Artificial Intelligence in Medicine 55 (2012) 163–175

Contents lists available at SciVerse ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Acute leukemia classification by ensemble particle swarm model selection

Hugo Jair Escalante[a,b,*], Manuel Montes-y-Gómez[a,c], Jesús A. González[a], Pilar Gómez-Gil[a], Leopoldo Altamirano[a], Carlos A. Reyes[a], Carolina Reta[a], Alejandro Rosales[a]

[a] National Institute of Astrophysics, Optics and Electronics, Department of Computational Sciences, Luis Enrique Erro # 1, Tonantzintla, Puebla 72840, Mexico
[b] Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, Ciudad Universitaria, San Nicolás de los Garza, NL 66450, Mexico
[c] Department of Computer and Information Sciences, University of Alabama at Birmingham, Birmingham, AL 35294, USA

ARTICLE INFO

Article history:
Received 21 July 2011
Received in revised form 13 March 2012
Accepted 16 March 2012

Keywords:
Ensemble learning
Swarm optimization
Full model selection
Morphological classification
Analysis of bone marrow cell images
Acute leukemia classification

ABSTRACT

Objective: Acute leukemia is a malignant disease that affects a large proportion of the world population. Different types and subtypes of acute leukemia require different treatments. In order to assign the correct treatment, a physician must identify the leukemia type or subtype. Advanced and precise methods are available for identifying leukemia types, but they are very expensive and not available in most hospitals in developing countries. Thus, alternative methods have been proposed. An option explored in this paper is based on the morphological properties of bone marrow images, where features are extracted from medical images and standard machine learning techniques are used to build leukemia type classifiers.
Methods and materials: This paper studies the use of ensemble particle swarm model selection (EPSMS), which is an automated tool for the selection of classification models, in the context of acute leukemia classification. EPSMS is the application of particle swarm optimization to the exploration of the search space of ensembles that can be formed by heterogeneous classification models in a machine learning toolbox. EPSMS does not require prior domain knowledge and it is able to select highly accurate classification models without user intervention. Furthermore, specific models can be used for different classification tasks.
Results: We report experimental results for acute leukemia classification with real data and show that EPSMS outperformed the best results obtained using manually designed classifiers with the same data. The highest performance using EPSMS was of 97.68% for two-type classification problems and of 94.21% for more than two types problems. To the best of our knowledge, these are the best results reported for this data set. Compared with previous studies, these improvements were consistent among different type/subtype classification tasks, different features extracted from images, and different feature extraction regions. The performance improvements were statistically significant. We improved previous results by an average of 6% and there are improvements of more than 20% with some settings. In addition to the performance improvements, we demonstrated that no manual effort was required during acute leukemia type/subtype classification.
Conclusions: Morphological classification of acute leukemia using EPSMS provides an alternative to expensive diagnostic methods in developing countries. EPSMS is a highly effective method for the automated construction of ensemble classifiers for acute leukemia classification, which requires no significant user intervention. EPSMS could also be used to address other medical classification tasks.

1. Introduction

According to the Leukemia and Lymphoma Society, "*leukemia is a malignant disease (cancer) of the bone marrow and blood characterized by an uncontrolled accumulation of blood cells* [1]". Leukemia is divided in myelogenous and lymphocytic types, where both types can be acute or chronic (which progresses slowly compared to acute leukemia). A few highlights taken from the Leukemia and Lymphoma Society facts 2010–2011 are as follows [1]:

- It is estimated that 259,889 people in the USA are living with, or are in remission from, leukemia.
- An estimated 43,050 new cases of leukemia will be diagnosed in the USA during 2011.
- In 2010, leukemia was expected to affect more than 10 times as many adults (39,733) as children (3317, aged 0–14 years).

* Corresponding author at: National Institute of Astrophysics, Optics and Electronics, Department of Computational Sciences, Luis Enrique Erro # 1, Tonantzintla, Puebla 72840, Mexico. Tel.: +52 222 2663100x8319; fax: +52 222 2663152.
E-mail address: hugojair@inaoep.mx (H.J. Escalante).

0933-3657/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2012.03.005

- The most common type of childhood leukemia (0–19 years old) is acute lymphocytic leukemia (ALL).
- In 2007, the most recent year for which data are available, 74% of new ALL cases occurred among children (approximately 2859 cases, aged 0–19 years).
- In 2010, it was anticipated that approximately 21,840 deaths (12,660 males and 9180 females) would be attributable to leukemia in the USA, i.e., 8950 attributable to acute myelogenous leukemia (AML), 4390 to chronic lymphocytic leukemia (CLL), 1420 to ALL, and 440 to chronic myeloid leukemia (CML).

Although these figures relate to the USA, proportional statistics are expected for other countries. Thus, it is clear that medical and technological advances in the understanding of leukemia will have a broad impact on the entire world population. In particular, acute leukemia is a deadly disease and the morphological identification of leukocytes is a fundamental task in its detection (the focus of this study). Acute leukemia may be either ALL or AML, with the following acute leukemia subtypes according to the French–American–British classification [2]: L1, L2, and L3 in the lymphocytic family; and M0, M1, M2, M3, M4, M5, M6, and M7 in the myelogenous family.

The morphological identification of acute leukemia is mainly performed by chemists and hematologists. The process starts when a bone marrow sample is taken from the patient's spine, which is prepared as a smear with Wright's staining method. This makes the white globules more visible during analysis. Depending on the economic resources of the hospital (because this equipment is very expensive), a flow cytometry test is conducted so the specific leukemia type and sub-type can be identified. After an accurate diagnosis, the appropriate treatment can be given to the patient. The flow cytometer test is usually considered reliable, which can make the morphological analysis obsolete, although high costs can mean that the morphological test is used. Indeed, most of the hospitals found in developing countries do not have flow cytometers, so morphological analysis is still required. During the morphological analysis, chemists and hematologists study the type and maturity level of leukocytes in the bone marrow sample. They use their knowledge to analyze the morphology of leukocytes to identify the type and subtype of acute leukemia. The identification by experts is reliable, but automated tools would be useful to support experts and reduce the costs for health institutions.

1.1. Proposed solution

This paper describes an automated approach for morphological acute leukemia classification from images based on machine vision and machine learning techniques. The proposed method consists of three main phases: cell segmentation, feature extraction, and classification. In a previous study, we focused on segmentation and feature extraction [3,4], whereas we concentrate on the classification stage in this paper. More specifically, we focus on the problem of selecting the best classification model (formed by data preprocessing, feature selection, and classification methods) to provide maximum classification accuracy.

In previous studies, physicians have reported errors of up to 40% when classifying acute leukemia subtypes.[1] We have obtained accuracies close to 90% (in average) in our own previous studies by manually combining methods of preprocessing, feature selection, and classification, while trying to identify appropriate parameters to increase accuracy [4]. Although we obtained satisfactory results via the manual selection of classification models, this method was extremely difficult and time-consuming. Thus, automatic methods for effectively selecting classifiers are required. Clearly, increasing the classification accuracy translates into improved diagnoses by the hematologist, which in turn means that patients will have a better treatments and an increased life expectancy.

We aimed to develop more powerful tools to improve the classification accuracy of our previous study [3,4] and provide more reliable tools for medical diagnosis. Previously, we were mainly focused on segmentation algorithms. We also worked on the extraction of descriptive features/characteristics for acute leukemia cells. However, we did not find a combination of algorithms, parameters, and sets of descriptive characteristics, guaranteeing outstanding results. In order to achieve our goal, we propose the use of ensemble particle swarm model selection (EPSMS) for the automatic selection of accurate classifiers for the morphological identification of acute leukemia.

EPSMS is a generic tool that explores the search space of candidate classifiers to automatically build ensemble classifiers [5]. The main benefit of EPSMS is that it can obtain very effective classification models without user intervention. EPSMS selects ensembles instead of single models [6,7], so it is more robust to noisy data and it provides more stable predictions. A distinctive feature of EPSMS is that the ensemble classifier is formed of heterogeneous full models, where a full model is composed of methods for preprocessing, feature selection, and classification. We use EPSMS for type/subtype acute leukemia classification in an one-vs-all classification method where we selected ad hoc classifiers for each binary type/subtype problem. This was advantageous because different classification problems may require different classification models. Our results compared favorably with those reported in a previous study [3,4], where classification models were constructed manually after machine learning experts spent long periods of time in development. Furthermore, the models selected using EPSMS can provide insights into distinctive features of the acute leukemia type/subtype classification task. The improvements in performance were significant and they may motivate further research on the application of EPSMS to other medical tasks.

As mentioned earlier, we adopted a morphological approach because this is an inexpensive method that is available in many hospitals in developing countries. We are aware of other more precise options for addressing the problem, such as flow cytometry or microarray gene expression analysis techniques. Unfortunately, these techniques are not accessible to most people living in poor countries.[2] We will also compare the performance of our technique with other approaches when data is available (as the cost of tests decrease) and when the method can be offered to more people. This is part of our future work. Meanwhile, we think the proposed approach is a practical alternative to expensive techniques. This statement is supported by experimental results that show the proposed approach achieves very similar performance to that obtained with alternative and more expensive methods.

The rest of this paper is organized as follows. The next section reviews related work on acute leukemia classification and ensemble member selection. Section 3 describes particle swarm model selection, which is the method EPSMS is based on. Section 4 introduces EPSMS and Section 5 describes how EPSMS was used for acute leukemia classification. Section 6 reports our experimental results acute leukemia classification. Finally, Section 7 summarizes our main findings and outlines future work areas.

[1] According to the information provided by physicians that perform morphological classification manually at the Mexican Social Security Institute (IMSS).

[2] At the moment of writing this paper, the cost of leukemia studies in Mexico range from 100 USD to 1300 USD. A large portion of Mexican population receives 100 USD or less as payment for a moth of work.

2. Related work

This section reviews related work on acute leukemia classification and classification using ensemble methods.

2.1. Acute leukemia classification

Many studies have been devoted to the development of accurate methods for automatically detecting different types of leukemia. Huang et al. reported several methods for the recognition and classification of leukemia [8]. They focused on the seminal work of Golub et al. who presented the first microarray-based and bioinformatics-oriented approaches for identifying and classifying tumor types [9,10]. In that study they used a signal-to-noise statistic to select a small set of genes, before developed a scheme based on microarray gene expression analysis to distinguish ALL from AML, and they reported recognition rates of 94.1%. Other research studies were inspired by Golub et al. and they used the same ALL/AML data sets presented in [8]. These studies applied models, such as multilayer perceptron networks, support vector machines, and the *k*-nearest neighbor method, where the accuracy ranged from 58% to 97%.

Li et al. proposed two Bayesian classification algorithms, which incorporate feature selection, for the classification of gene expression data derived from cDNA microarrays [11]. The authors evaluated their methods using three gene expression data sets for colon cancer, ovarian cancer and leukemia (ALL vs AML). In addition to providing acceptable performance, the proposed methods provide sparse solutions.

Zong et al. developed a tool for the identification of different white blood cell categories in a given blood sample [12]. Two approaches were implemented with two different parametric data clusters. In the first, a multidimensional space using artificial neural networks (ANNs) was trained followed by cross-validation using cytometry data. The second approach exploited gene expression profiling of ALL to classify its six subtypes. The system was also trained to assess the inherent problem of data overlap and to recognize abnormal blood cell patterns. The classification performance of the first approach reached up to 100% while the performance of the second approach was up to 92%. A novel ANN algorithm for optimizing the classification of multidimensional data, which focused on acute leukemia samples, was proposed by Adjouadi et al. [13]. The ANN technique classifies normal vs abnormal (i.e., ALL and AML) blood samples. The authors reported classification results of up to 96.67% with an increased data set size.

However, despite the very good results obtained using microarray information, the process for gathering this data is complex and expensive. In this paper we propose an approach based on the morphological analysis of bone marrow, which achieved a similar performance to that reported with microarray data but that does not require sophisticated equipment. This means it can be applied in most health facilities in developing countries. Like microarray data analysis, flow cytometry studies are also very precise (e.g., 99.99% confident). However, they are also very expensive and only a few hospitals (in developing countries) can afford them. Therefore, compared with other methods our morphological method for acute leukemia identification offers a better tradeoff between low cost and accuracy, and we think it may have a broad impact.

The studies reviewed above are a representative sample of the wide variety of methods proposed for acute leukemia classification. Different methods for data preprocessing, feature selection, and classification have been developed in different studies, which have proved to be very effective with different acute leukemia data sets. These methods have been manually designed/selected by experts on machine learning and/or leukemia classification. Although this development scenario is acceptable, there are many situations where both types of experts are not available, so automated methods for the construction of classification models are required. Even in scenarios where expert knowledge is available, the availability of automated methods for classifier construction can simplify the design and development process. In this study, we explore the use of automatic methods for the construction of classification models for acute leukemia classification. In particular, we study the benefits of using a technique for the automatic construction of ensembles of full models, where a full model is composed of methods for data preprocessing, feature selection, and classification.

2.2. Ensemble classifiers

The underlying idea of ensemble methods is that we can obtain more accurate and more robust predictions by considering multiple views of the same problem. This is justified by theoretic and empirical studies showing that, under certain conditions, a combination of multiple individual models is beneficial in terms of accuracy and the stability of predictors [14]. However, despite the fact that the ensemble learning paradigm has been studied for more than two decades, there are still some open issues that merit further study. One issue is the selection of a set of classifiers for creating an ensemble [14–17].

Previous studies suggest that the effectiveness of ensembles depends on the accuracy and diversity of the individual models [14–17]. Thus, successful ensemble methods aim to guarantee (at least) *default accuracy* and high diversity among its members by adopting different strategies. For example, learning weights for weak learners [18], randomizing the sets of features and examples that are considered for each classifier [19,20], partitioning the input space into clusters and learning different classifiers for the different clusters [21], determining the most appropriate classifier (from a predefined set) for each test example according to distance measures [22], or using different learning algorithms for each individual model [17,23–26]. The latter strategy, known as heterogeneous ensembles, is most relevant to our work.

Heterogeneous ensembles are based on an assumption that decision functions will be different because different learning algorithms have different biases, which may lead to higher diversity among the individual models. However, a problem with this approach is that it is not clear how to select the learning algorithms that will form the ensemble. Some researchers have adopted diverse search strategies for the selection of a set of models so the performance of the ensemble is optimized under a certain fusion strategy [17,23–26]. They have considered a pool of classification algorithms and used combinatorial optimization techniques to select the combination of methods that maximizes the performance of the ensemble [24,27,28]. Some researchers have also attempted to optimize the weight associated to each method in the ensemble [25,26]. However, the parameters of the learning algorithms are fixed in these approaches so the classifiers are not really optimized for individual problems. In addition, the same data preprocessing methods and feature selection techniques are used for all of the models that are considered in the ensemble, thereby reducing the potential diversity of the members of the ensemble. In this study, we built ensembles using heterogeneous classifiers, which were composed of different methods for data preprocessing, feature selection, and classification. We also used different parameters for the individual models. These methods were selected using full model selection methods.

Full model selection methods are aimed at selecting the best combination of methods for preprocessing, feature selection, and classification, starting with a training data set [6,7,29]. The main benefit of these methods is that very effective classification models can be obtained for diverse classification problems without

spending much time on the design and development of specific models for these problems. Thus, the work of the data analyst is greatly simplified. The main hypothesis of this study was that full model selection methods could be helpful for selecting member classifiers when building ensembles [5].

To the best of our knowledge, the two main full model selection strategies proposed to date are described in [6,7]. Both methods are based on the same formulation, they explore the search space of full models that can be generated using methods from different machine learning toolboxes, where the two methods differ in the technique used to explore the search space. Gorissen et al. used genetic algorithms for model type selection [7], whereas Escalante et al. proposed particle swarm model selection (PSMS) [6]. Both techniques reported satisfactory results in diverse domains. In this study, we focus on the suitability of models selected with PSMS for building ensembles and we postpone until future work the study of models selected using Gorrisen et al.'s method for building ensembles.

3. Particle swarm model selection

PSMS is the application of particle swarm optimization (PSO) to the problem of full model selection (FMS) [6]. Given a pool of methods for data preprocessing, feature selection, and pattern classification, and a data set associated with a classification task, FMS is the task of selecting the best combination of methods such that an estimate of generalization performance is maximized for the classification task. In addition, the hyper-parameters must be optimized for each of the selected methods. Thus, PSMS may be considered as a black-box tool that receives the input data set for a classification task and returns a very effective classification model.

A full model is comprised of the serial application of preprocessing, feature selection, and classification methods. For example, in the challenge learning object package (CLOP) [30], (the machine learning toolbox considered in this study), a sample full model is as follows:

$$chain(normalize(c = 0), \quad relief(m = 25), \quad svc(d = 0; \gamma = 0.13)),$$

where `chain` is the CLOP operator for building serial models. In this model, the data is first normalized (`normalize`) without previously centering (`c = 0`), before the `relief` technique is used for feature selection to select a maximum of 25 features (`m = 25`). Finally, the resulting data is used to train a support vector classifier (`svc`) with an `rbf` kernel and width $\gamma = 0.13$. The full list of methods available in the CLOP toolbox is described in Table 1. Thus, PSMS explores the space containing all the possible combinations of methods and parameters in Table 1 using PSO.

PSO was originally proposed by Kennedy and Eberhart [31], it is a population-based search heuristic that mimics the behavior of biological societies where individuals have common goals and show social and individual behaviors (e.g., swarms of bees or flocks of birds) [32]. Like evolutionary algorithms, PSO is useful when other techniques are not applicable such as gradient descend or direct analytical discovery. Combinatorial and real-valued optimization problems where the optimization surface possesses many locally optimal solutions (e.g., FMS) are well suited to swarm optimization. Comparable performances of PSO and other evolutionary computation methods (e.g., genetic algorithms) have been reported in the literature [32,33]. However, we selected PSO for FMS, rather than evolutionary algorithms, because of its simplicity and generality, and because no ad hoc modification was required when applying it to FMS [6]. PSO is easier to implement than evolutionary algorithms because it involves only a single operator for updating solutions. In contrast, evolutionary algorithms require a particular representation and specific methods for crossover, mutation, speciation, and selection.

Table 1
Classification (C), feature selection (F), and preprocessing (P) methods considered in our experiments. We show the method name and number of parameters for each technique.

Object name	Type	# pars.	Description
zarbi	C	0	Linear classifier
naive	C	0	Naïve Bayes
logitboost	C	3	Boosting with trees
neural	C	4	Neural network
svc	C	4	SVM classifier
kridge	C	4	Kernel ridge regression
rf	C	3	Random forest
lssvm	C	5	Kernel ridge regression
Ftest	F	4	*F*-test criterion
Ttest	F	4	*T*-test criterion
aucfs	F	4	AUC criterion
odds-ratio	F	4	Odds ratio criterion
relief	F	3	Relief ranking criterion
Pearson	F	4	Pearson correlation coefficient
ZFilter	F	2	Statistical filter
s2n	F	2	Signal-to-noise ratio
pc-extract	F	1	Principal components analysis
svcrfe	F	1	SVC-recursive feature elimination
normalize	P	1	Data normalization
standardize	P	1	Data standardization
shift-scale	P	1	Data scaling

In the basic implementation of PSO, the solutions to the problem at hand are coded as vectors of real numbers $\mathbf{x}_i \in \mathbb{R}^d$, where d is the dimensionality of the solutions. $\mathbf{x}_i$ is referred to as the ith particle or the position of the ith particle in the search space. Each particle is associated to a vector of velocities $\mathbf{v}_i \in \mathbb{R}^d$ that specifies how particles move in the search space.

Initially, m-solutions and their velocities are randomly initialized. Next, each solution is updated iteratively by considering the previous best position for that solution (personal best) and the global best solution so far (global best) [32]. The personal best solution introduces local knowledge from the previous positions of each particle, whereas the leader particle (global best) provides global knowledge. The goodness of solutions is evaluated by means of a fitness function. We considered the following updating equations for the PSO algorithm:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \tag{1}$$

$$\mathbf{v}_i^{t+1} = (W \times v_i^t) + (c_1 \times r_1 \times (\mathbf{p}_i - \mathbf{x}_i^t)) + (c_2 \times r_2 \times (\mathbf{g}^t - \mathbf{x}_i^t)) \tag{2}$$

where $\mathbf{p}_i$ is the best position obtained by particle (solution) $\mathbf{x}_i$ (i.e., personal best), $\mathbf{g}^t$ is the best particle in the swarm up to iteration t (i.e., global best), c_1 and c_2 are constant weights for the contribution of local and global knowledge, whereas $r_1, r_2 \sim U[0, 1]$ are random numbers. W is the so called inertia term, which weights the contribution of the previous velocity into the new one, see [32] for details. PSO ends when a fixed number of iterations (t_{max}) is performed. The PSO implementation we used is shown in Fig. 1.

In a previous study, we performed an extensive analysis of the influence of PSO parameters: c_1, c_2, t_{max}, W, and m, on the performance of PSMS and EPSMS [5,6,29]. In agreement with literature on PSO [32,34], we found that the most influential parameters were c_1, c_2 and W. In particular, we found that the configuration $c_1 = c_2 = 2$ and the adaptive inertia weight[3] W allowed PSMS to select very effective classification models and, more importantly, it allowed PSMS to avoid overfitting to some extent, see [6] for a detailed

[3] W is initialized with the value $w_{start} = 1.2$ and then W is linearly decreased through the iterations until the value $w_{end} = 0.4$.

Algorithm 1 Particle swarm optimization (PSO).

Requires:
– $\mathbf{c}_1, \mathbf{c}_2$: weights for local and global information;
– $\mathbf{m}$: number of particles in the swarm;
– t_{max}: number of iterations;
– $\mathbf{W}$: inertia weight
Initialize swarm ($\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$)
Compute fitness function $fitness(\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\})$
Identify global best ($\mathbf{g}^t$) solution
Identify personal best solutions ($\mathbf{p}_{1,\ldots,m} = \mathbf{x}_{1,\ldots,m}$)
$t = 1$
while $t < t_{max}$ **do**
  for all $\mathbf{x}_i \in \mathbf{S}$ **do**
    Calculate velocity $\mathbf{v}_i$ for $\mathbf{x}_i$ (Equation (2))
    Update position of $\mathbf{x}_i$ (Equation (1))
    Compute $fitness(\mathbf{x}_i)$
    Update $\mathbf{p}_i$ (*if needed*)
  end for
  Update $\mathbf{p}_g^t$ (*if needed*)
  Decrease $\mathbf{W}$
  t^{++}
end while
return $\mathbf{p}_g^t$

Fig. 1. Pseudo-code of the particle swarm optimization (PSO) algorithm.

discussion on parameter selection for PSMS. Thus, we used this configuration of parameters in our experiments.

In PSMS, we defined a codification of full models using their parameters as vectors of real numbers and we used the straight PSO implementation shown in Fig. 1. The fitness function for PSMS was the k-fold cross-validation (CV) error of the full models using training data ($k = 2$ was used for the experiments reported in this paper). We used CV as a mechanism to avoid overfitting, although it is known that the model selection criteria (CV in our case) can also be overfitted [35]. However, in a previous study we found that the search method performed in PSMS was helpful for avoiding overfitting to some extent [6]. This was due to a combination of several factors, i.e., the incorporation of local and global information in the generation of new solutions (through the local and global best solutions), the inclusion of an adaptive inertia weight (i.e., W), the randomness introduced in the process to generate new solutions, and the use of CV. The combination of these factors allowed us to successfully apply PSMS in a variety of domains including authorship verification [36] and object recognition [5,37]. Also, it was evaluated using benchmark data [6] and in several machine learning competitions [6,29,38].

4. Ensemble particle swarm model selection

EPSMS is an extension of PSMS, which has the goal of building ensemble classifiers from PSMS's partial solutions [5]. The intuition behind EPSMS is that a combination of candidate solutions can result in ensemble classifiers that are capable of outperforming individual models. EPSMS is motivated by the large number of solutions (i.e., a total of $(t_{max} + 1) \times m$) that are evaluated via PSMS's search process, most of which achieve performance better than random after a few iterations. EPSMS is also motivated by the fact that PSMS's partial solutions are formed by heterogeneous classifiers, i.e., models that differ in terms of the methods used for preprocessing, feature selection, and classification, while also have different hyper-parameter settings. Our hypothesis was that this heterogeneity is correlated with diversity (i.e., models make uncorrelated errors). Both the performance and the diversity of members of ensembles are known to be very influential factors when building accurate ensembles [14–17]. Therefore, we can build very effective ensemble classifiers by carefully selecting PSMS partial solutions to ensure that diversity and high accuracy is guaranteed to some extent. Note that no special coding of solutions is required for EPSMS because this method only selects classifiers generated by PSMS that are potential ensemble members, while it combines the outputs of the identified methods.

4.1. Selection of candidate solutions

In a previous study, we proposed three variants of EPSMS [5], which differ in the manner that PSMS partial solutions were selected for building the ensemble:

- **Best-set.** Uses the set of global best solutions found every h-iterations of PSMS (i.e., the global best is stored after every h-iterations).
- **Swarm.** Uses the resultant swarm at the end of the PSMS search process.
- **Best-per-iteration.** Uses the set formed by the best solution found after each PSMS iteration.

In [5], we evaluated the performance of the three strategies, in terms of the accuracy of the ensembles and the diversity of their members, using benchmark data and an object recognition data set. We found that the **Best-set** and **Swarm** strategies were outperformed (in terms of performance and diversity) by the **Best-per-iteration** approach, so we focused in the latter strategy in this study.

In the **Best-per-iteration** (BPI) formulation, we store the best solution found in the swarm after each iteration t of PSMS and we denote the full model associated with the best particle at iteration t as f_t. Note that f_t may or may not coincide with the global best solution at iteration t (i.e., $\mathbf{g}^t$). In this way, the solutions stored at each iteration are all different (guaranteeing diversity among models to some extent) and they provide better performance than the other models in the swarm at (least in) one iteration (guaranteeing the accuracy of the models to some extent). When the PSMS search process is complete, we have a set of $t_{max} + 1$ solutions,[4] which are used to build the ensemble. At this point, no extra computation is required for selecting the members of the ensemble compared with the straight PSMS, because we simply stored certain full models that were evaluated via the search process of PSMS.

Fig. 2 shows the straight PSMS approach (a), and illustrates how PSMS solutions are selected to build ensembles in EPSMS-BPI (b). We assume that the fitness function is a measure of the misclassification errors of each model using CV. Particles (i.e., full models) are represented as stars in each iteration t, while the circle encloses the global best solution (i.e., $\mathbf{g}^t$). During iterations $t = 2, 3, 4$, the global best does not change, so a total of six particles are depicted (five particles plus the global best solution) for these iterations. PSMS returns the global best solution $\mathbf{g}^{t_{max}}$, although a total of $(t_{max} + 1) \times m$ solutions were evaluated during the search process, see Fig. 2(a).

In Fig. 2(b), the particles enclosed in a square are those selected for building an ensemble in EPSMS-BPI. During the initialization (Init), $m = 5$ solutions are randomly generated. The solution with the lowest fitness value is marked as the global best (blue circle) and this solution is the one with the lowest value, so it is stored to be considered for the ensemble. In iteration $t = 1$, another five new solutions are generated using Eqs. (1) and (2). During this iteration, the global best solution is updated (because a new solution has obtained a lower fitness value) and this global best solution is also stored to be considered for the ensemble. During iteration $t = 2$,

[4] One solution from each of the t_{max} iterations plus the best solution in the initial swarm.

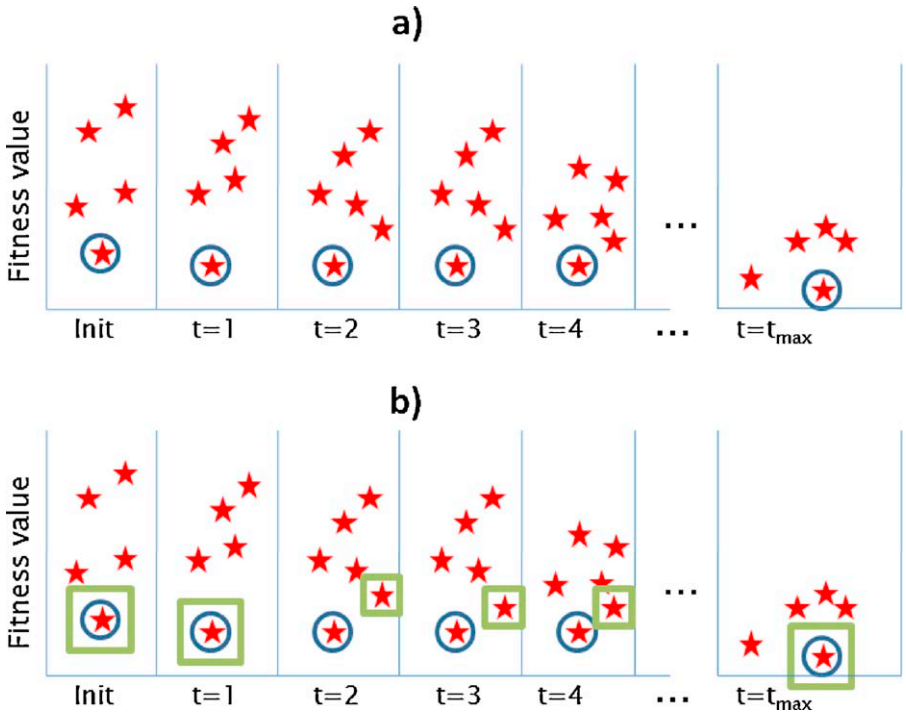

Fig. 2. Illustration of the PSMS (a) and EPSMS (b) approaches. (For interpretation of the references to color in the text, the reader is referred to the web version of the article.)

another five new solutions are generated, but the global best solution is not updated (because none of the solutions generated has a lower fitness value). However, the solution with the lowest fitness value (that enclosed in the square) is stored during this iteration, to be considered for the ensemble. This process is repeated for each iteration of PSMS.

If we only select the global best solutions at each iteration to build the ensemble, we may produce a set of candidate solutions with very low diversity because the global best solutions can be repeated during several iterations of PSMS. This is why the **Best-set** approach was not very helpful for building ensembles in a previous study [5]. With the **Swarm** approach, the m-particles at iteration t_{max} are used to build the ensemble, because this is the set of models produced at the end of the PSMS search process. Most of these solutions are similar, because all of the particles converge to the global best solution with high probability at the end of the search. Thus, the diversity of these models is also low, see [5].

4.2. Ensemble construction

When a set of solutions has been selected, they are combined using a standard averaging strategy to build the ensemble. When a new pattern $\mathbf{p}^T$ needs to be classified, all of the individual models (previously trained using training data) are used to classify the example. Each individual model f_j, $j = \{1, \ldots, t_{max+1}\}$ expresses its confidence for the class of the pattern $\mathbf{p}^T$, we denote with $f_j(\mathbf{p}^T) \in [-1, 1]$ the confidence of f_j for pattern $(\mathbf{p}^T)$. Next, we use the average confidence values as the confidence of the ensemble:

$$f_E(\mathbf{p}^T) = \frac{1}{t_{max}+1} \sum_{j=1}^{t_{max}+1} f_j(\mathbf{p}^T) \tag{3}$$

Finally, we assign the class corresponding to the sign of $f_E(\mathbf{p}^T)$ to the test pattern.

The considered classifiers are potentially heterogeneous, so their outputs are normalized before fusion to ensure that they are on a comparable scale. We use the following normalization method for a classifier f_j :

$$f_j(\mathbf{p}^T) = \frac{f_j(\mathbf{p}^T) - min(f_j(.))}{max(f_j(.)) - min(f_j(.))} \tag{4}$$

where $f_j(\mathbf{p}^T)$ is the output of classifier j for input $\mathbf{p}^T$, $min(f_j(.))$ and $max(f_j(.))$ are the minimum and maximum values, respectively, assigned by the jth classifier to an example in the test set.

4.3. Extensions to EPSMS

We now describe EPSMS modifications and extensions that have not been reported elsewhere. First, we describe an alternative fitness function and we describe an alternative for normalizing the outputs of individual classifiers before building the ensemble.

The original implementations of PSMS [6] and EPSMS [5] used the CV balanced error rate (BER) as the fitness function:

$$\mathrm{BER}(f) = \frac{E_+(f) + E_-(f)}{2} \tag{5}$$

where $E_+(f)$ and $E_-(f)$ are the misclassification error rates for the positive and negative classes, respectively, for model f. The main benefit of BER is that it is well suited to imbalanced domains because it takes into the account error rates of both classes [6,38]. However, although the resulting measure was very useful for model selection in previous studies [5,6,29,36,37], it was limited because it was dependent on a fixed classification threshold. For example, if the classification threshold is 0, examples $\mathbf{p}^T$ where $f(\mathbf{p}^T) \geq 0$ are considered to belong to the positive class whereas examples where $\mathbf{p}^T < 0$ belong to the negative class. In certain classification models, however, the optimal classification threshold may be different from 0 (e.g., a classifier where the outputs are probabilistic), so BER does not reflect the actual performance of those classification models. Since we build ensembles by combining the real output for classifiers (see Eq. (4)), a measure to evaluate the real output of classifiers would be preferable.

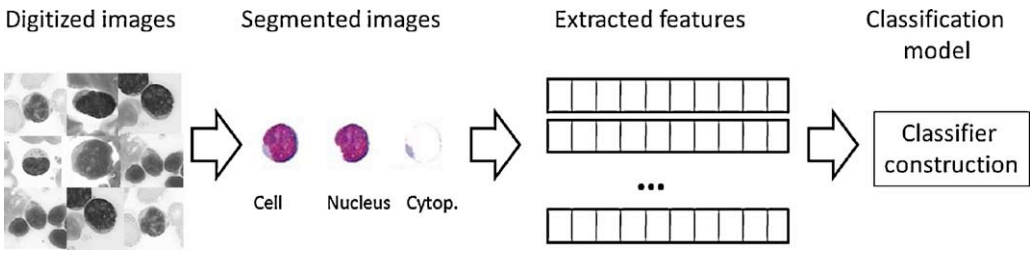

Fig. 3. Diagram of the considered scenario for morphological classification of acute leukemia.

Instead of using BER, we used a fitness function based on the cross-validation area under the ROC curve (AUC) [39], where AUC $\in[0, 1]$. To conform with the previous version of PSMS, which minimized an estimate of the classification errors, we used as fitness function $fitness(\mathbf{x}_i) = (1 - \text{AUC}(\mathbf{x}_i))$, where $\text{AUC}(\mathbf{x}_i)$ was the area under the ROC curve for the model associated with particle $\mathbf{x}_i$. This fitness function is independent of the classification threshold of the model and it directly evaluates the real output of the classification models, so it is well suited to EPSMS. Intuitively, a higher AUC produced by the model indicates that the model was better at assigning higher confidence values to positive examples and lower confidence values to negative ones. It is important to emphasize that AUC compares the performance of models that returned outputs in different scales. AUC is currently one of the main evaluation measures used in most classification tasks [39] (including medical domains), as well as most machine learning competitions that involve classification (see for example the Kaggle website[5] or I. Guyon's machine-learning-challenges website[6]).

The outputs of the members selected for the ensemble must be normalized before fusing them. This is because heterogeneous models may provide outputs at different scales. We previously considered the normalization described in (4), but we wanted to test whether an alternative normalization method might have an impact on the accuracy of the ensemble or if using the raw output from classifiers could result in effective ensemble classifiers. In this study, we used an alternative normalization method, given by:

$$f_j(\mathbf{p}^T) = \frac{f_j(\mathbf{p}^T) - mean(f_j(.))}{stddev(f_j(.))} \tag{6}$$

where $mean(f_j(.))$ is the average of the outputs of the classifier f_j while $stddev(f_j(.))$ is the standard deviation of the outputs of the classifier f_j. Formula (6) is a standardization that is used widely in statistics for hypothesis testing and as a data preprocessing step in machine learning [40]. In Section 6, we report experimental results with PSMS/EPSMS when using both forms of normalization, (4) and (6), and when using the raw outputs of classifiers.

The next section describes how EPSMS was used for acute leukemia classification. EPSMS is a generic tool for the selection of classification models and its performance must be evaluated and validated appropriately in specific application domains, see Section 6. It is desirable to tailor EPSMS to each particular application domain, so it can exploit all of the available prior domain knowledge. We emphasize that the use of EPSMS or any other method for building classification models in the medical domain should not be a substitute for the diagnoses of physicians, and instead they should be considered as decision support tools.

[5] http://www.kaggle.com/.
[6] http://clopinet.com/challenges/.

5. EPSMS for acute leukemia classification

This section describes our proposed approach for acute leukemia classification using EPSMS. We first describe the scenario. Next, we describe the methods used for cell segmentation and feature extraction. Finally, we describe how EPSMS was used to select competitive classification models.

5.1. Morphological identification of leukemia

Our problem was the morphological classification of acute leukemia from digitized bone marrow images. The scenario can be summarized as follows. First, digitized images were obtained. After preprocessing the images (to adjust for the contrast and filtering noise), the first step of the methodology was image segmentation, which involved the identification of regions of interest in images. Features were then extracted from the regions identified, before the classification model was built. Fig. 3 shows the scenario we considered.

In this study, we used a database of cell images from real patients [41]. In each record in our database, we identified smears from patients who were representative of acute leukemia type (lymphocytic or myelogenous). In each case, we had the results of the flow cytometry test and stored the acute leukemia subtype. This selection was performed with the help of domain experts (chemists and hematologists) who carefully helped us to choose samples to digitize based on their experience.

We then digitized our bone marrow smear images using a Carl Zeiss optical microscope with 100 objectives. The microscope was connected to a digital camera via a frame grabber. The process used to obtain an image of a bone marrow smear was as follows. First, the smear was cleaned of dirt particles and located on the microscope slide. The smear was observed by the domain expert using a 10× lens to spot interesting areas. It was possible to find several areas of interest in one smear. The areas of interest were digitized with the 10× lens at a resolution of 800 × 600 pixels. We then used the 100× lens with immersion oil to look for cells of interest inside the selected area. When the cells were located, they were digitized. This process was repeated as necessary, depending on the number of cells of interest found in each area. Next, we segmented the leukocytes and extracted significant characteristics to differentiate among the types and subtypes of acute leukemia cells. The segmentation method and features considered are described in the next section.

5.2. Segmentation and feature extraction

The segmentation phase of our methodology consisted of the isolation of leukemia cells from the digital images obtained in the previous stage. During this process, chemists and/or hematologists were required to identify regions of interest (ROIs). They selected cells that were representative of the types and subtypes of acute leukemia under study (which we used as training/test data in our machine learning task). After the domain expert had selected

Table 2
Description of the features used in our study. Statistical and texture features were extracted from each channel of the image in RGB format. IOD is the integrated optical density.

ID	Type	Cell, nucleus	Cytoplasm
1	Morphological	Area, perimeter, circularity, width, height, elongation, major axis, minor axis, eccentricity, extension, diameter, Euler number, convex area, solidity	Area
2	Statistical	Mode, mean, standard deviation, variance, IOD, average IOD	Mode, mean, standard deviation, variance
3	Texture	Entropy, contrast, correlation, energy, homogeneity	–

Table 3
Number of features used for each configuration of feature/region in our experiments.

Subset	Cell	Nucleus and cytoplasm	Description
A	58	78	Features from Table 2
B	120	120	Principal components
C	178	198	Features from Table 2 and principal components

the ROIs, we used machine vision techniques for segmentation (isolating leukemia cells or ROIs from the rest of the image). For segmentation, we used a method based on Markov random fields, which was previously used for the segmentation of remote sensing images [42]. After the segmentation stage, each cell in each image was divided into two parts: nucleus and cytoplasm. In order to determine which part of the image was the nucleus and which part corresponded to the cytoplasm, the set of rules defined in [43] was used. See [3,4,43] for further details on the segmentation phase and its evaluation during the detection of nuclei and cytoplasm.

After determining which image regions corresponded to the nucleus and cytoplasm, we extracted the visual attributes from these regions. The extracted features were used to represent the cells. Because the experts have identified which cells were associated with specific types of acute leukemia, we used the pairs of features and leukemia types as observations in the classification task. The set of features used in our study are described in Table 2. These features were extracted from the nucleus and cytoplasm, as well as extracted features from the whole cell. In addition to the features listed in Table 2, we performed principal components analysis (PCA) and used the top 30 components from each RGB channel and from the gray scale image for a total of 120 components. Note that PCA was applied directly to the ROIs, rather than the features extracted from the ROIs.

The features extracted from the nucleus and cytoplasm were combined and used as a single region. The performance of features extracted from the nucleus and cytoplasm was compared to the performance obtained with features extracted from the whole cell, i.e., there were two regions from which features could be extracted (the cell and nucleus + cytoplasm). Table 3 shows the number of features for each combination of features (A, B, or C) and the region where features were extracted (the whole cell, or the nucleus and cytoplasm used as individual parts of the cell) in our experiments. For further details on feature extraction we refer the reader to [3,43,4]. The next section describes our classification approach for morphological leukemia classification, which used the features described in this section to represent images.

5.3. Classification with EPSMS

EPSMS is an automatic tool for the selection of highly effective classification models in generic classification tasks. We used EPSMS for type/subtype acute leukemia classification. Our hypothesis was that models selected with EPSMS could achieve comparable or superior performance to that obtained with manually selected models. As well as improving performance, the use of EPSMS has additional benefits, e.g., specific models could be used for the classification of different acute leukemia types/subtypes, without spending long periods of time manually selecting the best model for each task. Thus, experts on machine learning or in the application domain were not required.

EPSMS is designed for binary classification problems where training examples can belong to one of two classes. Thus, the straight EPSMS implementation could be used for most acute leukemia classification tasks in this study. For example, models of leukemia type classification (ALL vs AML), or several configurations of leukemia subtype classification problems (e.g., L1 vs L2 or M1–M2 vs M3) could be selected with straight EPSMS. However, there were other scenarios where the classification problem was associated with more than two classes, i.e., a multiclass classification problem. For example, when the patient was positive for AML and we wanted to know the specific subtype (i.e., M1 vs M2 vs M3), or when we did not know the type of leukemia and we wanted to know the type and subtype of leukemia (e.g., L1 vs L2 vs L3 vs M1 vs M3 vs M5).

In the multiclass classification tasks, we used an one-vs-all (OVA) method because of its efficiency and proven performance [44]. In OVA, a set of K-binary classifiers is built given a multiclass classification problem with K classes, where each classifier is able to discriminate examples as class k_i (positive class) or $k_{\{j:j\neq i\}}$ (negative class), provided each classifier is independent. When a new observation $\mathbf{p}^T$ needs to be classified, it is passed through the K-classifiers and each classifier f^i returns a real value $f^i(\mathbf{p}^T)$ reflecting its confidence in the correct labeling of $\mathbf{p}^T$ suing k_i. The class corresponding to the classifier with the highest confidence is assigned to $\mathbf{p}^T$.

Fig. 4 shows the OVA method used for multiclass acute leukemia subtype classification[7] where the class corresponding to the classifier with the largest confidence value was selected as the output of the multiclass classifier. This is a fairly standard multiclass classification strategy. In previous studies, however, the same classifier f^i was used for all classes [4], whereas in this study a different classifier f^i was used for each class, where each classifier was an ensemble selected using EPSMS. We believe that the selection of ad hoc models for each of the subproblems can have a positive impact on the final multiclass classifier. In Section 6, we provide experimental results that support our hypothesis.

6. Experiments and results

This section describes the experimental results for the application of EPSMS for leukemia subtype classification. The next section describes the experimental methodology we used. Section 6.2 describes the experimental results for leukemia subtype classification using EPSMS.

[7] For the output combination stage, the outputs of the classifiers for each class are standardized using Formula (6)

Table 4
Type statistics for the data set used in our experiments.

Type/subtype	ALL	L1	L2	AML	M2	M3	M5
No. images	295	102	135	338	95	47	56

6.1. Experimental settings

In our experiments, we used the cell image collection from the Mexican Social Security Institute, which contains 633 bone marrow leukemia cell images with different color staining (the images correspond to real cases of acute leukemia). The images in this collection were digitized by Morales et al. [41], as described in Section 5.1. Table 4 shows the number of examples associated with each subtype of acute leukemia included in the collection.

We performed several acute leukemia classification experiments with EPSMS. In each experiment, we tested different classification tasks based on previous work [4]. These classification tasks include a leukemia type classification task (ALL vs AML) and subtype leukemia classification tasks for the types ALL (L1 and L2) and AML (M2, M3 and M5). We also report results for subtype leukemia classification, in the following settings: [M2 vs M3 vs M5] and [L1 vs L2 vs M2 vs M3 vs M5]. In the latter experiments we used a multiclass classification approach with the OVA method [44], see Section 5.

To assess the classification performance of EPSMS we used a 10-fold cross validation approach [45]. In this method, the available data were split randomly into 10 subsets with 10 rounds of training/testing, where in each round nine subsets were used for training (i.e., EPSMS was used for the selection of models and the construction of the ensemble) while one subset was used for testing (i.e., using the constructed ensemble), and a different testing subset was used in each round. The performance of the ensemble was evaluated in the test subset for 10 rounds and the average and standard deviation of its performance was reported. We would like to emphasize that the test samples were not used in any form during the model selection process or for training the selected models, thereby avoiding any "selective bias" [46,47]. In each experiment, we also ensured that the same partitions of training and testing were used by all the methods compared. Thus, all of the methods used exactly the same data for training and testing in each fold of the 10-fold cross validation. We used AUC as the main evaluation measure, see Section 4. In the multiclass classification problem, we reported the average AUC obtained by the independent models and the percentage of correct classifications.

Table 5
Summary of the methods compared in the next sections.

Method – ID	Description
Reference	Best result obtained using a manually selected model, as reported in [4].
Baseline	Random forest classifier, implemented by Dahinden [48].
PSMS	Model selected with straight PSMS.
E1	EPSMS-BPI using Expression (4) to normalize the outputs of individual models.
E2	EPSMS-BPI using Expression (6) to normalize the outputs of individual models.
E3	EPSMS-BPI using the raw output of individual models.

In addition to evaluating the performance of EPSMS in different settings we assessed the performance of the best single-model selected by PSMS (i.e., straight PSMS). This result will be useful for evaluating the advantages of EPSMS over PSMS. We also evaluated the performance of the random forest classifier implemented in CLOP [48], which was identified as the best performing model for the data sets we used. The results obtained using random forest were helpful for assessing the advantages of the full model selection strategy compared with a highly competitive classifier. However, identifying the random forest classifier as the best performing model in the classification tasks required a long period of trial and error experiments. Finally, in order to compare the performance of EPSMS with previous studies, we provide the results obtained using manually selected models as reported in Reta et al. [4], which were the previous best results for the data sets we used. Table 5 summarizes the methods compared in our experiments.

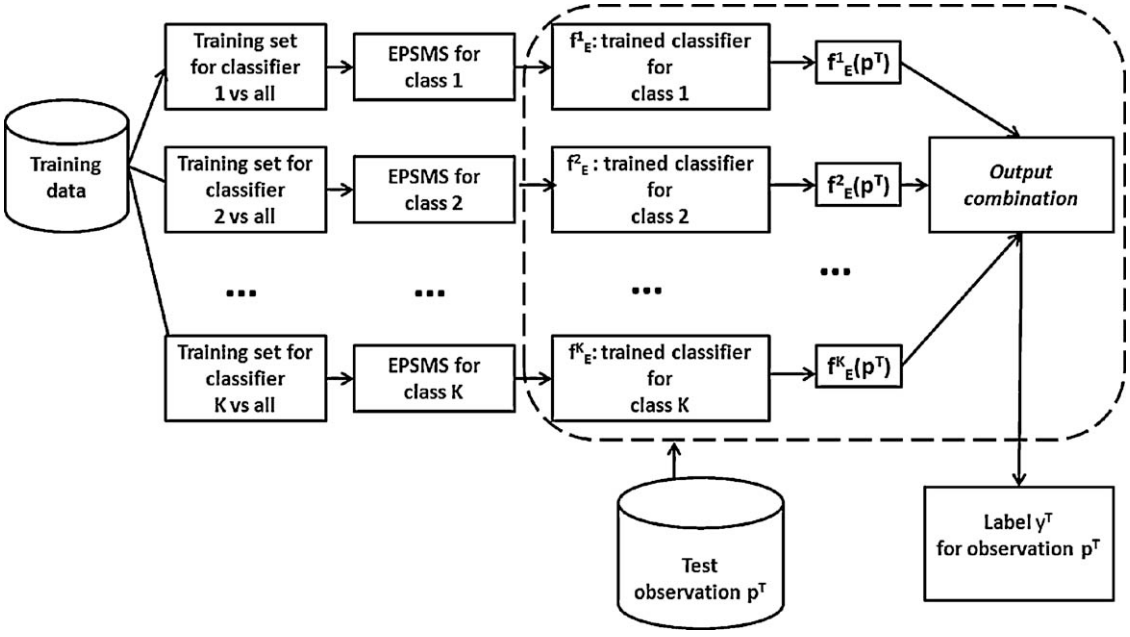

Fig. 4. OVA multiclass classification approach. K (independent) binary classifiers are built, where each can classify the observations in class k_i vs the rest $k_{j:j \neq i}$. A test region $\mathbf{p}^T$ is assigned the label corresponding to the classifier with the highest confidence $f_i(\mathbf{p}^T)$.

Table 6
Average AUC performance for the different type/subtype classification tasks (binary classification). The best result in each row is shown in bold. We report experimental results for different sets of features and those extracted from different image regions for the range of methods, see Section 6.1.

ID	Features	Region	Reference	Baseline	PSMS	E1	E2	E3
ALL vs AML								
1	A	C	89.26	93.63	93.87	**95.14**	**95.14**	92.43
2		N-C	90.65	94.37	95.16	96.06	**96.16**	92.58
3	B	C	79.06	84.38	83.41	**83.43**	83.31	77.82
4		N-C	**81.27**	80.59	79.74	80.32	80.32	76.08
5	C	C	89.92	93.04	93.15	**95.04**	94.87	91.95
6		N-C	92.17	93.17	95.66	96.48	**96.66**	93.77
L1 vs L2								
7	A	C	81.4	91.46	87.68	88.95	88.72	**92.24**
8		N-C	90.69	91.13	88.54	92	92	**94.1**
9	B	C	76.08	79.73	79.07	79.76	79.91	**84.11**
10		N-C	83.67	80.07	74.79	86.8	87.06	**87.41**
11	C	C	82.25	87.85	87.45	89.7	89.71	**92.38**
12		N-C	88.61	88.93	88.06	90.23	90.37	**93.09**
M2 vs M3, M5								
13	A	C	80.45	94.43	91.93	**95.78**	95.13	92.72
14		N-C	95.9	95.58	**96.71**	95.7	95.51	93.01
15	B	C	71.06	73.34	72.99	80.23	**80.33**	72.1
16		N-C	78.93	81.47	79.06	**83.48**	83.44	76.69
17	C	C	84.12	93.2	**95.93**	95.67	95.7	92.85
18		N-C	**94.68**	93.54	93.65	94.24	94.36	89.45
M3 vs M2, M5								
19	A	C	78.82	89.71	88.25	92.01	**92.57**	88.72
20		N-C	87.97	86.88	91.51	**96.62**	96.55	92.28
21	B	C	78.67	74.59	72.93	74.9	75	**79.86**
22		N-C	73.91	72.34	**77.26**	76.4	76.78	76.89
23	C	C	71.01	81.91	92.74	94.02	**94.15**	92.01
24		N-C	89.85	84.25	93.06	**93.24**	92.98	90.44
M5 vs M2, M3								
25	A	C	86.64	92.08	87.32	89.53	89.93	**95.28**
26		N-C	95.52	92.87	92.94	94.54	94.69	**97.68**
27	B	C	73.14	69.87	61.77	66.4	67.73	**75.78**
28		N-C	73.32	69.3	79.24	78.08	78.12	**85.91**
29	C	C	84.98	90.86	92.29	95.85	**96.01**	95.77
30		N-C	**93.54**	92.11	88.97	92.14	92.25	93.07

In the next section we use a Wilcoxon signed-rank test to determine the statistical significance of any performance differences among the methods. This test is recommended for the comparison of classifiers among multiple data sets [49].

6.2. Experimental results

This section reports the experimental results for acute leukemia subtype classification. The goals of this section are:

- to analyze the classification performance of the models selected by EPSMS for the different acute leukemia classification tasks;
- to evaluate the improvement provided by EPSMS compared with PSMS and the baseline classifier;
- to compare the performance of EPSMS using the normalization strategies described in Section 4;
- to compare the performance of the models obtained with EPSMS with those produced in previous studies [4].

Table 6 shows the experimental results with different methods for the different leukemia subtype classification tasks. We show the results for the different sets of features and for the different regions where features were extracted, see Section 5.2.

Table 6 shows that the results obtained using the different techniques were acceptable for most acute leukemia classification tasks. However, EPSMS consistently outperformed the results obtained using the reference, baseline, and straight PSMS methods with the different settings. Only three settings in the results from a previous study [4] outperformed EPSMS, i.e., rows 4, 18 and 30, although the difference in performance was very small. In fact, the differences in performance (across all settings) between the E1, E2, and E3 methods with the reference and baseline methods were statistically significant at the 99% level, where E1, E2 and E3 performed better. The random forest baseline method outperformed the reference method and PSMS in 18 and 16 out of the 30 settings, respectively, which demonstrated the high performance of the baseline method used. The difference between the reference and the baseline method was statistically significant, whereas the difference between PSMS and the baseline was not.

The results shown in Table 6 confirm that fusing the output of the selected models improved the performance of the best individual model selected using PSMS. The differences between PSMS and E1/E2 were statistically significant at the 99% level, whereas the difference between E3 and PSMS was not statistically significant. Thus, it was better to use a normalization strategy for fusing the outputs, rather than using the raw outputs of classifiers. However, the three normalization strategies performed similarly. In fact, the differences in performance among the three EPSMS variants were not statistically significant. The use of E1 and E2 is preferable because E3 did not outperform PSMS. In future work we would like to explore alternative strategies to fuse the outputs of the selected full models.

Table 6 shows that EPSMS delivered no significant difference in performance with the different feature subsets (A, B or C) or when features were extracted from different regions (N and NC). This was not true for the methods used in the reference study [4] where methods built using features from subset B had a limited performance in the different tasks. This is an important result

Table 7
Average AUC performance for the different type/subtype classification tasks (multiclass classification). The best result in each row is shown in bold. In this experiment we used the features subset C, based on results from Table 6.

Measure	Region	Reference	Baseline	PSMS	E1	E2	E3
M1 vs M3 vs M5							
Avg. AUC	C	78.66	92.58	92.37	**93.94**	**93.94**	93.82
Accuracy	C	66.13	82.29	**83.37**	81.32	79.76	78.79
Avg. AUC	N-C	92.80	92.35	92.36	**93.94**	93.92	93.28
Accuracy	N-C	**84.87**	82.87	81.84	81.87	82.34	79.34
L1 vs L2 vs M1 vs M3 vs M5							
Avg. AUC	C	84.03	92.34	91.13	**93.78**	93.76	83.40
Accuracy	C	55.86	74.95	72.86	75.83	**76.06**	74.92
Avg. AUC	N-C	92.33	91.76	90.62	**94.21**	94.09	86.09
Accuracy	N-C	**77.48**	73.81	71.72	74.50	75.65	74.03

Table 8
Members of ensembles generated using EPSMS with the best and worst average performance among the three variants of EPSMS for the results reported in Table 6. Column 2 indicates whether feature selection (FS) was performed before data preprocessing (P). Columns 3–5 show the preprocessing, feature selection and classification techniques used in each individual model.

ID	P/FS	Preprocessing	Feature selection	Classification
Models considered in the ensemble selected with EPSMS that gave the best result				
1	FS	Standardize(1), shift-scale(0)	Pearson(103)	lssvm($c=1;d=1;\gamma=0.4315$;sh = 0.6828;$b=1$)
2	P	Normalize(1), standardize(1), shift-scale(1)	*F*test(4)	logitboost($u=10$;sh = 0.33925;de = 1)
3	P	Normalize(1), shift-scale(1), standardize(1)	*F*test(4)	rf($u=100;m=1;b=1$)
4	P	Normalize(1), shift-scale(1), standard-ize(1)	Relief(65)	lssvm($c=0;d=2;\gamma=2.8358$;sh = 2;$b=1$)
5	–	Normalize(1), shift-scale(1), standard-ize(1)	–	lssvm($c=1;d=1;\gamma=2.0133$;sh = 0.92317;$b=0$)
6	P	Standardize(0), shift-scale(1)	*F*test(17)	rf($u=10;m=4;b=1$)
Models considered in the ensemble selected with EPSMS that gave the worst result				
1	–	Normalize(1)	-	rf($u=100;m=1;b=1$)
2	–	Normalize(1), shift-scale(0)	–	logitboost($u=101;1.77;d=1$)
3	–	Normalize(1), shift-scale(1), standardize(1)	–	logitboost($u=110$;sh = 2;de = 2)
4	P	Normalize(1), shift-scale(1)	*F*test(16)	neural($u=25$;sh = 1.42;$b=1;e=10$)
5	P	Normalize(1), shift-scale(1)	gs(40)	neural($u=25$;sh = 1.14;$b=1;e=10$)
6	–	Standardize(0)	–	rf($u=100;m=2;b=1$)

because we can practically achieve similar accuracy using features extracted from the whole cell, or the nucleus and cytoplasm. This means it is not necessary to perform image segmentation to achieve acceptable classification performance. A final observation from the results shown in Table 6 is that EPSMS did not obtain the best results in the 30 classification tasks but it was the most reliable method with all settings. This confirmed that no single classifier or method guarantees the best classification model for all classification tasks, although EPSMS was a much better choice than the other model selection strategies considered.

The computational complexity of EPSMS/PSMS is difficult to estimate because it depends on the complexities of the methods used during the search process (different methods for classification, feature selection, and preprocessing), which also depend on other factors such as the number of examples or the dimensionality of the problem (see [6] for an extended discussion of the computational complexity of PSMS). During each run of PSMS, a total of $(t_{max}+1)\times m$ solutions were evaluated and each evaluation requires a k-fold cross validation. Therefore, EPSMS/PSMS may be a computationally expensive process for large or high-dimensional data sets. In practice, however, this complexity is manageable. For example, the average processing time for EPSMS/PSMS in the experiments reported in Table 6 was 45.63 min on a laptop with 2 GB of RAM and a Dual Core Processor at 1.5 GHz. This was the average time required for the evaluation of the 10 folds in the cross-validation. Thus, 45.63 min were required for ten runs of EPSMS/PSMS. We should also emphasize that the generation of ensembles using EPSMS does not add to the complexity of PSMS.

Fig. 5 summarizes the results from Table 6, to show the best result obtained using each method in each binary classification task. With the exception of the task M2 vs M3, M5 EPSMS had the best model in all tasks. The improvement over the reference method performance was considerable. It is important to point out that the performance achieved with EPSMS (ranging between 92% and 98%) was similar to results obtained using microarray data or flow cytometer tests (see Section 2), so we consider that our method offers a good tradeoff between low cost and accuracy.

Table 7 shows the results obtained using the different methods in the multiclass classification problems. These results agreed with those reported in Table 6. EPSMS methods outperformed the other techniques in terms of the AUC measure. The improvements over the reference results were more major for the attributes extracted from the *C* region. In terms of accuracy, the reference result

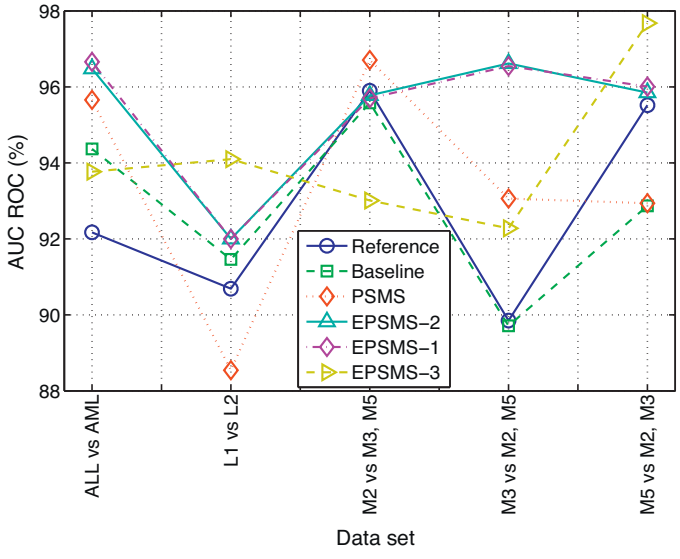

Fig. 5. Best result obtained using each method in each binary classification task.

outperformed E1, E2, and E3 in terms of the features extracted from the *N-C* region, although the differences were rather small.

Table 8 shows the members of two ensembles selected using EPSMS for the results reported in Table 6. In particular, we highlight the ensembles with the worst and the best performance with any EPSMS variant in Table 6. This table shows that different individual models were used for building the ensembles, which differed in terms of preprocessing methods, feature selection techniques, and classification model, while the parameters of each model were different. Thus, this table shows that individual models were diverse when compared with each other. It was interesting that the individual models with the worst performing ensemble did not use any feature selection method or preprocessing method in 4 out the 6 models. In contrast, 5 out of the 6 individual models with the best performing ensemble included preprocessing and feature selection methods. This suggests that the incorporation of preprocessing and feature selection methods is an important factor that affects the performance of the ensemble. It is also interesting that the classifiers with the best ensembles were less complex than those with the worst ensembles (e.g., compare the number of units (u) used for the *logitboost* and *rf* classifiers in the best and worst ensembles).

7. Conclusions

We proposed the application of EPSMS to the problem of acute leukemia classification. The classification of acute leukemia types/subtypes is an important task because it ensures patients receive appropriate treatments. Very effective methods and tests are available for this task, but they are complex and very expensive. Thus, these methods are not available in most developing countries. The morphological classification of acute leukemia, where bone marrow cell images are analyzed, is an inexpensive alternative to these complex methods.

EPSMS is a tool for automatically building ensembles that does not require prior knowledge of the domain or machine learning. EPSMS is a generic technique but it can generate classification models that are specifically designed for each acute leukemia classification task. In this study we proposed improvements for EPSMS and reported experimental results based on real data for the acute leukemia classification task.

The experimental results showed that acceptable performance could be obtained with EPSMS in the classification of acute leukemia subtypes. We found that EPSMS outperformed previous methods with different settings. EPSMS outperformed manually constructed models [4], a strong baseline classifier [48], and PSMS [6]. The results were consistent in different subtype classification tasks, using different features, and different image regions where features were extracted. In addition to its accuracy, another benefit of EPSMS is that no user interaction is required to produce highly effective classification models. Thus, other medical (and non-medical) classification tasks could benefit from EPSMS. Furthermore, the analysis of models selected using EPSMS could provide the analyst with insights into the different classification tasks, which may help to build classification models for other related problems. It is important to emphasize that the classification performance of models selected with EPSMS was very close to that obtained with more accurate, but expensive methodologies. Further, EPSMS achieved similar results when using features extracted from segmented and unsegmented images. This is important because image segmentation is still an open issue in computer vision.

Several future research directions arose throughout the development of this work. In particular, we would like to develop alternative strategies for the selection of individual models and/or for the combination of heterogeneous models with the goal of improving the performance of EPSMS. Also, we would like to develop parallel versions of EPSMS to increase the efficiency of the method and to explore the development of alternative heuristic search methods for the selection of ensemble classifiers. We would like to develop hierarchical classification models based on EPSMS for acute leukemia classification. Finally, we are very interested in applying EPSMS to other medical classification tasks.

Acknowledgements

The first author was supported by PROMEP under grant 103.5/11/4330 and under the UANL-PAICYT program 2010. The authors are grateful with Dr. Rúben Lobato and Dr. José E. Alonso from the Department of Hematology, Mexican Social Security Institute, Puebla, Mexico, for their help in the collection and annotation of samples. A. Rosales and C. Reta thank CONACyT for scholarship nos. 335690 and 212409, respectively. The authors are grateful to the reviewers and editors for their comments, which have helped us to improve significantly the paper.

References

[1] The Leukemia and Lymphoma Society (LLS). Leukemia, lymphoma, myeloma facts 2010–2011; 2011. Online, Available at http://www.lls.org/content/nationalcontent/resourcecenter/freeeducationmaterials/generalcancer/pdf/facts.pdf [accessed 19.10.11].
[2] Bennett J, Catovsky D, Daniel M, Flandrin G, Galton D, Gralnick H, et al. Proposals for the classification of the acute leukaemias. French–American–British (FAB) cooperative group. British Journal of Haematology 1976;33(4):451–8.
[3] Gonzalez JA, Olmos I, Altamirano L, Morales BA, Reta C, Galindo M, et al. Leukemia identification from bone marrow cells images using a machine vision and data mining strategy. Intelligent Data Analysis 2011;15(3):443–62.
[4] Reta C, Altamirano L, Gonzalez JA, Diaz R, Guichard JS. Segmentation of bone marrow cell images for morphological classification of acute leukemia. In: Guesgen HW, Charles Murray R, editors. Proceedings of the twenty-third international Florida artificial intelligence research society conference. Menlo Park, CA: AAAI Press; 2010.
[5] Escalante HJ, Montes M, Sucar E. Ensemble particle swarm model selection. In: Proceedings of the international joint conference on neural networks. Piscataway, NJ, USA: IEEE; 2010. p. 1814–21.
[6] Escalante HJ, Montes M, Sucar E. Particle swarm model selection. Journal of Machine Learning Research 2009;10:405–40.
[7] Gorissen D, Dhaene T, de Turck F. Evolutionary model type selection for global surrogate modeling. Journal of Machine Learning Research 2009;10:2039–78.
[8] Huang CJ, Liao WC. A comparative study of feature selection methods for probabilistic neural networks in cancer classification. In: Proceedings of the IEEE international conference on tools with artificial intelligence. Piscataway, NJ, USA: IEEE; 2003. p. 451–8.
[9] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999;286:531–7.
[10] Slonim DK, Tamayo P, Mesirov JP, Golub TR, Lander ES. Class prediction and discovery using gene expression data. In: Shamir R, Miyano S, Istrail S, Pevzner P, Waterman M, editors. Proceedings of the fourth annual international conference on computational molecular biology. New York, NY, USA: ACM Press; 2000. p. 263–72.
[11] Li Y, Campbell C, Tipping M. Bayesian automatic relevance determination algorithms for classifying gene expression data. Bioinformatics 2002;18(10):1332–9.
[12] Zong N, Adjouadi M, Ayala M. Artificial neural networks approaches for multidimensional classification of acute lymphoblastic leukemia gene expression samples. Transactions on Information Science and Applications, World Scientific and Engineering Academy and Society 2005;2(8):1071–8.
[13] Adjouadi M, Ayala M, Cabrerizo M, Zong N, Lizarraga G, Rossman M. Classification of leukemia blood samples using neural networks. Annals of Biomedical Engineering 2010;38(4):1473–82.
[14] Dietterich T. Ensemble methods in machine learning. In: Kittler J, Roli F, editors. First workshop on multiple classifier systems, LNCS, vol. 1857. Berlin, Heidelberg: Springer-Verlag; 2000. p. 1–15.
[15] Wang W. Some fundamental issues in ensemble methods. In: Proceedings of the international joint conference on neural networks. Piscataway, NJ, USA: IEEE; 2008. p. 2244–51.
[16] Kuncheva L, Whitaker C. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 2003;51:181–207.
[17] Bian S, Wang W. On diversity and accuracy of homogeneous and heterogeneous ensembles. International Journal of Hybrid Intelligent Systems 2007;4:103–28.
[18] Freund Y, Schapire R. Experiments with a new boosting algorithm. In: Saitta L, editor. Proceedings of the thirteenth international conference on machine learning. San Francisco, CA, USA: Morgan Kaufmann; 1996. p. 148–56.

[19] Breiman L. Bagging prediction. Machine Learning 1996;14:123–40.
[20] Breiman L. Random forests. Machine Learning 2001;45(1):5–32.
[21] Kuncheva L. Cluster and selection method for classifier combination. In: Howlett RJ, Jain LC, editors. Fourth international conference on knowledge-based intelligent information engineering systems and allied technologies. Piscataway, NJ, USA: IEEE; 2000. p. 185–8.
[22] Giacinto G, Roli F. Methods for dynamic classifier selection. In: Proceedings of the 10th international conference on image analysis and processing. Piscataway, NJ, USA: IEEE; 1999. p. 659–64.
[23] Wichard JD, Christian M, Ogorzalek M. Building ensembles with heterogeneous models. In: 7th Course on the International School on Neural Nets IIASS; 2002. http://www.j-wichard.de/publications/salerno_lncs_2003.pdf [accessed 31.01.12].
[24] Park C, Cho S. Evolutionary computation for optimal ensemble classifier in lymphoma cancer classification. In: Zhong N, Ras ZW, Tsumoto S, Suzuki E, editors. Foundations of intelligent systems, Proceedings of the 14th international symposium, LNCS, vol. 2871. Berlin, Heidelberg: Springer-Verlag; 2003. p. 521–30.
[25] Macas M, Ruta D, Gabrys B, Lhotska L. Particle swarm optimization of multiple classifier systems. In: Sandoval F, Prieto A, Cabestany J, Gra na M, editors. Proceedings of the 9th international work conference on artificial neural networks, LNCS, vol. 4507. Berlin, Heidelberg: Springer-Verlag; 2007. p. 333–40.
[26] Yang L, Qin Z. Combining classifiers with particle swarms. In: Wang L, Chen K, Ong Y-S, editors. Advances in natural computation, first international conference, LNCS, vol. 3611. Berlin, Heidelberg: Springer-Verlag; 2005. p. 756–63.
[27] Giacinto G, Roli F. An approach to the automatic design of multiple classifier systems. Pattern Recognition Letters 2001;22(1):25–33.
[28] Ruta D, Gabrys B. Classifier selection for majority voting. Information Fusion 2005;6(1):63–81.
[29] Escalante HJ, Montes M, Sucar E. PSMS for neural networks on the IJCNN 2007 agnostic vs prior knowledge challenge. In: Proceedings of the international joint conference on neural networks. Piscataway, NJ, USA: IEEE; 2007. p. 1191–7.
[30] Saffari A, Guyon I. Quickstart guide for CLOP. Technical report. Graz University of Technology and Clopinet, May; 2006. http://www.ymer.org/research/files/clop/QuickStartV1.0.pdf [accessed 31.01.12].
[31] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the international conference on neural networks, vol. IV. Piscataway, NJ, USA: IEEE; 1995. p. 1942–8.
[32] Engelbrecht AP. Fundamentals of computational swarm intelligence. Hoboken, NJ, USA: John Wiley and Sons; 2005.
[33] Angeline PJ. Evolutionary optimization vs particle swarm optimization: philosophy and performance differences. In: Porto VW, Saravanana N, Waagen D, Eiben AE, editors. Proceedings of the 7th conference on evolutionary programming, LNCS, vol. 1447. Berlin, Heidelberg: Springer-Verlag; 1998. p. 601–10.
[34] van den Bergh F. An analysis of particle swarm optimizers. PhD thesis. University of Pretoria, Sudafrica; 2001.
[35] Cawley G, Talbot N. On overfitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research 2010;11:2079–107.
[36] Escalante HJ, Montes M, Villase nor L. Particle swarm model selection for authorship verification. In: Bayro-Corrochano E, Eklundh J-O, editors. Proceedings of the 14th Iberoamerican congress on pattern recognition, LNCS, vol. 5856. Berlin, Heidelberg: Springer-Verlag; 2009. p. 563–70.
[37] Escalante HJ, Montes M, Sucar LE. An energy-based model for image annotation and retrieval. Computer Vision and Image Understanding 2011;115(6):787–803.
[38] Guyon I, Saffari A, Dror G, Gavin Cawley. Analysis of the IJCNN 2007 competition agnostic learning vs. prior knowledge. Neural Networks 2008;21(2–3):544–50.
[39] Bradley A. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 1997;30(5):1145–59.
[40] Guyon I, Gunn S, Nikravesh M, Zadeh L. Feature extraction, foundations and applications. In: Series studies in fuzziness and soft computing. Berlin, Heidelberg: Springer-Verlag; 2006.
[41] Morales B, Olmos I, Gonzalez JA, Altamirano L, Alonso J, Lobato R. Bone marrow smears digitalization. Puebla, Mexico: Laboratory of Specialities of the Mexican Social Security Institute; 2005.
[42] Lopez ED, Altamirano L. A method based on tree structured Markov random field and a texture energy function for classification of remote sensing images. In: Proceedings of the 5th international conference on electrical engineering, computing science and automatic control. Piscataway, NJ, USA: IEEE; 2008. p. 540–4.
[43] Reta C. Segmentation and classification of leukemia cells from contextual information in digital images. Master's thesis. National Institute of Astrophysics, Optics and Electronics, Puebla, Mexico; 2009 (Spanish).
[44] Rifkin R, Klautau A. In defense of one-vs-all classification. Journal of Machine Learning Research 2004;5:101–41.
[45] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference and prediction. In: Springer series in statistics. Berlin, Heidelberg: Springer-Verlag; 2006.
[46] Wang LP, Chu F, Xie W. Accurate cancer classification using expressions of very few genes. IEEE Transactions on Computational Biology and Bioinformatics 2007;4(1):40–53.
[47] Ambroise C, McLachlan GJ. Selection bias in gene-expression data. Proceedings of the National Academy of Sciences 2002;99:6562–6.
[48] Dahinden C. Classification with tree-based ensembles applied to the WCCI 2006 performance prediction challenge datasets. In: Proceedings of the 19th international joint conference on neural networks. IEEE; 2006. p. 1669–72.
[49] Demsar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 2006;7:1–30.