and pattern recognition. Gene selection primarily aims to simplify data representation by decreasing the count of features or variables that describe it without losing essential information [4,5]. This technique is beneficial for numerous reasons, such as improving computational efficiency, enhancing model accuracy, and facilitating data visualization. Selecting the most valuable genes can also minimize the risk of overfitting, which happens when a model becomes too complex and fits the training data too closely, resulting in weak performance on unseen data [4]. Gene selection algorithms can be categorized into filter, embedded, wrapper, ensemble, and hybrid methods. On the one hand, filter methods use statistical measures to rank the variables according to their relevance to the target variable [6]. One of the main advantages of filter methods is their simplicity and ease of implementation, making them particularly suitable for large datasets with numerous features. Moreover, these methods are fast and do not require a learning algorithm. However, filter methods have limitations, such as limited accuracy due to their reliance on statistical properties of the data, inability to capture interactions between genes, and sensitivity to irrelevant genes in the dataset. Consequently, while filter methods can quickly minimize the count of genes in a dataset, they should be used cautiously and in combination with other gene selection approaches to obtain optimal results [4]. On the other hand, wrapper methods assess an ML model's performance using subsets of features. They search for the optimal gene subset that maximizes the model's performance [7]. These approaches can be

computationally expensive, but they are generally more accurate than filter methods [4]. Several wrapper methods rely on heuristic search algorithms to remove irrelevant genes. These methods initiate the process with a solution produced randomly, and in each iteration, they step closer towards the optimal subset of genes [3]. In the literature, it has been found that various meta-heuristic algorithms were employed as wrapper methods, such as Particle Swarm Optimization Algorithm [8,9,10,11], Genetic Algorithm (GA) [11,12,14], Ant Colony Optimization Algorithm [13,15,53], Simulated Annealing Algorithm [16,19], Binary Whale Optimization Algorithm [17], Binary Bat Algorithm [18], and many other algorithms. The key advantage of wrapper methods is the ability to assess the performance of an ML model using different subsets of features, allowing for a more comprehensive search across the feature space. This can result in improved accuracy and generalizability of the model. However, one major disadvantage of wrapper methods is their computational complexity, mainly when applied to high-dimensional datasets. Furthermore, because wrapper approaches pick a subset of genes that are dependent on a specific classifier, they may fail to identify important features that could be relevant to other classifiers. Moreover, wrapper methods' exhaustive search approach poses the risk of overfitting, resulting in reduced performance when tested on new data [3,4]. Therefore, while wrapper methods offer significant advantages, these limitations must be considered when selecting an appropriate feature selection methodology. On the contrary, embedded methods combine feature selection with the model-building process. The selection process is performed as part of training an ML algorithm [6]. A decision tree was used in [20], while the authors in [21] employed a backward feature selection process within the Multiple Criteria Linear Programming. Additionally, in [22], the authors developed a weighted gene selection approach embedded in the Bacterial Colony Optimization algorithm. One of the primary benefits of embedded methods is that they can integrate feature selection directly into the model training process rather than requiring a separate feature selection step. This can lead to faster and more efficient feature selection, as well as reducing the risk of overfitting [23]. However, one major drawback of embedded methods is that they are often limited by choice of learning algorithm because not all ML algorithms support this feature selection method [24]. Conversely, ensemble methods combine multiple feature selection methods to enhance the model's overall performance. The idea is to leverage the strengths of different feature selection methods and reduce their weaknesses by combining them. For example, an ensemble approach may combine a filter method with a wrapper or an embedded method to choose the most useful genes [4]. For example, the authors in [25] developed a cross-entropy-based multi-filter ensemble approach for gene selection. Furthermore, an ensemble of Chi-Square, ReliefF, Symmetrical, and Uncertainty filters was proposed in [26]. The authors in [27] developed an ensemble method that integrated G-Forest and GA. The main advantages of ensemble methods are that they can lead to more accurate and reliable performance with a reduced risk of overfitting. However, they can be computationally expensive [4]. In addition, they rely on the assumption that the individual algorithms being combined are themselves accurate and reliable. If one of the algorithms produces inaccurate results, it can negatively impact the overall performance of the ensemble. Hybrid methods, on the other hand, combine two or more feature selection methods to obtain a more compact and informative set of features. For example, the authors in [28] developed a two-stage feature selection model that integrated Minimum redundancy, maximum relevance ensemble, and GA. Poongodi and Sabari in [29] suggested a hybrid model that combined a parallelized mRMRe and GA. Furthermore, a hybrid gene selection method consisting of two stages was proposed in [30]. Firstly, an ensemble of Chi-square, Information Gain, and ReliefF was implemented. Secondly, the selected genes were input to Particle Swarm Optimization (PSO) to obtain the final subset of og genes. In a paper by Kundu et al. (2022) [31], a gene selection approach consisting of two phases was introduced. In the first phase, Pasi Luukka's feature ranking algorithm was employed to eliminate genes that were not relevant. In the second phase, an improved version of the whale optimization algorithm, called

the altruistic whale optimization algorithm, was utilized. This approach incorporated the concept of altruism into the whale population. In [32], the experts developed a novel hybrid approach. This approach has two stages: the first stage involves using one-class SVM for detecting anomalies, and the second stage involves developing a guided GA to identify the best subset of genes. Moreover, the authors in [33] developed a two-phase gene selection method. Firstly, an ensemble method that combined numerous filter-based methods, including the chi-square test, information gain ratio, and ReliefF, was developed. Secondly, a recursive flower pollination search algorithm is applied to identify the optimal subset of genes. Recently, hybrid gene selection methods have been widely developed due to their potential to enhance the accuracy and robustness of gene selection. Moreover, they require less computational time than wrapper methods and are more efficient than filter methods. Additionally, hybrid methods limit the risk of overfitting. However, there are also some disadvantages of hybrid gene selection methods. These methods increase the complexity of the developed gene selection method. Moreover, there is a risk of overfitting. They can also be impacted by merging different gene selection algorithms. Despite these challenges, hybrid gene selection methods offer promising avenues for improving the effectiveness of gene selection compared with other approaches. Finally, both ensemble and hybrid approaches can effectively improve the performance of ML models by selecting or generating a more informative set of genes. However, they can also be more complex and computationally expensive than individual gene selection methods.

### 1.1 Problem Statement

Let D be the input dataset containing gene expression values of size n × m, where n is the count of observations and m is the count of genes. Each entry in D is the expression level of a gene in an observation. Let *Y* be the vector of size n, where each element represents the class label of a sample, where $y_i$ takes an integer value from 1 to K, representing one of K possible cancer types. The target is to identify a minimum number of genes S from the set of all genes {1, 2, ..., p}, where p is the total count of genes, and build a classifier *f* based on this subset *S* that maximizes the prediction accuracy

$$\max Acc(S) = (1/n) * sum_{i=1}^{n}[y_i = f(x_i; S)] \quad (1)$$

$$\text{subject to } S \subseteq \{1, 2, ..., p\}, |S| \leq p,$$

on a test set. Formally, we want to solve the following optimization problem:

where *Acc(S)* is the accuracy of the classifier *f* built using the subset of genes *S*. *n* is the total count of observations in the test set, $y_i$ is the true label of the *i–th* observation, xi is the gene expression profile of the *i–th* observation, and *f(xi ; S)* is the predicted label of the i–th observation using the subset of genes *S*.

### 1.2 Motivation and Contributions

Motivated by the necessity of accurate identification of informative genes from high-dimensional microarray gene expression data for faster computation and improved classification performance, there is a critical need for advanced hybrid gene selection methods that can effectively obtain the most informative genes from microarray gene expression datasets. Since swarm optimization has successfully solved many optimization problems, it has been used extensively in the gene selection domain. The “No Free Lunch” theorem, however, states that there are no algorithms that are able to address all problems. As a result, this research paper investigates the latest swarm algorithms and their features to select one of them for implementing a novel two-stage gene selection method to tackle the curse of dimensionality of microarray gene expression data. This paper selects the COA algorithm, which has many unique updating mechanisms designed to mimic the actions of coatis observed in their natural habitat. At its core, COA aims to simulate two primary behaviors exhibited by coatis: their hunting strategy when pursuing and capturing iguanas and their ability to avoid potential threats from predators. Moreover, the algorithm’s performance was tested using 51 objective functions. These functions comprise 29 from the

IEEE CEC-2017 test suite and 22 real-world applications from the IEEE CEC-2011 test suite. The performance of COA was then compared to that of eleven other commonly applied meta-heuristics algorithms. Due to COA's versatility in exploration and exploitation strategies, the COA algorithm can solve various optimization problems. As a result, this paper develops a new hybrid gene selection algorithm that combines IG and COA to address the curse of dimensionality. The key contributions of this study are as follows.

- Gene selection is a growing area of biomedical data analysis that is crucial to improving the performance of ML algorithms. This paper aims to suggest a hybrid gene selection algorithm that is made up of two phases for the prediction of cancer based on gene expression.
- In the first phase, IG is used as filter method to eliminate some irrelevant genes. In the second phase, this paper exploits the ability of COA algorithm, which is a wrapper method, to address gene selection issue. To the best of our knowledge, this is the first use of COA for feature/gene selection.
- In this study, four swarm intelligence algorithms are integrated with IG to compare the developed IG-COA method, including both recent and former algorithms: Kepler Optimization Algorithm (KOA), Social Ski-driver Optimization (SSD), Whale Optimization Algorithm (WOA), and Artificial Bee Colony Algorithm (ABC).
- The IG-COA method is evaluated using six microarray gene expression datasets.
- The proposed algorithm is compared with many recent studies in this domain, which proposed between 2012 to 2023, and it performs better than exciting algorithms.

### 1.3 Paper Structure

This paper is organized as follows. Section 2 explains some of the basic concepts applied in the proposed methods. The used methodology is explained in Section 3. Finally, Section 4 discusses the experimental results, and Section 5 provides the conclusion of this paper.

## 2 Background

This section presents the necessary information required to understand the suggested method.

### 2.1 Coati Optimization Algorithm

A COA is a newly proposed meta-heuristics algorithm which simulates the characteristics of coatis found in nature. COA's basic essence is to replicate two key behaviors of coatis: their approach when attacking and hunting iguanas and their evasion from hunters [34]. Coatis, commonly called coatimundis, are omnivorous mammals that consume both invertebrate and small vertebrate prey. Notably, the green iguana constitutes a significant component of the coatis' diet. Due to their arboreal nature, coatis frequently forage for iguanas in trees and often hunt collectively. The coatis' hunting strategy may involve some group members climbing trees to startle the iguana into leaping to the ground while others rapidly attack it. However, despite their effective predation tactics, coatis are vulnerable to attacks from hunters and large raptors. The COA algorithm aims to simulate the coatis' behaviors. The following subsections summarize the mathematical model of COA algorithm.

### 2.2 Stage 1: initialization

Each coati's location in the search space corresponds to the decision variable values, and each location is a suggested solution to the optimization problem. The COA begins by randomly initializing the coatis' positions in the search space using Equation 2.

$$P_i : p_{i,j} = Pmin_j + random() \cdot (Pmax_j - Pmin_j), \quad i = 1, 2, ..., M, j = 1, 2, .., n, \quad (2)$$

where $P_i$ represents the placement of *ith* coati in search space; $p_{i,j}$ refers to the jth dimension. $Pmax_j$ and $Pmin_j$ define the upper and lower bounds of the *jth* dimension respectively, and *random()* generates a random real number

between 0 and 1. Equation 3 shows a population matrix of the coatis.

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_M \end{bmatrix}_{M\times n} = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,n} \\ P_{2,1} & P_{2,2} & \dots & P_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ P_{M,1} & P_{M,2} & \dots & P_{M,n} \end{bmatrix}_{M\times n} \quad (3)$$

Coatis use this matrix to update their positions. Each location is a possible solution, and it is evaluated by using an objective function as shown in Equation 4.

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_M \end{bmatrix}_{M\times 1} = \begin{bmatrix} F(P_1) \\ F(P_2) \\ \vdots \\ F(P_M) \end{bmatrix}_{M\times 1} \quad (4)$$

where *F* represents a vector of the computed objective function and $F_i$, where i= {1, 2, ..., M} is the value obtained using objective function according for the *ith* coati.

## 2.3 Stage 2-exploration: a strategy for hunting and attacking iguanas

This stage simulates a collection of coatis climb a tree with the intent to intimidate an iguana, while some remain on the ground awaiting the moment when the iguana falls. Once the iguana falls, the coatis attack and hunt it. This strategy allows the coatis to explore various locations in the search space, which showcases their ability to explore globally when solving problems. The COA algorithm assumes that the optimal coati of the population is located at the same location as the iguana. In addition, it is supposed that fifty percent of the coatis are ascending the tree, while the others are patiently waiting for the iguana to plummet to the earth. To mathematically simulate the position of the coatis climbing up the tree, Equation 5 is utilized.

$$P_i^{p1} : p_{i,j}^{p1} = p_{i,j} + random() \cdot (Iguana_j - I \cdot p_{i,j}), \quad for\ i = \{1, 2, ..., (M/2)\}, and\ j = \{1, 2, .., n\}, \quad (5)$$

Once the iguana falls to the ground, it is randomly placed in a new location within the search space. Coatis on the ground then move within this search space, which is simulated using Equations 6 and 7, based on the iguana's new random position.

$$Iguana^{\varphi} : iguana_j^{\varphi} = Pmin_j + random() \cdot (Pmax_j - Pmin_j), \quad j = \{1, 2, ..., m\}, \quad (6)$$

$$P_i^{p1} : p_{i,j}^{p1} = \begin{cases} p_{i,j} + random() \cdot (Iguana_j^{\varphi} - I \cdot p_{i,j}), if\ F_{Iguana^{\varphi}} < F \\ p_{i,j} + random() \cdot (p_{i,j} - Iguana_j^{\varphi}), otherwise \end{cases} \quad (7)$$
$$for\ i = \{[N/2] + 1, [N/2] + 2, ..., N\}\ and\ j = \{1, 2, ..., m\}$$

where *random()* generates a random real number between 0 and 1. The position of the best member, Iguana, in the search space is represented by the variable 'Iguana', with its *jth* dimension being $\text{Iguana}_j$ . An integer '*I*' is randomly picked from the set {1,2}. The location of the iguana on the ground is randomly computed and represented by $\text{Iguana}^{\varphi}$, with its *jth* dimension being $\text{Iguana}_j^{\varphi}$ and its objective function value being $F_{\text{Iguana}^{\varphi}}$. The floor function (also called greatest integer function) is denoted by [.].

If the new location of each coati leads to an improvement in the objective function value, then it is considered acceptable for the update process. However, if there is no improvement, the coati will remain in its previous location. This condition applies to all N coatis, which is simulated using Equation 8.

$$P_i = \begin{cases} p_{i,j}^{p1}, & if(F_i^{p1} < F_i), \\ P_i, & otherwise \end{cases}, \quad (8)$$

The *ith* coati's new location, $P_i^{p1}$, is computed using the value of its *jth* dimension, $P_{i,j}^{p1}$, and its objective function value, $F_{i,j}^{p1}$.

## 2.4 Phase 3-exploitation: a process of escaping from predators

This stage of updating coatis' location in the search space is designed to imitate coatis' natural behavior when facing hunters and fleeing from them. When a predator attacks a coati, it quickly moves away from its current location to a safer one. Coatis move in such a way as to end up in a secure location close to their current location, demonstrating their proficiency in finding local solutions. In order to imitate the coatis' behavior, a new location is created randomly near their present position using Equations 9 and 10.

$$P_i^{p2}: p_{i,j}^{p2} = p_{i,j} + (1 - 2random()) \cdot (Pmin_j^{local} + random() \cdot (Pmax_j^{local} - Pmin_j^{local})), \quad (9)$$

$$i = \{1, 2, ..., N\} \; and \; j = \{1, 2, ..., m\}$$

$$Pmin_j^{local} = Pmin_j/Iter, \; Pmax_j^{local} = Pmin_j/Iter, \; where \; Iter = \{1, 2, ..., MaxIter\} \quad (10)$$

If the value of the objective function enhances, which is represented by Equation 11, then the recently computed location is considered satisfactory.

$$P_i = \begin{cases} P_i^{p2}, & if(F_i^{p2} < F_i), \\ P_i, & otherwise \end{cases}, \quad (11)$$

where $P_i^{p2}$, represents the updated position of the *ith* coati computed during this stage of COA. Its *jth* dimension is denoted as $P_{i,j}^{p2}$, and its objective function value is represented by $F_{i,j}^{p2}$,. *random*() generates a random number within the range of 0 to 1 is involved in the calculation process, along with an iteration counter "*Iter*". , $Pmin_j^{local}$, and $Pmax_j^{local}$ represent the local lower and upper bounds of the jth decision variable, respectively. Similarly, $Pmin_j$ and $Pmax_j$ refer to the lower and upper bounds of the *jth* decision variable, respectively.

### 2.5 Repetition process

The completion of an iteration of COA occurs once all coatis in the search space have had their positions updated according to the the second and third stages. The population is then updated using Equations 5 through 11 and the process is repeated until the final iteration of the algorithm. At the end of the entire COA run, the output returned is the best solution obtained overall iterations. Figure 1 depicts the flowchart of COA algorithm. For more details, refer to the main paper that proposed COA algorithm [34].

## 3 Proposed IG-COA Gene Selection Method

### 3.1 Framework of IG-COA method

The IG-COA framework is shown in Fig. 2. As shown in this figure, the IG-COA method consists of three phases; the first phase is gene preprocessing, the second phase is gene selection which involves ranking all features in the dataset using IG. Afterwards, a population with *M* individuals is initialized based on the top *G* genes selected by IG. Then, the search process of COA is established to obtain a final optimal subset of genes. Finally, the third phase is the classification phase where the final optimal subset of genes is used for cancer classification. The following subsection explains the proposed method in detail.

### 3.2 Phase 1: Gene preprocessing

Data preprocessing is a crucial step in analyzing microarray gene expression datasets. The raw data obtained from microarray experiments often contain missing values, which can occur for various reasons, such as experimental error or technical limitations. That can affect the accuracy and reliability of gene expression analyses. Failure to preprocess the data adequately can result in erroneous analysis and misleading results. Therefore, proper data preprocessing is essential for obtaining a meaningful analysis of the gene expression patterns [4]. In the following lines, the used data preprocessing techniques are highlighted.

#### 3.2.1. Train-test Split

This process is a fundamental ML technique for evaluating a model's performance on unseen data. By splitting the data, we can train the model on one subset of the data and then evaluate its performance on another subset that the model has not seen before. The goal is to assess how well the model will generalize to new, unseen data. In this paper, stratified train-test splitting is used. This technique divides the dataset into two sets - one for training and the other for testing. It ensures that each set has a balanced number of instances for all classes. The split is done in an 80-20 ratio, with 80% of data reserved for training and 20% for testing.

#### 3.2.2. Imputation of Missing Genes

Some datasets have missing values. In this paper, the kNN algorithm imputes missing

values by replacing them with the average values of their closest neighbors in the training set. In this case, two instances are considered similar if their existing gene values are similar. If an instance does not have a class label, it is generally omitted instead of being imputed.

### 3.2.3. Data Normalization

The goal of data normalization is to transform a dataset into a standard format, typically to aid in comparing variables with different units, scales, or distributions. It involves adjusting the values of features in a dataset to have a common scale without distorting differences in the ranges of values or losing information. This can enhance the accuracy and performance of ML models [4]. This paper uses a min-max normalizer.

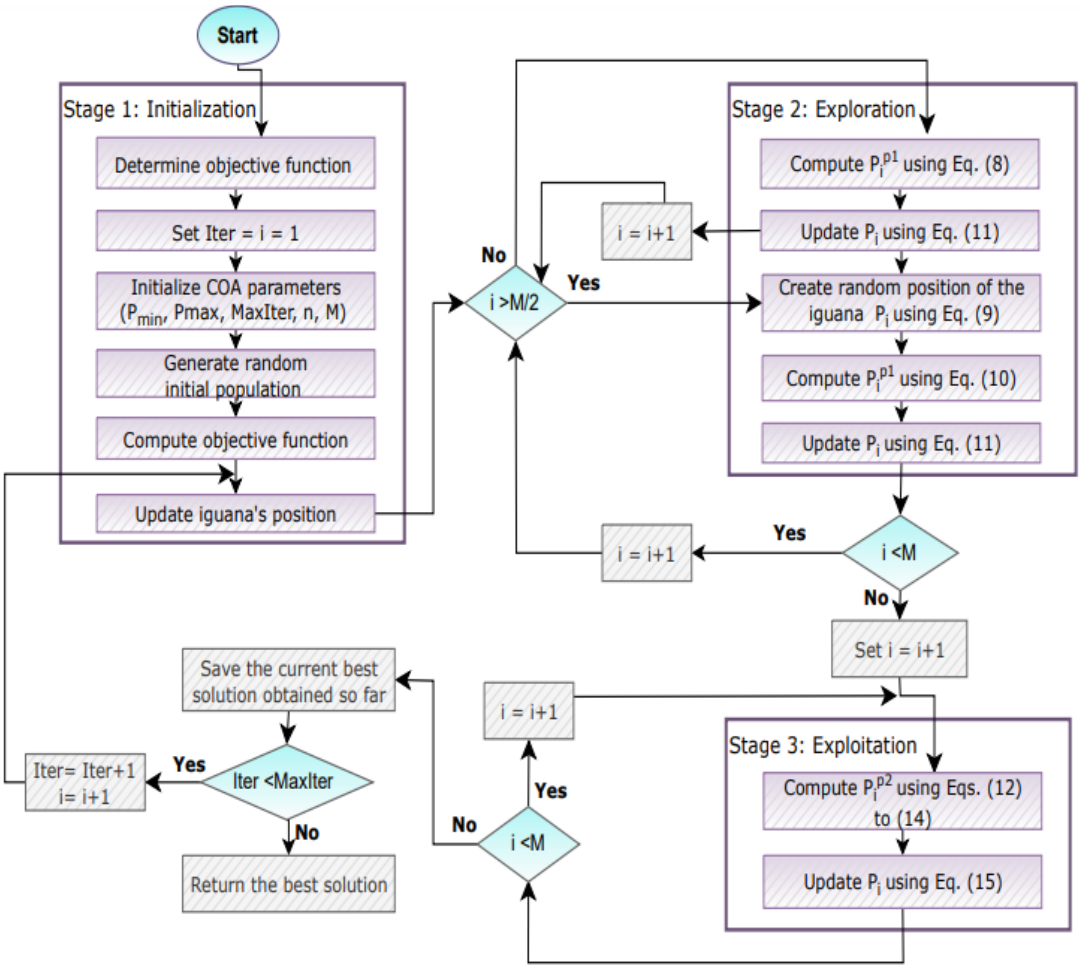


*Fig. 1: A flow chart of COA algorithm*

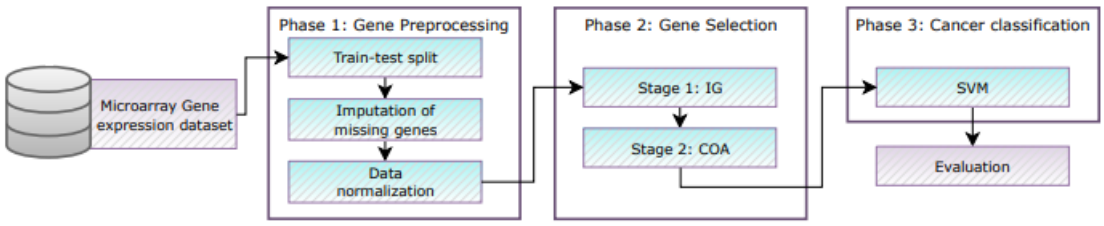


*Fig. 2: An outlined of the proposed IG-COA method*

## 3.3. Phase 2: Gene Selection

The suggested gene selection method is a hybrid method combining IG and COA. The main stages of the gene selection phase are shown in the following lines. Algorithm 1 shows the details of the proposed IG-COA method.

### 3.3.1. Stage 1

IG To speed up the selection process in the COA stage, the dimensionality of gene expression datasets is reduced by applying IG to select the most useful G genes (Line 1 in Algorithm 1). According to previous studies [35,36,37,30], a fixed subset of 100 genes is useful for filter methods. Afterwards, the dataset with picked genes is used as an input to the COA algorithm that is applied for further dimensionality reduction.

### 3.3.2. Stage 2: Representation of Candidate Solutions

COA is a continuous swarm algorithm that operates in continuous search spaces. It can handle a wide range of problems that can take on any real-valued number within specified bounds. However, converting this continuous solution to a discrete one is desirable in the gene selection case. Rounding is a standard method for converting a continuous solution to a discrete one. It involves rounding the continuous variables to the nearest integer value within a specified range and removing the repetitions from the new solution. Finally, these integer values represent solutions in the discrete search space. The position $loc_i'$ converts to discrete using Equation 12.

$$loc_i^{'} = f_u(f_r(loc_i)) \qquad (12)$$

where $loc_i'$ refers to the updated discrete positions at j iteration, and $f_u$ finds the unique positions. $f_r$ is used to round the value of each dimension in loci to the nearest integer less than or equal to this value.

### 3.3.3. Stage 3: Population Initialization

Initialize the parameters of COA, including Number of iterations (*MaxIter*), population size (M), lower bound (*Pmin*), upper bound(*Pmax*), dimensions (*m*). Then, create a random initial solution according to on *Pmin* and *Pmax* using Equation 2 (Line 4 in Algorithm 1).

### 3.3.4. Stage 4: Objective Function

The fitness function evaluates the effectiveness of each solution by measuring its ability to achieve the best performance. In gene selection, it is crucial to consider both high classification accuracy and a lower number of genes simultaneously. If many sets of genes lead to the same accuracy, the smallest set is picked. SVM is a widely and accurately used classifier in literature. Thus, in this study, the objective function is calculated using SVM, and its meta-parameters are fine-tuned using a grid search algorithm with k-fold cross-validation, where k=3. COA uses a fitness function that combines accuracy with the count of selected genes, calculated as the average of a k-fold cross-validation algorithm (where k=3). The obtained accuracy and corresponding gene count are compared with the best global solution and its accuracy. Based on these considerations, Algorithm 2 defines the objective function.

### 3.3.5. Stage 5: Optimization Loop

The COA optimization loop, described in Algorithm 1, starts at Line 7 and goes up to Line 32. Firstly, the algorithm's random parameters are initialized. Secondly, the candidate solutions are generated, then this solution is converted into a discreet solution (as shown in Section 3.3). Then, each candidate solution is assessed using the objective function (as shown in Section 3.3). Afterwards, the best solution is determined. After that, these solutions are updated, and then this new solution is converted into a discreet solution and evaluated. Finally, the optimization process is terminated when *MaxIter* is reached.

## 3.4. Phase 3: Cancer Classification

An SVM algorithm, introduced in Drucker et al.'s paper [54], is a highly efficient method

widely used for classifying gene expression data. According to our previous comprehensive review in the domain of this paper [4], we found that the accurate classifier used in the literature is an SVM, and it outperforms other classifiers used. Therefore, this paper uses an SVM as a classifier. Additionally, a grid search algorithm is used to tune the SVM meta-parameters, such as kernel type and kernel parameter, if exist. Before applying grid search, a technique called k-folds cross-validation where k=3 is applied to get meta-parameters for SVM. This involves dividing the training set into k subsets, then using k-1 subsets to train the model and the remaining subset for validation. This process is repeated k times, with each subset serving as the validation set precisely once. By averaging the results of these iterations, we obtain optimized parameters for our SVM model.

**Algorithm 1:** Pseudo code of IG-COA algorithm

**Input:** Dataset($D_{in}$), Number of genes $G$, Number of iterations ($MaxIter$), population size ($M$), lower bound ($Pmin$), upper bound($Pmax$), and number of dimensions (n)
**Output:** Best solution

Top genes ($TG$) ← Apply IG on $D_{in}$ to obtain top G genes
$D_{ig} \leftarrow D_{in}[:, TG]$
%Initialization (Section 2.2)
Randomly initialize the locations using Equation 2
Convert continues solution to discrete as shown in Section 3.3
Evaluate the objective function for this solution as shown in Section 3.3.
Select on of these solutions to be the best solution (i.e., position)
**repeat**
    Update the position of the iguana based on best solution.
    %Exploration: a strategy for hunting and attacking iguanas (Section 2.3)
    **repeat**
        Use Equation 5 to generate a new position for *ith* coati.
        Use Equation 8 to update the position of *ith* coati.
        Convert continues solution to discrete as shown in Section 3.3
        Evaluate the objective function for this solution as shown in Section 3.3.
        $i \leftarrow i + 1$
    **until** $(i < [M/2])$
    $i \leftarrow 1 + [M/2]$
    **repeat**
        Use Equation 6 to compute random location for iguana.
        Use Equation 7 to generate a new position for *ith* coati.
        Use Equation 8 to update the position of *ith* coati.
        Convert continues solution to discrete as shown in Section 3.3
        Evaluate the objective function for this solution as shown in Section 3.3.
        $i \leftarrow i + 1$
    **until** $(i < M)$
    %Exploitation: a process of escaping from predators (Section 2.4)
    Use Equation 9 to compute the local bounds for variables
    $i \leftarrow 1$
    **repeat**
        Use Equation 10 to compute new location for the *ith* iguana.
        Use Equation 11 to update the position of *ith* coati.
        Convert continues solution to discrete as shown in Section 3.3
        Evaluate the objective function for this solution as shown in Section 3.3.
        $i \leftarrow i + 1$
    **until** $(i < M)$
    Save the current best solution obtained so far
    $Iter \leftarrow Iter + 1$
**until** $(Iter < MaxIter)$
**Return** the best solution

**Algorithm 2:** Pseudo code of objective function evaluation

**Input:** Current position ($Curr_{pos}$), global best solution ($Gbest_{pos}$), corresponding fitness($Gbest_{fit}$), training dataset ($D_{tr}$), test Dataset ($D_{te}$)

**Output:** Fitness ($fit$)

Use grid search algorithm with 3-fold cross-validation to tune SVM meta-parameters,
Build a cancer classification model ($Model_{svm}$) by training SVM using $D_{tr}[:, Curr_{pos}]$.
Evaluate $Model_{svm}$ by calculating the accuracy $Ac$ using $D_{te}[:, Curr_{pos}]$.
% Update global solution and its corresponding fitness
**IF** ($Ac > Gbest_{fit}$) or ($Ac == Gbest_{fit}$ and len($Curr_{pos}$)<len($Gbest_{pos}$)) **do**
    $Gbest_{fit} \leftarrow Ac$
    $Gbest_{pos} \leftarrow Curr_{pos}$
**End If**

## 4. Experimental Analysis and Discussion

Firstly, this section highlights the used microarray gene expression datasets. Secondly, it assesses the effectiveness of the suggested IG-COA algorithm. In this section, IG-COA is compared with other swarm-based gene selection algorithms, conventional gene selection methods, and some of the state-of-the-art methods in the literature, including 2023 developed methods. 4.1 Datasets In this study, five commonly used microarray gene expression datasets are utilized. They include binary- and multi-class datasets. These detests are shown in Table 1.

Table 1: List of microarray gene expression datasets used in this paper and corresponding URLs.

| Code | Dataset | Disease | #Genes | #Observations | Type | #Classes | URL |
|---|---|---|---|---|---|---|---|
| DS 1 | Yeoh [38] | Leukemia | 12625 | 248 | Multi-class | 6 | $URL^1$ |
| DS 2 | SRBCT [39] | Small round blue cell tumors | 2308 | 83 | Multi-class | 4 | $URL^2$ |
| DS 3 | Ovarian [40] | Ovarian cancer | 15154 | 253 | Binary-class | 2 | $URL^2$ |
| DS 4 | Leukemia [41] | Leukemia cancer | 7129 | 72 | Binary-class | 2 | $URL^2$ |
| DS 5 | West [42] | Breast Cancer | 7129 | 49 | Binary-class | 2 | $URL^1$ |
| DS 6 | Central Nervous System (CNS) [43] | Tumours of the Central Nervous System | 7129 | 60 | Binary-class | 2 | $URL^2$ |

$URL^1$ : https://github.com/kivancguckiran/microarray-data
$URL^2$ : https://csse.szu.edu.cn/staff/zhuzx/Datasets.html

### 4.2. Experimental Setup

The proposed IG-COA method is implemented using Python language. Initially, we split the datasets into 80% for training and 20% for testing. We conducted all experiments on the same PC to ensure an unbiased comparison. In order to ensure fairness in the comparison, we ran all swarm algorithms twenty times independently with the same initial positions, the maximum number of iterations, the number of dimensions, and the number of search agents. These parameters are listed in Table 2. We utilize the grid search optimizer with k-fold cross-validation, where k = 3, to optimize the SVM's meta-parameters.

### 4.3. Evaluation Metrics

In this paper, the suggested method is evaluated based on several predefined criteria in order to measure its effectiveness at maximizing the desired outcome:

- Mean accuracy (i.e., fitness value) - measures the method's overall performance regarding maximizing the objective function. A higher mean accuracy indicates that the method performs better at achieving the desired outcome.
- The number of selected genes - evaluates the method's efficiency in selecting the most relevant genes for maximizing the objective function. A lower number of selected genes implies that the method is more effective at identifying essential genes

that can aid in achieving a better solution.

- Standard deviation - this parameter helps understand how consistent the results of the method are in maximizing the objective function. Suppose the mean value is high and the standard deviation value is low. In that case, we can infer that the method is reliable and produces consistent results in maximizing the objective function. By using these criteria, we can comprehensively evaluate the suggested method's ability to maximize the objective function, enabling us to determine its efficacy for the gene selection problem.

### 4.4. Result analysis and discussion

This study proposes a new gene selection method called IG-COA, which is compared with other swarm algorithms in Section 4.4 and widely used filter-based gene selection methods in Section 4.4. Finally, we evaluate the performance of our IG-COA method against state-of-the-art gene selection methods discussed in Section 4.4. Through this comprehensive evaluation, we aim to demonstrate that our proposed method outperforms existing approaches in terms of accuracy, number of selected genes, and stability.

#### 4.4.1. Comparison with other swarm optimization algorithms

In this study, the IG-COA algorithm has been thoroughly evaluated and compared with other state-of-the-art swarm algorithms, such as ABC, WOA, SSD, and KOA. This comparison aims to demonstrate its superiority regarding obtaining the highest mean fitness value (i.e., accuracy) with the minimum number of selected genes. To validate the effectiveness of our proposed method, we employ statistical metrics, including mean and stander deviation (Std.). Table 3 shows the mean accuracy, the number of selected genes, and Std. of the developed IG-COA in comparison with IG-ABC, IG-WOA, IG-SSD, and IG-KOA. It can be witnessed that the proposed IG-COA method exceeds others. The reported mean accuracy, number of selected genes, Std. are computed for the optimal solution obtain so fair over 20 independent executions. It can be witnessed that, over all datasets, the IG-COA method outperforms other methods with accepted Std., followed by IG-SSD. Additionally, IG-COA method attains the highest results for five datasets, DS 1, Ds 2, DS 3, DS 5, and DS 6, while IG-ABC outperforms others in DS 1. Moreover, while some methods obtain the same accuracy as the proposed IG-COA method (100%), the IG-COA selects the minimum number of genes. Fig. 3 compares the performance of the proposed IG-COA with others overall datasets. As shown in Fig. 3, IG-COA achieves the highest accuracy (99.25%) with the minimum number of selected genes (19.14), followed by IG-SSD, accounting for 98.68% accuracy, and 16.05 selected genes.

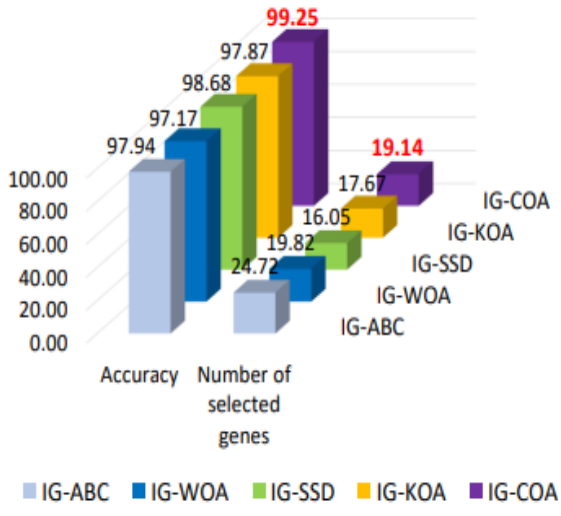


*Fig.3: The average performance of IG-COA algorithms and other algorithms overall datasets*

#### 4.4.2. Comparison with traditional gene selection methods

The IG-COA method is compared with some well-known gene selection methods, including IG, Fisher Score (FS), Relieff (RF), and Minimum Redundancy, Maximum Relevance (mRMR). As shown in Table 4, overall, the developed IG-COA method outperforms other filter methods in terms of average accuracy and the number of selected genes. IG-COA is the only method which obtains 100% accuracy for DS 1, DS 3, and DS 6, whereas in DS 1, and DS 2 several filter methods achieve 100% accuracy, with a high number of selected genes compared with the proposed IG-COA method.

Table 2: Parameter settings for the swarm optimization algorithms used in the experiments

| Parameter Name | Value |
|---|---|
| Population size | 20 |
| Lower bounds | 0 |
| Upper bounds | 99 |
| Maximum number of iterations | 150 |
| Number of runs | 20 |

#### 4.4.3. Comparison with the up-to-date gene selection methods

The proposed IG-COA method is compared with some of the most advanced methods developed between 2012 to 2023 as presented in Table 5. The chosen literature included diverse categories of gene-selection methods: ensemble, filter, wrapper, and hybrid. The IG-COA method surpasses other up-to-date methods with 100% accuracy for DS 2, DS 3, DS 4, and DS 5, and 99.2% and 99.25% for DS 1 and DS 6, respectively. The proposed IG-COA improves the accuracy of overall datasets by 2.525% and reduces the number of selected genes by 46.69.

Table 3: Comparison with other swarm algorithms in terms of mean accuracy, average number of selected genes and Std. over 20 independent runs

| Dataset | Criteria | IG-ABC | IG-WOA | IG-SSD | IG-KOA | IG-COA |
|---|---|---|---|---|---|---|
| DS 1 | Mean | 96 | 98 | 98.8 | 97.6 | **99.2** |
| | #genes | 49 | 35 | 4.8 | 39 | **36** |
| | Std. | 0 | 0 | 0.97 | 0.8 | 0.97 |
| DS 2 | Mean | 100 | 100 | 100 | 100 | **100** |
| | #genes | 15.6 | 12.0 | 17.0 | 14.0 | **11.4** |
| | Std. | 0 | 0 | 0 | 0 | 0 |
| DS 3 | Mean | 100 | 100 | 99.21 | 99.6 | **100** |
| | #genes | 6.4 | 7.8 | 9.4 | 9.2 | **5.4** |
| | Std. | 0 | 0 | 0.96 | 0.78 | 0 |
| DS 4 | Mean | **100** | 100 | 100 | 100 | 100 |
| | #genes | **7.4** | 8.3 | 9.6 | 8.2 | 8.0 |
| | Std. | 0 | 0 | 0 | 0 | 0 |
| DS 5 | Mean | 100 | 100 | 100 | 100 | **100** |
| | #genes | 42.9 | 39.0 | 33.1 | 32.8 | **28.8** |
| | Std. | 0 | 0 | 0 | 0 | 0 |
| DS 6 | Mean | 91.70 | 85.0 | 94.05 | 90.0 | **96.30** |
| | #genes | 27.0 | 16.8 | 22.42 | 17.8 | **25.22** |
| | Std. | 0.35 | 0.56 | 0.95 | 0.82 | 1.11 |

Table 4: The performance of IG-COA method, and other commonly used filter-based gene selection methods, in terms of accuracy

| Dataset | Criteria | RF | FS | IG | mRMR | IG-COA |
|---|---|---|---|---|---|---|
| DS 1 | Accuracy | 98.39 | 95.16 | 96.77 | 96.77 | **99.2** |
| | #genes | 100 | 100 | 100 | 100 | **36** |
| DS 2 | Accuracy | 100 | 100 | 100 | 100 | **100** |
| | #genes | 100 | 100 | 100 | 100 | **11.4** |
| DS 3 | Accuracy | 100 | 100 | 100 | 100 | **100** |
| | #genes | 100 | 100 | 100 | 100 | **6.6** |
| DS 4 | Accuracy | 94.44 | 94.44 | 94.44 | 94.44 | **100** |
| | #genes | 100 | 100 | 100 | 100 | **8.0** |
| DS 5 | Accuracy | 96.23 | 30.77 | 46.15 | 46.15 | **100** |
| | #genes | 100 | 100 | 100 | 100 | **28.8** |
| DS 6 | Accuracy | 60.00 | 73.33 | 60.00 | 66.67 | **96.25** |
| | #genes | 100 | 100 | 100 | 100 | **21.49** |

Table 5: The performance of the proposed IG-COA method compared with some of cutting-edge methods

| Type | Year | References | Datasets | Accuracy | Number of selected genes |
|---|---|---|---|---|---|
| Ensemble | 2012 | [44] | DS 3 | 98.8 | N/A |
| | | | DS 4 | 88.04 | N/A |
| | | | DS 6 | 63.33 | N/A |
| Filter | 2015 | [45] | DS 6 | 61.67 | N/A |
| Wrapper | 2017 | [46] | DS 2 | 86.8 | 8.7 |
| | | | DS 4 | 92.0 | 5.6 |
| Hybrid | 2018 | [47] | DS 4 | 94.44 | N/A |
| Filter | 2019 | [48] | DS 3 | 96.85 | 500 |
| | | | DS 4 | 99.44 | 51 |
| Filter | 2019 | [49] | DS 1 | 98.0 | 231 |
| | | | DS 5 | 94.2 | 51 |
| Hybrid | 2020 | [50] | DS 3 | 98.6 | N/A |
| | | | DS 4 | 98.6 | N/A |
| | | | DS 6 | 83.95 | N/A |
| Hybrid | 2021 | [51] | DS 1 | 98.0 | 1263.6 |
| | | | DS 5 | 81.7 | 714.6 |
| Hybrid | 2021 | [52] | DS 2 | 96.8 | 28.27 |
| | | | DS 3 | 99.5 | 20.6 |
| | | | DS 4 | 97.2 | 36.47 |
| | | | DS 6 | 91.0 | 31.27 |
| Hybrid | 2022 | [30] | DS 3 | 100 | 13 |
| | | | DS 4 | 100 | 26 |
| | | | DS 6 | 91.1 | 49 |
| Hybrid | 2022 | [31] | DS 2 | 100 | 25 |
| | | | DS 4 | 100 | 30 |
| Hybrid | 2023 | [33] | DS 2 | 97.14 | 7.0 |
| | | | DS 3 | 98.40 | 26.0 |
| | | | DS 4 | 95.89 | 3.0 |
| | | | DS 6 | 86.67 | 9.0 |
| Hybrid | 2023 | [32] | DS 2 | 97.4 | N/A |
| | | | DS 3 | 99.9 | N/A |
| | | | DS 4 | 98.6 | N/A |
| Hybrid | 2023 | **IG-COA method** | DS 1 | **99.2** | **36** |
| | | | DS 2 | **100** | **11.4** |
| | | | DS 3 | **100** | **5.4** |
| | | | DS 4 | **100** | **8.0** |
| | | | DS 5 | **100** | **28.8** |
| | | | DS 6 | **99.25** | **25.22** |

## 5. Conclusion

Microarray gene expression analysis has been widely used in cancer research to identify molecular subtypes of cancer. However, gene expression data's high-dimensional nature is a major challenge for accurate cancer classification. This research introduces a new gene selection method named IG-COA to tackle this problem. The IG-COA method integrates IG and COA to select the most valuable genes for cancer subtype classification. The COA algorithm is a recent swarm algorithm, and as far as we know, there has yet to be any prior investigation of the COA algorithm to select

features or genes. Moreover, the IG method filters out irrelevant genes. Then, the COA algorithm is utilized to select the optimal gene subset. The proposed IG-COA method is tested on publicly available microarray gene expression datasets and surpasses existing state-of-the-art methods. The IG-COA method enhances the performance of cancer classification models by reducing the number of irrelevant genes while accurately classifying cancer subtypes. This study contributes to advancing gene expression data analysis in cancer research. Despite outperforming other state-of-the-art methods, future research needs to address some areas for improvement. The weaknesses are as follows. Hybrid methods often require more computational resources than individual methods, leading to longer processing times and higher computational costs. Although the computational time of the IG-COA method may be slower than that of filter methods, it is deemed acceptable as a smaller subset of genes with high accuracy is selected. Moreover, the performance of hybrid gene selection methods can depend heavily on the specific combination of algorithms used, making it difficult to generalize findings across different domains.

## 6. References


[1] Sun, L., Kong, X., Xu, J., Xue, Z.A., Zhai, R. and Zhang, S., 2019. A hybrid gene selection method based on Reli- efF and ant colony optimization algorithm for tumor classification. Scientific reports, 9(1), p.8978. https://doi.org/10.1038/ s41598-019-45223-x

[2] Sun, L., Zhang, X.Y., Qian, Y.H., Xu, J.C., Zhang, S.G. and Tian, Y., 2019. Joint neighborhood entropy-based gene se- lection method with fisher score for tumor classification. Applied Intelligence, 49, pp.1245-1259. https://doi.org/10.1007/ s10489-018-1320-1

[3] Pirgazi, J., Alimoradi, M., Esmaeili Abharian, T. and Olyaee, M.H., 2019. An Efficient hybrid filter-wrapper metaheuristic- based gene selection method for high dimensional datasets. Scientific reports, 9(1), p.18580. https://doi.org/10.1038/ s41598-019-54987-1

[4] Osama, S., Shaban, H. and Ali, A.A., 2022. Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review. Expert Systems with Applications, p.118946. https://doi.org/10.1016/j.eswa.2022.118946

[5] Momeni, Z., Hassanzadeh, E., Abadeh, M.S. and Bellazzi, R., 2020. A survey on single and multi omics data mining methods in cancer data classification. Journal of Biomedical Informatics, 107, p.103466. https://doi.org/10.1016/j.jbi.2020.103466

[6] Shen, C. and Zhang, K., 2021. Two-stage improved Grey Wolf optimization algorithm for feature selection on high-dimensional classification. Complex & Intelligent Systems, pp.1-21. https://doi.org/10.1007/s40747-021-00452-4

[7] Liu, C., Wang, W., Zhao, Q., Shen, X. and Konan, M., 2017. A new feature selection method based on a validity index of feature subset. Pattern Recognition Letters, 92, pp.1-8. https://doi.org/10.1016/j.patrec.2017.03.018

[8] Xue, B., Zhang, M. and Browne, W.N., 2012. Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE transactions on cybernetics, 43(6), pp.1656-1671. https://doi.org/10.1109/TSMCB.2012.2227469

[9] Garibay, C., Sanchez-Ante, G., Falcon-Morales, L.E. and Sossa, H., 2015. Modified binary inertial particle swarm optimization for gene selection in DNA microarray data. In Pattern Recognition: 7th Mexican Conference, MCPR 2015, Mexico City, Mexico, June 24-27, 2015, Proceedings 7 (pp. 271-281). Springer International Publishing. https://doi.org/10.1007/978- 3-319-19264-2 26

[10] Mohapatra, P. and Chakravarty, S., 2015, October. Modified PSO based feature selection for Microarray data classification. In 2015 IEEE Power, Communication and Information Technology Conference (PCITC) (pp. 703-709). IEEE. https://doi.org/10.1109/PCITC.2015.7438088

[11] Zhang, Y., Deng, Q., Liang, W. and Zou, X., 2018. An efficient feature selection strategy based on multiple support vector machine technology with gene expression data. BioMed research international, 2018. https://doi.org/10.1155/2018/7538204

[12] Wu, Y.L., Tang, C.Y., Hor, M.K. and Wu, P.F., 2011. Feature selection using genetic algorithm and cluster validation. Expert Systems with Applications, 38(3), pp.2727-2732. https://doi.org/10.1016/j.eswa.2010.08.062

[13] Meenachi, L. and Ramakrishnan, S., 2020. Differential evolution and ACO based global optimal feature selection with fuzzy rough set for cancer data classification. Soft Computing, 24(24), pp.18463-18475. https://doi.org/10.1007/s00500 020-05070-9

[14] Benitez, I.P., Sison, A.M. and Medina, R.P., 2018, April. An improved genetic algorithm for feature selection in the classification of disaster-related Twitter messages. In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 238-243). IEEE. https://doi.org/10.1109/ISCAIE.2018.8405477

[15] Kabir, M.M., Shahjahan, M. and Murase, K., 2012. A new hybrid ant colony optimization algorithm for feature selection. Expert Systems with Applications, 39(3), pp.3747-3763. https://doi.org/10.1016/j.eswa.2011.09.073

[16] Jeong, I.S., Kim, H.K., Kim, T.H., Lee, D.H., Kim,

K.J. and Kang, S.H., 2018. A feature selection approach based on simulated annealing for detecting various denial of service attacks. Software Networking, 2018(1), pp.173-190. https://doi.org/10.13052/ jsn2445-9739.2016.010

[17] Tawhid, M.A. and Ibrahim, A.M., 2020. Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. International Journal of Machine Learning and Cybernetics, 11, pp.573-602. https://doi.org/10.1007/ s13042-019-00996-5

[18] Chatra, K., Kuppili, V., Edla, D.R. and Verma, A.K., 2019. Cancer data classification using binary bat optimization and extreme learning machine with a novel fitness function. Medical & Biological Engineering & Computing, 57, pp.2673-2682. https://doi.org/10.1007/s11517-019-02043-5

[19] Al-Baity, H.H. and Al-Mutlaq, N., 2021. A New Optimized Wrapper Gene Selection Method for Breast Cancer Prediction. Computers, Materials & Continua, 67(3). https://doi:10.32604/cmc.2021.015291

[20] Ram, M., Najafi, A. and Shakeri, M.T., 2017. Classification and biomarker genes selection for cancer gene expression data using random forest. Iranian journal of pathology, 12(4), p.339. https://doi.org/10.1007/s40747- 021-00452-4

[21] Zhu, M. and Song, J., 2013. An embedded backward feature selection method for MCLP classification algorithm. Procedia Computer Science, 17, pp.1047-1054. https://doi.org/10.1016/j.procs.2013.05.133

[22] Wang, H., Jing, X. and Niu, B., 2017. A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data. Knowledge-Based Systems, 126, pp.8-19. https://doi.org/10.1016/j.knosys.2017.04.004

[23] Mahendran, N., Durai Raj Vincent, P.M., Srinivasan, K. and Chang, C.Y., 2020. Machine learning based computational gene selection models: a survey, performance evaluation, open issues, and future research directions. Frontiers in genetics, 11, p.603808. https://doi.org/10.3389/fgene.2020.603808

[24] Hira, Z.M. and Gillies, D.F., 2015. A review of feature selection and feature extraction methods applied on microarray data. Advances in bioinformatics, 2015. https://doi.org/10.1155/2015/198363

[25] Sun, Y., Lu, C. and Li, X., 2018. The cross-entropy based multi-filter ensemble method for gene selection. Genes, 9(5), p.258. https://doi.org/10.3390/genes9050258

[26] Ghosh, M., Adhikary, S., Ghosh, K.K., Sardar, A., Begum, S. and Sarkar, R., 2019. Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods. Medical & biological engineering & computing, 57, pp.159-176. https://doi.org/10.1007/s11517-018-1874-4

[27] Abdulla, M. and Khasawneh, M.T., 2020. G-Forest: an ensemble method for cost-sensitive feature selection in gene expression microarrays. Artificial Intelligence in Medicine, 108, p.101941. https://doi.org/10.1016/j.artmed.2020.101941

[28] Deng, X., Li, M., Deng, S. and Wang, L., 2022. Hybrid gene selection approach using XGBoost and multi-objective genetic algorithm for cancer classification. Medical & Biological Engineering & Computing, 60(3), pp.663-681. https://doi.org/10.1007/ s11517-021-02476-x

[29] Poongodi, K. and Sabari, A., 2022. Identification of Bio-Markers for Cancer Classification Using Ensemble Approach and Genetic Algorithm. Intelligent Automation & Soft Computing, 33(2). https://doi.org/10.32604/iasc.2022.023038

[30] Alrefai, N. and Ibrahim, O., 2022. Optimized feature selection method using particle swarm intelligence with ensemble learning for cancer classification based on microarray datasets. Neural Computing and Applications, 34(16), pp.13513 13528. https://doi.org/10.1007/s00521-022-07147-y

[31] Kundu, R., Chattopadhyay, S., Cuevas, E. and Sarkar, R., 2022. AltWOA: Altruistic Whale Optimization Algorithm for feature selection on microarray datasets. Computers in biology and medicine, 144, p.105349. https://doi.org/10.1016/j.compbiomed.2022.105349

[32] Akhavan, M. and Hasheminejad, S.M.H., 2023. A two-phase gene selection method using anomaly detection and genetic algorithm for microarray data. Knowledge-Based Systems, p.110249. https://doi.org/10.1016/j.knosys.2022.110249

[33] Li, M., Ke, L., Wang, L., Deng, S. and Yu, X., 2023. A novel hybrid gene selection for tumor identification by combining multifilter integration and a recursive flower pollination search algorithm. Knowledge-Based Systems, p.110250. https://doi.org/10.1016/j.knosys.2022.110250

[34] Dehghani, M., Montazeri, Z., Trojovska, E. and Trojovský, P., 2023. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowledge-Based Systems, 259, p.110011. https://doi.org/10.1016/j.knosys.2022.110011

[35] Shukla, A.K. and Tripathi, D., 2020. Detecting biomarkers from microarray data using distributed correlation based gene selection. Genes & genomics, 42, pp.449-465. https://doi.org/10.1007/s13258-020-00916-w

[36] Dabba, A., Tari, A., Meftali, S. and Mokhtari, R., 2021. Gene selection and classification of microarray data method based on mutual information and moth flame algorithm. Expert Systems with Applications, 166, p.114012. https://doi.org/10.1016/j.eswa.2020.114012

[37] Houssein, E.H., Abdelminaam, D.S., Hassan, H.N., Al-Sayed, M.M. and Nabil, E., 2021. A hybrid barnacles mating optimizer algorithm with support vector machines for gene selection of microarray cancer classification. IEEE Access, 9, pp.64895-64905.https://doi.org/10.1109/ACCESS.2021.3075942

[38] E.J., Ross, M.E., Shurtleff, S.A., Williams, W.K., Patel, D., Mahfouz, R., Behm, F.G., Raimondi, S.C., Relling, M.V., Patel, A. and Cheng, C., 2002. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer cell, 1(2),

pp.133-143. https://doi.org/10.1016/S1535-6108(02)00032- 6

[39] Zhu, Z., Ong, Y.S. and Dash, M., 2007. Markov blanket-embedded genetic algorithm for gene selection. Pattern Recognition,40(11), pp.3236-3248. https://doi.org/10.1016/j.patcog.2007.02.007

[40] Petricoin, E.F., Ardekani, A.M., Hitt, B.A., Levine, P.J., Fusaro, V.A., Steinberg, S.M., Mills, G.B., Simone, C., Fishman, D.A., Kohn, E.C. and Liotta, L.A., 2002. Use of proteomic patterns in serum to identify ovarian cancer. The lancet, 359(9306), pp.572-577. https://doi.org/10.1016/S01406736(02)07746- 2

[41] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A. and Bloomfield, C.D., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science, 286(5439), pp.531-537. https://doi.org/10.1126/science.286.5439.531

[42] West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson Jr, J.A., Marks, J.R. and Nevins, J.R., 2001. Predicting the clinical status of human breast cancer by using gene expression profiles. Proceedings of the National Academy of Sciences, 98(20), pp.11462-11467. https://doi.org/10.1073/pnas.201162998

[43] Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerova, L.C., Black, P.M., Lau, C. and Allen, J.C., 2002. Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature, 415(6870), pp.436-442. https://doi.org/10.1038/415436a

[44] Bolon-Canedo, V., Sanchez-Maroño, N. and Alonso-Betanzos, A., 2012. An ensemble of filters and classifiers for microarray data classification. Pattern Recognition, 45(1), pp.531-539. https://doi.org/10.1016/j.patcog.2011.06.006

[45] Michel, R., García-Torres, M., Schaerer, C.E. and Divina, F., 2015, September. Feature selection via approximated Markov blankets using the CFS method. In 2015 International Workshop on Data Mining with Industrial Applications (DMIA) (pp.38-43). IEEE. https://doi.org/10.1109/DMIA.2015.17

[46] Wang, A., An, N., Yang, J., Chen, G., Li, L. and Alterovitz, G., 2017. Wrapper-based gene selection with Markov blanket. Computers in biology and medicine, 81, pp.11-23. https://doi.org/10.1016/j.compbiomed.2016.12.002

[47] Nagpal, A. and Singh, V., 2018. A feature selection algorithm based on qualitative mutual information for cancer microarray data. Procedia computer science, 132, pp.244-252. https://doi.org/10.1016/j.procs.2018.05.195

[48] Cilia, N.D., De Stefano, C., Fontanella, F., Raimondo, S. and Scotto di Freca, A., 2019. An experimental comparison of feature- selection and classification methods for microarray datasets. Information, 10(3), p.109. https://doi.org/10.3390/info10030109

[49] GÜÇKIRAN, K., Cantürk, İ. and ÖZYILMAZ, L., 2019. DNA microarray gene expression data classification using SVM, MLP, and RF with feature selection methods relief and LASSO. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi,23(1). https://doi.org/10.19113/sdufenbed.453462

[50] Kilicarslan, S., Adem, K. and Celik, M., 2020. Diagnosis and classification of cancer using hybrid model based on ReliefF and convolutional neural network. Medical hypotheses, 137, p.109577. https://doi.org/10.1016/j.mehy.2020.109577

[51] Alzaqebah, M., Briki, K., Alrefai, N., Brini, S., Jawarneh, S., Alsmadi, M.K., Mohammad, R.M.A., ALmarashdeh, I., Alghamdi, F.A., Aldhafferi, N. and Alqahtani, A., 2021. Memory based cuckoo search algorithm for feature selection of gene expression dataset. Informatics in Medicine Unlocked, 24, p.100572. https://doi.org/10.1016/j.imu.2021.100572

[52] Dabba, A., Tari, A. and Meftali, S., 2021. Hybridization of Moth flame optimization algorithm and quantum computing for gene selection in microarray data. Journal of Ambient Intelligence and Humanized Computing, 12(2), pp.2731-2750. https://doi.org/10.1007/s12652-020-02434-9

[53] Tabakhi, S., Najafi, A., Ranjbar, R. and Moradi, P., 2015. Gene selection for microarray data classification using a novel ant colony optimization. Neurocomputing, 168, pp.1024-1036. https://doi.org/10.1016/j.neucom.2015.05.022

[54] Drucker, H., Wu, D. and Vapnik, V.N., 1999. Support vector machines for spam categorization. IEEE Transactions on Neural networks, 10(5), pp.1048-1054. https://doi.org/10.1109/72.788645