JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

GA(M)E-QSAR: A Novel, Fully Automatic Genetic-Algorithm-(Meta)-Ensembles Approach for Binary Classification in Ligand-Based Drug Design

Yunierkis Pérez-Castillo,*[,†,‡,§] Cosmin Lazar,[†] Jonatan Taminau,[†] Mathy Froeyen,[§] Miguel Ángel Cabrera-Pérez,[‡,∥] and Ann Nowé*[,†]

[†]Computational Modeling Lab (CoMo), Department of Computer Sciences, Faculty of Sciences, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium

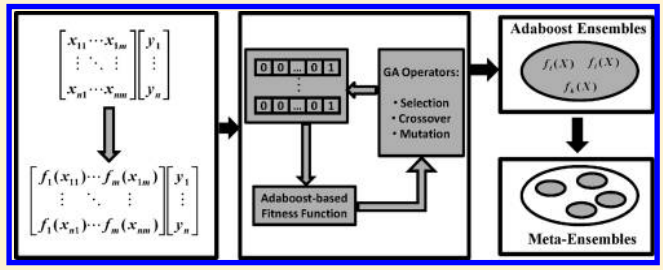
[‡]Molecular Simulations and Drug Design Group, Centro de Bioactivos Químicos, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba

[§]Laboratory for Medicinal Chemistry, Rega Institute for Medical Research, Katholieke Universiteit Leuven, Minderbroedersstraat 10, B-3000 Leuven, Belgium

[∥]Engineering Department, Pharmacy and Pharmaceutical Technology Area, Faculty of Pharmacy, University Miguel Hernandez, Alicante 03550, Spain

Supporting Information

ABSTRACT: Computer-aided drug design has become an important component of the drug discovery process. Despite the advances in this field, there is not a unique modeling approach that can be successfully applied to solve the whole range of problems faced during QSAR modeling. Feature selection and ensemble modeling are active areas of research in ligand-based drug design. Here we introduce the GA(M)E-QSAR algorithm that combines the search and optimization capabilities of Genetic Algorithms with the simplicity of the Adaboost ensemble-based classification algorithm to solve binary classification problems. We also explore the usefulness of Meta-Ensembles trained with Adaboost and Voting schemes to further improve the accuracy, generalization, and robustness of the optimal Adaboost Single Ensemble derived from the Genetic Algorithm optimization. We evaluated the performance of our algorithm using five data sets from the literature and found that it is capable of yielding similar or better classification results to what has been reported for these data sets with a higher enrichment of active compounds relative to the whole actives subset when only the most active chemicals are considered. More important, we compared our methodology with state of the art feature selection and classification approaches and found that it can provide highly accurate, robust, and generalizable models. In the case of the Adaboost Ensembles derived from the Genetic Algorithm search, the final models are quite simple since they consist of a weighted sum of the output of single feature classifiers. Furthermore, the Adaboost scores can be used as ranking criterion to prioritize chemicals for synthesis and biological evaluation after virtual screening experiments.



INTRODUCTION

Computer-aided drug design has become an important component of the drug discovery process. Both ligand and structure-based modeling techniques are now used to save time and money along the drug discovery pipeline.[1−3] In this sense, and among the ligand-based drug design methods, Quantitative Structure−Activity Relationships (QSAR) studies are used to correlate the chemical properties of molecules with their biological activity. This modeling technique involves data collection, calculation of molecular descriptors, selection of the molecular descriptors that are relevant to explain the observed structure−activity relationship, construction of the models, and their validation. Successful stories where the predictions of QSAR models have been experimentally corroborated can be found elsewhere.[4−7]

Despite the achievements of the ligand-based drug design methodologies, several hurdles remain. Many researchers have focused their attention on the main factors behind the failure of QSAR models such as the way the data set for modeling is selected and prepared, the activity cliffs in the structure−activity relationship, the process of training and selecting a model, the validation of the models, and the overestimation of the internal cross-validation parameters.[8−12]

Received: March 15, 2012
Published: August 2, 2012

In recent years, many artificial intelligence algorithms have been used to solve classification and regression problems in the context of QSAR modeling. Among the classification and regression approaches that have been widely used in ligand-based drug design are clustering algorithms such as *k*-NN, decision trees, bayesian methods, partial least-squares, artificial neural networks, and support vector machines.[13−21] On the other hand, in a regular QSAR study, the data set contains a few training cases and hundreds of features. This means the data set contains redundant or meaningless information that, if used to train a model, will lead to overfitted models that lack generalization capabilities. To overcome this difficulty, feature selection techniques are employed to select those features relevant to describe the observed structure−activity relationships. It has also been shown that machine learning optimization algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Artificial Immune System (AIS), and Ant Colony Optimization (ACO) are very powerful in QSAR modeling for feature selection.[22−32] The wide range of application of machine learning techniques to QSAR modeling has been extensively reviewed elsewhere.[33−36]

Among all these machine learning approaches, Genetic Algorithms (GA) have been widely used in QSAR modeling. This is a stochastic and heuristic optimization algorithm that belongs to the family of Evolutionary Algorithms, which refers to a class of stochastic, parallel search heuristics inspired by the biological model of evolution. The robustness of genetic algorithms has made possible to apply them with success to many problems.[37,38]

The idea behind the algorithm is to evolve a population of candidate solutions, each represented as a chromosome, toward a global objective, formulated by means of a fitness function. Throughout this simulated evolution the population will be subjected to different steps. These steps, which involve a fair amount of stochastic operations due the biological processes they are based on, will generate a new population of candidate solutions. The whole process is reiterated until a specified criterion is met. A typical Genetic Algorithm starts with a randomly generated population. Every individual of the population is evaluated with respect to an objective score or fitness function. Based on this score, several individuals are selected (they can be seen as the potential solutions worthy to further explore) for the next generation. In order to introduce enough variation, genetic operators like mutation and crossover are applied next. This combination of reproducibility by selection and modification by mutation and crossover is the basis of the exploration-exploitation mechanism of the Genetic Algorithm.

Despite the success in the application of machine learning algorithms to QSAR modeling, these strategies are not exempt of problems. In a recent study of Huang and Fan,[39] they propose that the vast number of equivalent models to be chosen and the insufficient validation strategies are primarily responsible for the failure of many QSAR models. They also show that combining models using voting systems can provide better performance than the classical single model approaches.

In this last respect, ensemble models have become popular during the last years in QSAR modeling.[40−43] This type of methodologies attempt to increase the coverage of the chemical space of the models by combining diverse models into an ensemble with higher accuracy and predictability. The effectiveness of the ensemble modeling approaches when a machine learning algorithm is used to explore the structure−activity landscape is closely related to the stochastic nature of many of these computational algorithms that trend to get trapped in a local minimum of a search space that contains multiple equivalent solutions.

Among all ensemble techniques, one of the most popular is Adaboost. This algorithm was first proposed in 1997 by Freund and Schapire.[44] The success of this algorithm comes from two main facts: 1) it is simple and 2) it adapts to correctly classify misclassified data samples at each iteration. This is achieved by assigning a weight to each training sample that is updated at each round of the algorithm in a way that correctly classified samples get lower weights, while misclassified ones get higher weights. Therefore, misclassified samples are more effective in the selection of the ensemble member on the next Adaboost iteration since they will have a higher impact in the determination of the overall error function. However, Adaboost is a sequential forward search algorithm using a greedy selection strategy. Because of this, the found ensemble and coefficients are not optimal. To overcome this drawback of the algorithm, several researchers have proposed to use Adaboost in combination with other search methods such as Genetic Algorithms to explore several combinations of weak classifiers. Specifically, the combination of the efficient search capabilities of Genetic Algorithms with the effectiveness and simplicity of Adaboost has previously been used to solve classification problems in different research applications such as image processing, pattern recognition, information filtering, and general feature selection problems.[45−51] However, no application of this combination of Adaboost with Genetic Algorithms has been reported in the context of QSAR modeling.

Here we present a novel, fully automatic algorithm named GA(M)E-QSAR based on the combination of genetic algorithms, Adaboost and Ensemble Voting for QSAR modeling. The GA is used to explore the feature space with the aim of selecting those features relevant for modeling the properties under investigation. The fitness function used inside the GA is based on the Adaboost algorithm and takes as input the single feature classifiers trained before starting the GA evolution. In this way, the GA searches for an ensemble of single feature classifiers in the form of a linear combination of them that considerably outperforms each individual ensemble member's accuracy. Due to the stochastic nature of the GA, running the algorithm from a different initial population each time yields models with similar statistical parameters but distant in the descriptors space. We study how these equivalent solutions can be combined in a Meta-Ensemble by using both Adaboost and Voting Ensembles to improve the performance of individual ensembles. To guarantee diversity on the Meta-Ensemble members, we use a cluster-based approach to select them. We tested our algorithm using five data sets extracted from the literature and show that with the proposed methodology we can obtain results comparable with what has been previously reported for these data sets. Furthermore, we show that the methodology here proposed is able to yield models as accurate, robust, and generalizable as those obtained with state of the art feature selection and classification algorithms. We also show that the models here developed are able to reduce the chances of misclassification of the active compounds for the most active chemicals in a virtual screening scenario.

COMPUTATIONAL METHODS

Data Sets and Molecular Descriptors. To validate our methodology, we selected five data sets of chemical compounds from the literature that have been previously used for binary

dx.doi.org/10.1021/ci300146h | J. Chem. Inf. Model. 2012, 52, 2366−2386

classification. Four of these data sets were previously compiled and used for classification studies by Sutherland et al. and consist of cyclooxygenase-2 (COX2) inhibitors, benzodiazepine receptor (BZR) ligands, estrogen receptor (ER) ligands, and dihydrofolate reductase (DHFR) inhibitors.[52] These data sets have been widely used to validate methodologies and algorithms for QSAR, virtual screening, and features selection.[53−59] We also used for modeling the data set of 531 molecules compiled by Fourches et al. that contains chemicals related to the induction of drug-induced liver injury (DILI).[60] For each data set, we used the same class assignments provided on the original citations.

The BZR data set was assembled from a set of 406 compounds mainly from one lab. In vitro binding affinities as measured by inhibition of [3H] diazepam binding are expressed as IC50 values, ranging from 0.34 nM to >70 μM. A threshold of 7.0 for pIC_{50} was selected to split the data set into the actives and inactives groups. This is a heterogeneous data set that mainly consists of benzodiazepine derivates such as benzodiazepinone, thiazolbendiazepine, pyrazol-[e]-diazepine, benzo-triazolodiazepine, and benzo-imidazodiazepine derivates. The data set also contains benzoazepin, benzopirydine, pyrrolopyrazin, pyrrolpyrimidine, pyridine, pyrizopiperydine, pyrazopyridine, carbazole, and pyridoindole derivates.

The COX-2 data set was assembled from the published work of a single research group and contained inhibitors with activities ranging from 1 nM to >100 μM. This resulted in the compilation of 467 chemicals for which a threshold of $pIC_{50} = 6.5$ was used for classification. This data set is also noncongeneric and mainly includes pyrrole, pyrazole, cyclopentene, bipnenyl, imidazole, spyro compounds, ozazole, thiophene, thiazol, tiazole, pyrimidine, and benzopyrane derivates.

The DHFR data set was assembled from a collection of 756 inhibitors of the dihydrofolate reductase enzyme tested on the same lab. The ranges of activities against *P. carinii* DHFR was between 0.034 nM to >1000 μM. The threshold for classification was set to $pIC_{50} = 6.0$. This is a heterogeneous collection mainly composed of pyrimidine, triazine, pyrido pyrimidine, pteradine, quinazoline, and pyrrolo pyrimidine derivates.

The ER data set was assembled from a compilation of binding affinities to the estrogen receptor for 616 nonredundant compounds prepared by the National Toxicology Program at the National Institute of Environmental Health Sciences. The data have been reported using the relative binding affinity (RBA) scale, which measures affinities with respect to β-estradiol. An additional subset of 393 chemicals was also compiled from the literature. This is also a heterogeneous data set for which the activity threshold RBA = 1 (after rounding to the nearest integer) was selected for designating compounds as actives or inactives. This value can be used for toxicological prioritization rather than pharmaceutical screening. The ER data set is composed, among several types of chemical, by steroids, phenol, anilide, pyrethroids, azo compound, polycyclic aromatic, organochlorine, triazine, stilbene, indene, indone, flavones, biphenyl, netahexestrols, hexestrols, and indole derivates.

For the four BZR, COX2, DHFR, and ER data sets, in the source reference they used a "cherry picking" with a maximum dissimilarity algorithm to assign 40% of the samples to the external test set while the remaining 60% were reserved for the training set. The splitting of the data into the training and external sets was preceded by a filtering step where redundant information was discarded, and a subset of the original data above a certain similarity threshold was kept. This resulted in Training/External sets composed of 181/125, 178/125, 233/160, and 266/180 chemicals for the BZR, COX2, DHFR, and ER data sets respectively. During our research, the composition of the external set for these data sets was the same as in the original paper, while the training data they used was randomly split into five different disjointing subsets. The objective of this data splitting procedure is to test the algorithm with different compositions of the training (80% of data) and selection subsets (20% of data). The training set was used to train the algorithm, while the selection set is reserved for the model validation and selection steps. The external subset was used to evaluate the generalization of the final model. In consequence, for each modeling replica, one of these disjointing subsets was used as selection data, while the remaining four subsets were used as training data.

To collect the DILI data set, the authors employed a combination of lexical and linguistic tools to extract relationships that exist between any therapeutic compound and a range of liver pathologies and hepatic physiological observations. This data mining procedure was carried out for article abstracts from the MEDLINE database. The chemicals identified during the data mining process were subject of a data curation step that resulted in 951 compounds that were classified as inducing liver injury in humans, rodents, and nonrodents. During the group assignment procedure it was also taken into account that some drugs were found to induce liver injury in more than one species simultaneously or in only one of the three species. For QSAR studies the compounds inducing liver injury in humans only (248) and in rodents only (238) were selected as the two classes. The DILI data set is a very heterogeneous one and consists of multiple QSARs since its end point property should be driven by multiple potential target interactions. However, a deeper analysis of this data set, or the rest of them used to validate the classification methodology here proposed, including a breakdown by congeneric subsets is out of the scope of our investigation, and we focus on the overall performance of our methodology.

To model the DILI data set we followed the same procedure used on the original reference.[60] The whole data set was randomly split into 5 disjoint subsets, and each of them was considered as external data on five independent algorithm evaluations. At each of the five independent evaluations, the remaining nonexternal four subsets were randomly split into training (80%) and selection (20%) subsets.

For each data set all the 1D and 2D molecular descriptors from the DRAGON v.6[61] software were calculated, 3763 total, and from all pairs of descriptors with a correlation greater than 0.9 only one was kept. Constant variables were also removed.

The GA(M)E-QSAR Algorithm. The GA(M)E-QSAR algorithm takes as input a $N \times M$ matrix of N training samples and M raw features [Raw Features: Unmodified molecular descriptors.], a $N \times 1$ training class assignment vector, a $N' \times M$ matrix of N' selection samples, a $N' \times 1$ selection class assignment vector as well as a $N'' \times M$ matrix of N'' external samples and a $N'' \times 1$ external class assignment vector. The algorithm returns individual Adaboost Ensembles [Adaboost Ensemble: Ensemble of single feature LDA models trained using the Adaboost algorithm] of LDA models [LDA models: Linear Discriminant Analysis models. To establish the LDA classifier, a linear function is calculated for each class. The class function yielding the highest score represents the predicted class. In the GA(M)E-QSAR algorithm one LDA model per raw feature is trained, and a filtered subset of them is used as the input for the GA.] and four Meta-Ensembles [Meta-Ensembles: Ensembles of

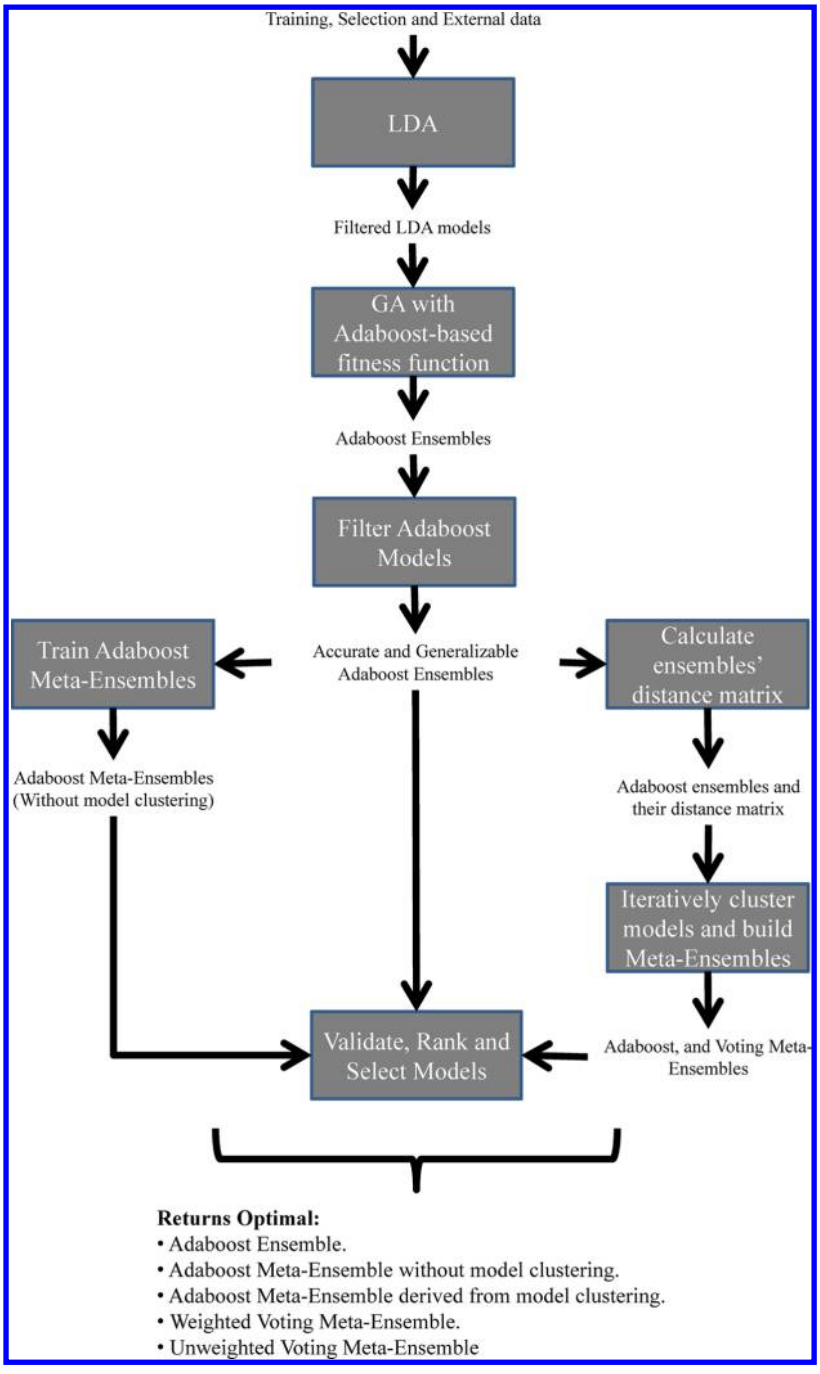


Figure 1. Workflow of the GA(M)E-QSAR algorithm.

Adaboost Ensembles obtained using either the Adaboost algorithm or Voting schemes, and which members are Adaboost Ensembles resulting from the Genetic Algorithm optimization.] (which members are the previous individual Adaboost Ensembles) that are obtained with and without model clustering. One Adaboost Meta-Ensemble is build using the Adaboost Ensembles without model clustering in an iterative way as will be explained later. In a separate step of the algorithm, a model clustering procedure is performed to select representative Adaboost Ensembles that are used to build Meta-Ensembles based on Adaboost and Voting schemes (Weighted and Unweighted). The optimal Adaboost Ensemble and Meta-Ensembles are selected using a consensus ranking approach that returns models that combine accuracy, robustness, and generalization capabilities. The overall workflow for the proposed algorithm is shown in Figure 1, and its general pseudocode is presented in Chart 1. For a more clear presentation of the algorithm, in Chart 1 we separated the pseudocode of each algorithm block of Figure 1.

As seen from Figure 1 the algorithm takes as input three matrices of training, selection, and external validation data as well as the group assignment for each data sample on each set. We train LDA models with this data and filter them. The filtered LDA models are used to train an Adaboost classifier using a GA that searches for Adaboost Ensembles of single feature LDA models. These Adaboost Ensembles are then evaluated for their performance on the training and selection data subsets to discard low performance models. The remaining ensembles after

dx.doi.org/10.1021/ci300146h | J. Chem. Inf. Model. 2012, 52, 2366−2386

Chart 1. GA(M)E-QSAR Algorithm Pseudocode

```
Require: N samples x M features training data matrix, N x 1 training class assignment vector; N' samples x M features test samples, N' x 1 test class assignment vector and N'' samples x M features external samples, N'' x 1 external class assignment vector
//
// LDA block
//
1. Train one Linear Discriminant Analysis (LDA) model per feature using the training data
2. Discard random LDA models (Sensitivity < 0.5 or Specificity < 0.5)
//
// GA block
//
For i=1:GA_iter
3.  Initialize population
4.  Score initial population
    For j=1:max_GA_gen
5.  Select Elite offspring
6.  Select parents for crossover and mutation
7.  Create the new population = Elite children + Xover children + Mutate children
8.  Score the new population
9.  End
10. End
//
// Filter Adaboost Models block
//
11. Discard duplicated individuals (Keep unique ensembles)
12. Discard ensembles with Acc. Train<mean(Acc. Train) and Acc. Test<mean(Acc. Test)
//
// Calculate models' distance matrix block
//
13. Calculate the models' distance matrix
//
//Iteratively cluster models and build Meta-Ensembles block
//
14. For i=1:n_clusters
15.  Select i representative ensembles.
16.  Aggregate the i models in an Adaboost Meta-Ensemble
17.  Aggregate the i models in a Weighted Voting Meta-Ensemble
18.  Aggregate the i models in a Unweighted Voting Meta-Ensemble
19. End
//
//Train Meta-Adaboost models block
//
20. For i=1:n (for practical reasons we set n=n_clusters)
21.  Aggregate i models in an Adaboost Meta-Ensemble
22. End
//
//Validate, Rank and Select Models block
//
23. For i=1:No. filtered Adaboost Ensembles
24.  LOO and Bootstrap validation of the i-th Adaboost Ensemble
25. End
26. Rank Adaboost Ensembles considering accuracy, robustness and generalization.
27. For i=1:n_clusters
28.  LOO and Bootstrap validation of the i-th Adaboost Meta-Ensemble obtained without model clustering.
29.  LOO and Bootstrap validation of the i-th Adaboost Meta-Ensemble derived from model clustering
30. End
31. Rank Adaboost Meta-Ensembles considering accuracy, robustness and generalization.
32. Rank Voting Meta-Ensembles considering accuracy and generalization.
33. Return Optimal Adaboost Ensemble and evaluate performance on external data set
34. Return Optimal Adaboost Meta-Ensemble and evaluate performance on external data set
35. Return Optimal Voting Meta-Ensemble and evaluate performance on external data set
```

filtering are used to build new Adaboost Meta-Ensembles using the classical greedy search strategy the Adaboost algorithm is based on and to iteratively build Voting and Adaboost Meta-Ensembles using the representative models derived from model clustering. Finally, a validation and selection process is carried out to select the optimal model of each type: Single Adaboost Ensemble, Adaboost Meta-Ensemble without model clustering, Adaboost Meta-Ensemble derived from model clustering, and Weighted and Unweighted Voting Meta-Ensembles. In the next subsections we detail the steps involved at each component of the algorithm as shown in Figure 1. The whole algorithm was implemented in MATLAB R2009a.[62]

● *LDA Block.* The algorithm starts by training one LDA classifier per input raw feature using the training data set and

dx.doi.org/10.1021/ci300146h | J. Chem. Inf. Model. 2012, 52, 2366−2386

Chart 2. GA(M)E-QSAR Fitness Function for Modified GA

//
// *10-fold cross-validated Adaboost-based fitness function*
//

1. **For i=1:10**
2. Train an adaboost model (see **Chart 3**) using the training samples not left out (all samples not in the i^{th} subset)
3. Predict the left-out training cases (samples in the i^{th} subset)
4. **End**
5. Calculate the cross-validated training error based on the prediction of the left-out samples
6. **Return AIC = log(σ^2) + #used_LDA_models/#trainig_cases ; being** $\sigma = 1 - \sqrt{Sensitivity \cdot Specificity}$
7. **End**

Chart 3. Adaboost Pseudocode

Require: N x T matrix of the predictions of N samples using T classifiers.

1. Initialize weights $w_{1,i} = 1/N$
2. **For t=1:T**
3. Normalize weights $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
4. **For each classifier j**
5. Evaluate the weighted error of each classifier $\varepsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$, where $h_j(x_i)$ is the classification result of sample x_i using the classifier h_j and y_i is the observed classification of sample x_i.
6. **End**
7. Choose the classifier h_t with the lowest error $\varepsilon_t = \min(\varepsilon_j)$
8. Update weights $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$, where $e_i = 0$ if sample x_i is correctly classified, $e_i = 1$ otherwise and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
9. **End**
10. **Return the final Adaboost Ensemble:**

$$h(x) = \begin{cases} 1 & if \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}, \text{ where } \alpha_t = \log \frac{1}{\beta_t}$$

saving the predicted classification of the training, selection, and external subsets. Then the random classifiers (those with either Sensitivity [Correctly classified positives/Total positives] or Specificity [Correctly classified negatives/Total negatives] on the training set lower than 0.5) are removed. This yields a set of nonrandom LDA models that are the input for the GA-based search of the Adaboost Ensembles.

●. *GA Block.* Nonrandom single feature classifiers are used to train the GA that will minimize an Adaboost-based fitness function. The GA is run 25 times, each one with a different initial population and for 1000 generations. One of the advantages of this algorithm is that single feature classifiers are computed only once before running the GA, and this guarantees a high computational efficiency since the Adaboost algorithm is very simple as we will show below.

For the GA, we use a bit-string encoding of the chromosomes. A **1** at the i-th position means that the i-th LDA model will be considered by the Adaboost-based fitness function, while **0** means that this feature will be excluded from fitness computation. The initial population is randomly created in such a way that each individual contains at most 10 **1**-coded genes and the length of each individual is set equal to the number of input LDA models.

The selection function is set to a tournament of size 2, while the crossover operator randomly combines each position of two parents to produce an offspring. In brief, the tournament selection consists of the random selection of n individuals from the population, compare their fitness values and select the one with the lowest fitness as the winner one. The mutation process randomly selects a **1**-coded position on the parent and sets it to **0** and in the same way sets to **1** one of the excluded LDA models. The population size is set to 100 individuals, the crossover and mutation rates are set to 0.7 and 0.3 respectively, and two elite offspring automatically survive and pass to the next generation. The overfitting problem was controlled in two ways inside the fitness function of our modified GA: by using a 10-fold cross-validation training scheme and by minimizing the Akaike Index (AIC)[63] that aims to keep a balance between the fitting of the model and the number of ensemble members. The Adaboost Ensemble is trained using the nonrandom, weak single feature classifiers coded on each individual. The pseudocode for this fitness function is shown in Chart 2.

The 10-fold cross-validation strategy is used with the objective to prevent overfitting and slow down the algorithm's speed of convergence toward a local minimum. The Akaike's index AIC is used to measure the fitness of the individual to obtain ensembles with a good trade-off between the fitting and the number of LDA models. In this sense, among two individuals with the same cross-validated error value, the one encoding less LDA models will be better scored and hence will have better opportunities to reproduce and survive during the evolution.

In the GA(M)E-QSAR algorithm, Adaboost is used in the fitness function of the GA and to combine the solutions of multiple GA runs in a Meta-Ensemble. At each iteration, the Adaboost algorithm minimizes the weighted error on the training set and returns an ensemble of single feature LDA models. The

dx.doi.org/10.1021/ci300146h | J. Chem. Inf. Model. 2012, 52, 2366−2386

weighted error of this ensemble is computed and used to update the weights of each training sample. The effect of the changes in the weights of the training samples is to put more weight on the samples misclassified during the previous iteration and reduce the weight of the samples that were correctly classified. The aim of this weights update scheme is to select in the next round a weak classifier that correctly classifies previously misclassified samples, and in consequence, at each round, Adaboost tries to solve more difficult classification problems. After the weights of each classifier are determined, the final classification of each sample is calculated according to the linear combination of the product of these weights and each classifier output. A precomputed classification threshold is used to assign the groups membership. It is worth noting that for virtual screening experiments, these scores can be used to rank positively predicted compounds for synthesis and assay prioritization. The pseudocode of the Adaboost algorithm is shown in Chart 3.

●. *Filter Adaboost Models Block.* After all the 25 GA runs are completed, we put all the unique individuals (each one encoding for an Adaboost Ensemble) from all GA runs together, remove duplicated individuals, and evaluate the accuracy on both the training and the selection subsets of the Adaboost Ensemble encoded by each of these unique individuals. Next, we remove the individuals that encode Adaboost Ensembles with accuracy lower than the mean accuracy value on both the training and selection sets. A model with low accuracy on the training set can be a consequence of a low performance individual that was produced by a mutation during the last GA generation.

On the other hand, the prediction of the selection data subset gives an estimation of the generalization capabilities of the models and will discard Adaboost Ensembles that cannot be generalized to new data. This filter step based on the accuracy in predicting both the training and selection sets allows us to choose for further cross-validation and Meta-Ensembles construction only models combining good performance on the training data as well as on the selection subset that was not used during the learning process. The result of this filtering step is a number of Adaboost Ensembles solutions from which the optimal Adaboost Ensemble will be chosen based on a consensus ranking approach that is described in the next sections.

●. *Train Meta-Adaboost Models Block.* As we discussed in the Introduction of the paper, due to the stochastic nature of many machine learning algorithms such as the GA and to the high number of existing equivalent suboptimal solutions, the algorithm can converge to good enough solutions rather than the global minimum each time it is run. To combine the solutions from different GA runs into Meta-Ensembles we used Voting and Adaboost Ensembles. We should stress that the construction of a Meta-Ensemble consists in the combination of a subset of Adaboost Ensembles derived from the GA search into a new solution (Meta-Ensemble) using either Voting or Adaboost Ensembles.

At this step of the algorithm, we investigate how the Adaboost algorithm (see Chart 3) can combine the solutions found during the GA search into Meta-Ensembles using its classical greedy search strategy. In this case, the inputs to the Adaboost algorithm are the predictions made by all the solutions (Adaboost Ensembles) derived from the GA search. We iteratively changed the number of cycles of the Adaboost algorithm from 2 to 50 in order to obtain Adaboost Meta-Ensembles of variable size.

●. *Calculate Model's Distance Matrix Block.* Since one of the conditions that a good ensemble should satisfy is the diversity of its members, we carried out a clustering of the models in order to select a set of representative and diverse Meta-Ensemble members. The first step to produce the clusters of models is to calculate the pairwise distance matrix between the models. Here we used a distance criterion proposed by Todeschini et al. that takes into account the correlation of original variables within and between models and allows the discovery of clusters of similar models.[64] We stress the fact that the computation of the model distances is performed on the original variables space. This means that for Adaboost Ensembles A and B, the distance metric is computed from the raw features that correspond to the individual LDA models in A and B.

This distance metric for two models A and B is defined as $d^2_{A,B} = a + b - 2r_{AB}$, where a is the number of features encoded by model A and not by model B, b is the number of features encoded by model B but not by model A, and r_{AB} accounts for the intra- and intercorrelation between the two subsets of features. This correlation coefficient is obtained from the non-negative eigenvalues λ_i of the modified cross-correlation matrix of the two subsets of features $Q = C_{AB}{\cdot}C_{BA}$, where C_{AB} and C_{BA} are the correlation matrix of subset A with subset B and the correlation matrix of subset B with subset A respectively. Finally the correlation coefficient r_{AB} is calculated as $r_{AB} = \sum(\lambda_i)^{1/2}$.

●. *Iteratively Cluster Models and Build Meta-Ensembles Block.* The distance matrix obtained above is then used to cluster the models using agglomerative hierarchical clustering. To cluster the models we used the average linkage method where the distance between any two clusters A and B is taken to be the average of all distances between pairs of objects ″x″ in A and ″y″ in B. The cluster member with the lowest sum of distances to the rest of the cluster members is selected as the representative model of that cluster (the cluster centroid). In the GA(M)E-QSAR algorithm we iteratively change the number of Meta-Ensemble members from 2 to 50, and for each number of model clusters we built Adaboost (described above) and Voting Meta-Ensembles.

One of the issues that can be faced when Voting Meta-Ensembles are constructed is how to weight the individual vote of each member. For example, if there is an Unweighted Voting Meta-Ensemble which members are A and B, and Adaboost Ensemble A classifies sample "x" in group **0** (inactive group) and Adaboost Ensemble B classifies it in group **1** (active group), this sample will not be assigned to any of both groups. For this reason, we implemented the classical Voting scheme where the winner class takes a whole vote from each Adaboost Ensemble and where we only considered an odd number of Adaboost Ensemble members.

Let us now suppose that model A has Specificity Sp = 0.7, while model B has Sensitivity Se = 0.9; this scenario raises a simple question: How reliable is the prediction of each model to consider their votes equally important. It is obvious that the classification of sample "x" by model B in the active group is more reliable than the classification as inactive provided by model A. Considering this, we implemented a Weighted Voting Meta-Ensemble approach that calculates the sum of the Sensitivity of all models that classify the sample in the active group and the Specificity of all models that classify the sample in the inactive group. If the aggregated Sensitivity is higher than the aggregated Specificity, then the sample is classified in the active group, else the sample is assigned the inactive group.

●. *Validate, Rank, and Select Models Block.* Once we have trained the Single Adaboost Ensembles as well as the Meta-Ensembles build using the Voting and Adaboost strategies as described above, they are subject to Leave-One-Out (LOO)[65]

and 1000 cycles of Bootstrap[66] cross-validation in order to evaluate their robustness. Following the concept that the ideal model should combine fitting, predictability, and robustness, we considered the accuracy on predicting the training subset an estimator of the fitting of the model; the accuracy on the selection set as a descriptor of the generalization capabilities of the model and the LOO and Bootstrap cross-validation accuracies as estimators of the robustness of the model. To retrieve one model from the whole pool that combines good fitting, generalization capabilities, and robustness we used the Borda-Kendall consensus ranking approach.[67]

Under this approach, all the models are ranked according to each of the four decision makers: accuracy on the training set, accuracy on the selection set, LOO accuracy, and Bootstrap accuracy. Following this procedure, a rank value is assigned to each model according to each decision maker, and these individual ranks are sum to obtain its overall ranking. Although not optimal, this approach is effective to solve ranking problems and avoid the high computational cost associated with stricter consensus ranking methodologies such as those proposed by Kemeny and Snell,[68] Bogart,[69] and Cook et al.[70] The final optimal (Meta)Ensemble is then the one with the lowest Borda-Kendall consensus ranking. The only additional consideration is that for the Meta-Ensembles, in the case when two of them have equal Decision Maker values, the one with the lower number of members takes the lowest ranking.

This validation and selection strategy was applied to each of the five types of models we developed: Adaboost Ensembles, Adaboost Meta-Ensembles without model clustering, Adaboost Meta-Ensembles trained with the representative models of the clustering, Weighted Voting Meta-Ensembles, and Unweighted Voting Meta-Ensembles. Finally, the top ranked model according to each modeling strategy is selected.

Support Vector Machines. To train the Support Vector Machine (SVM) models, each feature was scaled to mean 0 and standard deviation 1. The Radial Basis Function (RBF) kernel was selected, and no feature selection was performed. The SVM models were trained using the MATLAB implementation of the LIBSVM package.[71]

The SVM parameters C and γ were optimized through a grid search using a 10-fold cross-validation of the training data set. This grid search was split into two parts; we first performed a coarse grid search to find a pair (C_c, γ_c), and then, in a second search, we used a finer grid search around this point to obtain the optimal C and γ values. During the coarse grid search, C was exponentially varied between 2^{-5} and 2^{15} with a grow factor of 2^2, while γ was changed using the same strategy in the interval from 2^{-15} to 2^3. Once C_c and γ_c are found, the fine grid search explored the region enclosed by the intervals C_c*2^{-2}:C_c*2^2 and γ_c*2^{-2}:γ_c*2^2 with a grow factor of $2^{0.25}$.

Once the optimal C and γ parameters were obtained, they were used to train a SVM model using the whole training data. Afterward, this model was subject to Leave-One-Out and Boostrap cross-validation following the same procedure that we used for the Adaboost Ensembles obtained with the GA(M)E-QSAR algorithm. Finally this SVM was used to predict both the selection and external data sets.

Least Squares Support Vector Machines. The Least Squares Support Vector Machines (LS-SVM) models were developed with the LS-SVMlab Toolbox for MATLAB.[72] Each feature was scaled to mean 0 and standard deviation 1, the Radial Basis Function (RBF) kernel was selected, and no feature selection was performed. The γ and σ^2 parameters of the LS-SVM models were optimized using a combination of Coupled Simulated Annealing (CSA) and grid search as implemented on the LS-SVMlab Toolbox for MATLAB. This parameters optimization stage was guided by the misclassification rate of the 10-fold cross-validated training data. The CSA algorithm was first used to find good starting values of the kernel parameters, and then the grid search strategy was performed for the fine-tuning of them.

Once the optimal γ and σ^2 parameters were obtained, they were used to train a LS-SVM model using the whole training data. Afterward, this model was subject to Leave-One-Out and Boostrap cross-validation following the same procedure as for the Adaboost Ensembles obtained with the GA(M)E-QSAR algorithm. Finally this LS-SVM was used to predict both the selection and external data sets.

Genetic Algorithm-SVM and Genetic Algorithm-LS-SVM Wrappers. The genetic operators for the Genetic Algorithm-SVM (GA-SVM) and Genetic Algorithm-LS-SVM (GA-LS-SVM) wrappers were the same used for the GA(M)E-QSAR algorithm, and the selection of the optimal model was also performed following the same consensus ranking approach. The only difference between the GA-Adaboost and the GA-SVM and GA-LS-SVM wrappers was the use of Adaboost and SVM/LS-SVM classifiers respectively in the fitness function of the GA. The SVM and LS-SVM models were trained using the MATLAB implementation of the LIBSVM and LS-SVM packages respectively.[71,72]

The parameters C and γ for the SVM and γ and σ^2 for the LS-SVM models were optimized using the whole training data following the same procedure described above for the SVM and LS-SVM models without feature selection before the GA evolution started. These parameters were kept fixed during the GA search.

Feature Selection Using Bagged Trees. We trained an ensemble of 100 classification trees where each tree was grown using an independent bootstrap sample of the training data set. The prediction error of the ensemble was determined by computing the predictions for each tree on its Out-Of-The-Bag observations, using the majority vote rule to assign each observation prediction and then comparing the predicted response with the observed class. To build the decision trees we used a minimum number of observations per leaf of 1, and a random subset of features equals to the square root of the total number of variables was considered for each split. The nodes split criterion was set to the Gini's diversity index which measure the node's impurity based on the fraction of samples with each class that reaches that node.

To determine the optimal size of the ensemble of decision trees, we followed a consensus ranking procedure analogous to that described for the selection of the optimal models derived from the GA(M)E-QSAR algorithm. The sole difference is that for the bagged trees the decision makers we considered were the accuracy in the prediction of the training set, the accuracy in the prediction of the selection set, and the prediction error for the Out-Of-The-Bag samples. It should be noted that the i-th ensemble is that which members are the trees from 1 to i.

Using this optimal size of the classification trees ensemble, the importance of each feature on this optimal ensemble was calculated as the increase in the classification error if the values of the variable are permuted across all Out-Of-The-Bag samples. We selected the 25 most important features according to this metric for further classifiers training.

Table 1. Summary of LDA Model Filtering

Data set	Training samples	Init. MD[a]	Filter	No. LDA models after filtering				
				Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
BZR	145	675	Se/Sp	189	214	207	202	201
BZR	145	675	Error	550	543	584	580	548
COX2	142	745	Se/Sp	209	242	228	229	207
COX2	142	745	Error	543	532	547	528	525
DHFR	187	702	Se/Sp	188	176	179	201	181
DHFR	187	702	Error	507	481	497	510	509
ER	212	680	Se/Sp	210	189	189	189	200
ER	212	680	Error	440	451	426	419	454
DILI	319	908	Se/Sp	78	63	101	84	82
DILI	319	908	Error	594	587	661	638	674

[a]Number of Molecular Descriptors after running Dragon and discarding all pairs of variables with a pairwise correlation greater than 0.9 as well as constant features.

These 25 top features were transformed to have median 0 and standard deviation 1, and then we iteratively trained Adaboost, SVM, LS-SVM, and Linear Discriminant Analysis (LDA) classifiers changing the number of features considered to build the classifiers from 1 to 25. To build the Adaboost ensembles we first trained a LDA model per variable using the training data set, we excluded the random single feature LDA models, and then we used these nonrandom LDA models to assign the single feature LDA model classification for the training, selection, and external sets. This process lead to 25 models (less for the Adaboost Ensembles) that were trained with different number of features for each of the four classification approaches considered (Adaboost, SVM, LS-SVM, and LDA). For each of the SVM and LS-SVM modeling cycles, we optimized the RBF kernel parameters as described above considering only the features to be used to train the model during this parameters optimization step.

All the built classifiers were cross-validated using LOO and Bootstrap strategies. The selection of the optimal complexity of each type of model was carried out using the same consensus ranking approach described for the GA(M)E-QSAR algorithm that considers the accuracies on the prediction of the training and selection sets and the LOO and Bootstrap accuracies. This feature selection approach using bagged trees was implemented in MATLAB.

Feature Selection Using Feature Ranking. We selected the top 25 features according to the ranking based on the nonparametric Wilcoxon rank-sum test.[73,74] These 25 features were used to conduct the same classifiers training process carried out with the features subset derived from the Bagged Trees. This process lead to four classifiers (Adaboost Ensemble, SVM, LS-SVM, and LDA) trained with features derived from the feature ranking process. The feature selection strategy based on features ranking and the classifiers training and selection processes were implemented in MATLAB.

Determination of the Applicability Domain of the Models. The determination of the Applicability Domain (AD) of the models was based in a descriptors range based method.[75] We first performed a Principal Component Analysis (PCA) of the molecular descriptors included in the model for the training data, and based on these Principal Components (PCs) we built a hyper-rectangle defined by the maximum and minimum values of the transformed features. To avoid the consideration of redundant and meaningless information during the AD determination process, we selected the subset of PCs that explained 99% of the observed variance in either the whole training data, when no feature selection is performed, or the selected features subset when the model was derived using a feature selection algorithm. This PCA filtering step guarantees the built of the lowest dimensional hyper-rectangle based on 99% of the information provided by either the training data set or subset.

Afterward, the selection and external data subsets were projected into the PC transformed space using the same transformations used for the training set. A sample belonging to the selection or external sets was considered to be inside the AD of the model if it was inside the hyper-rectangle previously defined. That is, a compound was inside the AD of the model only if its PC transformed coordinates had values between the minimum and maximum values of the respective training set PC transformed coordinates.

Comparison of Different Classification Approaches. For each of the five modeling folds of each of the five data sets that we used in this study, we tested diverse feature selection and classification algorithms. This process led to 25 different experiments, and for each of such experiments a diverse set of classification methodologies was tested. To determine which classification approach could be considered optimal for each experiment, we performed a consensus ranking procedure similar to the one we described to select the optimal model of each of the feature selection-based classification methods that we tested when more than one classifier is trained.

Since for each experiment the optimal model per modeling approach is already able to classify the training data set with high accuracy, we preferred to focus this selection step on the robustness of the models and their ability to predict unseen data. Furthermore, not all methods are able to reach the same balance in the intergroups classification accuracy. For these reasons, in each experiment we considered as decision makers the accuracy in predicting the selection set, $(Sensitivity*Specificity)^{1/2}$ for the selection set and the LOO and Bootstrap cross-validation accuracies of each of the optimal model per modeling strategy. With these decision makers we performed a Borda-Kendall consensus ranking procedure, and we selected the modeling approach having the lowest consensus ranking value as the optimal modeling approach to use on a particular experiment. One additional consideration during this consensus ranking procedure is that in the case of two classification strategies with the same value on a particular decision maker, the model with the lowest number of features gets the lowest ranking.

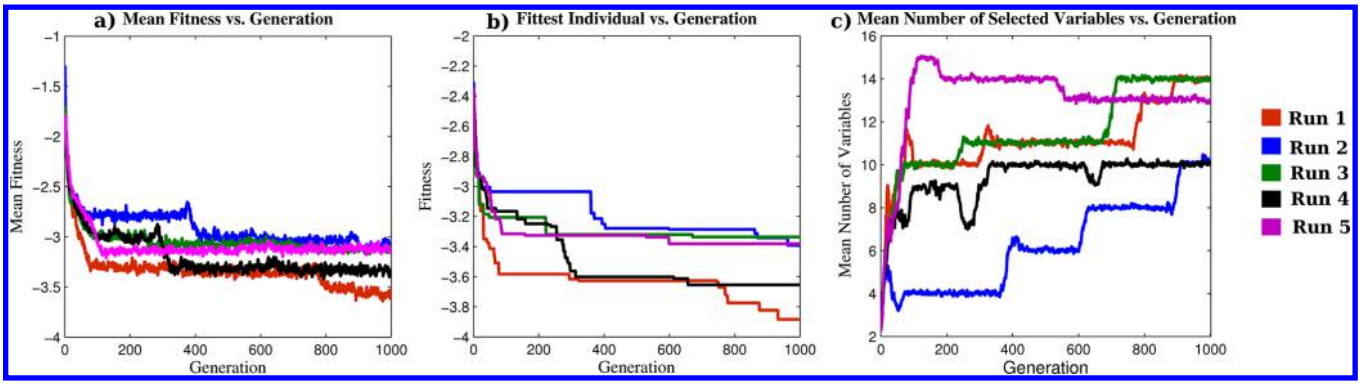


Figure 2. Evolution of the mean fitness (a), the fittest individual (b), and the mean number of selected LDA models in the population (c) during a typical GA evolution.

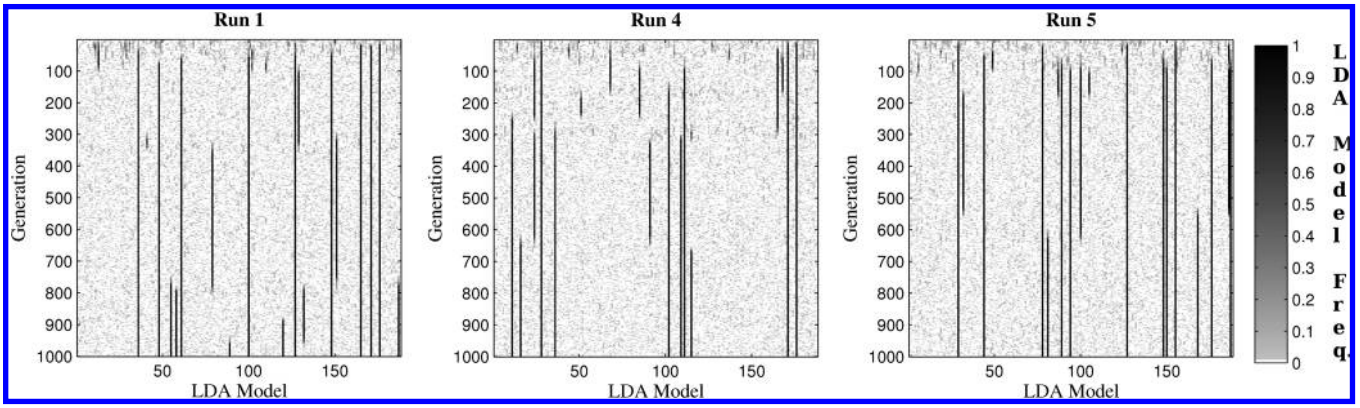


Figure 3. Frequency each LDA model is on the population during the GA evolution.

RESULTS AND DISCUSSION

We ran the GA(M)E-QSAR algorithm for the five folds of the five data sets extracted from the literature described in the Computational Methods section. After the calculation of the Molecular descriptors and the removal of their pairs with correlation greater than 0.9, the single feature LDA models were trained and filtered based on Sensitivity and Specificity to use only those with Se and Sp greater than 0.5 for the GA search. We also studied the scenario where the LDA models are filtered according to their overall error, and those with classification error lower than 0.5 are used to train the GA. During this filtering step the number of LDA models to be considered by the GA to build the Adaboost Ensembles is reduced by 70−74% for the BZR, COX2, DHFR, and ER data sets and by 91% for the DILI data set when the Se/Sp filter is used and by 17−36% when the overall error filter is applied. As expected, the Se/Sp filter is stricter and considerably reduces the number of input LDA models for the GA. As we will show later, despite this reduction in the number of features to be considered by the GA, the Se/Sp and overall error filters yield Adaboost Ensembles with similar performance. Furthermore, the Se/Sp filter provides models with better balance between Se and Sp. The results of the LDA model filtering are summarized in Table 1.

To illustrate the performance of the GA and the most common scenarios we faced, we selected five typical runs for the first data fold of the DHFR data set. In Figure 2 we show how the mean fitness function value (a), the fitness of the best individual (b), and the mean number of selected LDA models (c) evolve along the 1000 generations for the selected case study.

It can be seen in Figure 2 (a) that for all the 5 GA runs analyzed there is a continuous decrease on the average fitness in the population for 100−150 generations indicating that the GA has reached a local minimum. Afterward, the behavior of each population diverges and what happens next is highly dependent on the ability of the GA to move to another minimum or to continue exploring solutions around the current one. Despite the fact that evolutions 3 and 5 are not able to improve the mean fitness value on the population after about 200 generations, they continue exploring new solutions, and fitter individuals can be obtained even when there is no improvement on the mean fitting value as seen from Figure 2 (b). It should also be noted that during the third run of the GA the convergence to a stable fitness value is slower than in the fifth evolution. On the opposite side are GA runs 1, 2, and 4 where after the fast initial stabilization the crossover and mutation operators guide the GA to explore new local minima and higher improvements on the fittest individuals are observed at later generations.

The monitoring of the mean number of selected LDA models in the population illustrated in Figure 2 (c) shows the effectiveness of the AIC-based fitness function we used to keep a balance between the number of selected features and the fitting of the Adaboost Ensembles. In this sense, it can be observed a rapid increase in the mean number of features in the population at early stages of the GA search. The later changes, either increase or decrease, on the mean size of the selected LDA models subsets are closely related to the inclusion of new LDA models that provide nonredundant information to the Adaboost Ensembles. This effect can be better observed when the frequency each LDA model appears in the population at each generation of the GA, shown in Figure 3, is analyzed. For example, during the 250−350 generations of the fourth GA run there is a rapid decrease in the mean population fitness that is also linked to a continuous

dx.doi.org/10.1021/ci300146h | J. Chem. Inf. Model. 2012, 52, 2366−2386

Table 2. Minimum and Maximum Values of the Accuracy Statistics of the Models Obtained Using the GA(M)E-QSAR Methodology When Applied to the Five Data Sets Studied

	Size (Min/Max)[a]	Min(Train)[b]	Max(Train)[c]	Min(Sel)[d]	Max(Sel)[e]	Min(Ext)[f]	Max(Ext)[g]	LOO (Min/Max)[h]	Boot (Min/Max)[i]
				BZR					
A.E	9/18	84 (80/88)	92 (94/91)	72 (70/77)	84 (100/67)	65 (67/63)	70 (71/69)	80/91	77/84
ME-AB	2/15	85 (83/88)	91 (92/90)	69 (70/69)	75 (71/79)	66 (62/69)	70 (71/69)	85/91	81/91
ME-AB NC	2/9	87 (87/87)	92 (92/93)	69 (70/69)	81 (76/84)	69 (63/74)	74 (67/81)	87/91	85/91
ME-Vot W	4/26	81 (76/86)	93 (94/93)	69 (61/85)	84 (95/72)	66 (52/81)	74 (67/82)	-	-
ME-Vot	5/25	80 (75/86)	91 (91/91)	69 (70/69)	78 (76/79)	68 (59/77)	72 (68/76)	-	-
Reported in ref 52		-	79 (81/76)	-	-	-	75 (70/81)	-	-
				COX2					
A.E	6/11	87 (89/86)	90 (90/90)	72 (77/70)	89 (75/96)	63 (65/62)	74 (76/72)	86/89	79/84
ME-AB	1/46	87 (91/85)	91 (93/89)	71 (56/84)	89 (75/96)	64 (67/62)	74 (76/72)	87/91	83/90
ME-AB NC	2/2	88 (93/85)	92 (91/93)	69 (69/68)	89 (75/96)	63 (65/62)	74 (76/72)	83/92	87/91
ME-Vot W	5/11	90 (87/92)	93 (97/90)	72 (77/70)	92 (83/96)	66 (73/61)	74 (82/68)	-	-
ME-Vot	9/39	90 (90/90)	92 (91/92)	77 (69/84)	92 (83/96)	64 (71/59)	72 (78/68)	-	-
Reported in ref 52		-	85 (83/87)	-	-	-	73 (75/72)	-	-
				DHFR					
A.E	9/15	83 (84/83)	88 (89/87)	70 (58/79)	85 (81/87)	68 (64/69)	76 (69/78)	80/84	77/80
ME-AB	4/15	85 (82/86)	88 (89/87)	64 (53/71)	85 (81/87)	70 (62/73)	74 (69/76)	83/88	81/86
ME-AB NC	2/4	83 (84/83)	88 (89/87)	64 (53/71)	85 (79/88)	71 (57/75)	76 (69/78)	83/88	83/87
ME-Vot W	6/17	84 (86/84)	88 (84/91)	68 (53/79)	87 (81/90)	73 (71/73)	76 (71/78)	-	-
ME-Vot	9/29	85 (86/85)	88 (85/90)	66 (53/75)	87 (81/90)	71 (62/74)	74 (74/74)	-	-
Reported in ref 52		-	82 (85/81)	-	-	-	72 (74/71)	-	-
				ER					
A.E	8/15	86 (89/84)	89 (93/85)	75 (68/79)	85 (86/84)	74 (72/76)	78 (79/78)	82/89	80/84
ME-AB	3/20	86 (88/85)	90 (91/89)	72 (79/68)	85 (89/80)	75 (72/77)	78 (79/78)	86/90	84/89
ME-AB NC	2/2	87 (90/85)	90 (91/89)	72 (79/68)	85 (89/80)	73 (72/73)	77 (72/81)	87/90	86/89
ME-Vot W	6/15	87 (93/83)	90 (93/88)	72 (79/68)	91 (96/84)	76 (76/75)	79 (75/83)	-	-
ME-Vot	3/37	87 (91/85)	91 (91/90)	74 (79/71)	89 (96/80)	75 (77/73)	78 (76/80)	-	-
Reported in ref 52		-	81 (87/76)	-	-	-	79 (77/80)	-	-
				DILI					
A.E	9/12	69 (66/72)	75 (76/74)	63 (60/67)	70 (68/72)	57 (55/58)	68 (60/75)	69/73	62/68
ME-AB	2/21	70 (65/74)	76 (76/75)	61 (54/69)	67 (55/80)	58 (50/64)	66 (66/66)	70/76	69/74
ME-AB NC	2/2	70 (65/74)	76 (76/75)	60 (58/63)	67 (55/80)	55 (48/61)	70 (62/77)	70/76	69/75
ME-Vot W	4/19	70 (69/71)	76 (78/73)	62 (50/75)	69 (64/74)	58 (60/58)	69 (74/64)		
ME-Vot	3/39	69 (69/69)	76 (78/73)	61 (52/71)	71 (68/74)	58 (60/58)	69 (56/80)		
Reported in ref 60 (DRAGON Descriptors)		−78	94			−56	71		

[a]Minimum and maximum size of the (Meta)Ensembles. For the Adaboost Ensembles, it refers to the number of single feature LDA models that Adaboost Ensemble contains. In the case of the Meta-Ensembles, the size refers to the number of Adaboost Ensembles it is composed of. [b,c]Minimum and maximum accuracy in the prediction of the training data set respectively represented as Accuracy(Sensitivity/Specificity). [d,e]Minimum and maximum accuracy in the prediction of the selection data set respectively represented as Accuracy(Sensitivity/Specificity). [f,g]Minimum and maximum accuracy in the prediction of the external data set respectively represented as Accuracy(Sensitivity/Specificity). [h]Maximum and minimum LOO cross-validation accuracy represented as Minimum/Maximum. [i]Maximum and minimum Bootstrap cross-validation accuracy represented as Minimum/Maximum.

improvement on the fitness value of the best individual during this evolution interval and to a reduction and then increase in the mean number of selected LDA models in the population. If now we look at Figure 3, it can be noted that during this interval of the GA evolution 4, features that were present in almost all individuals were removed, and 6 new features were added and became dominant in the population. The overall result of this process is a change in the regions the GA exploration takes place and the improvement of the current solutions that is the goal of the GA search.

Similar conclusions can be derived for GA evolutions 1 and 5. In the first case, the improvement on the population fitting during the last 250 evolutions is related to the removal of 2 LDA models that were dominant until that moment and the addition of 5 new LDA models that become dominant by the end of the evolution. In contrast to GA evolutions 1 and 4, during the fifth GA run and after the first 150 generations, some LDA models become dominant and remain in this state until the end of the evolution with the sole addition of few new variables until generation 600 when one dominant LDA model is removed and two new ones are added to the dominant LDA models subset. This LDA models addition is reflected in a slight improvement on the quality of the population mean fitness without moving the evolution away of the local minimum the GA was already exploring. These examples show how the stochastic nature of the GA combined with the existence of multiple equivalent solutions on different local minima lead to different ensembles that have similar fitness values being selected.

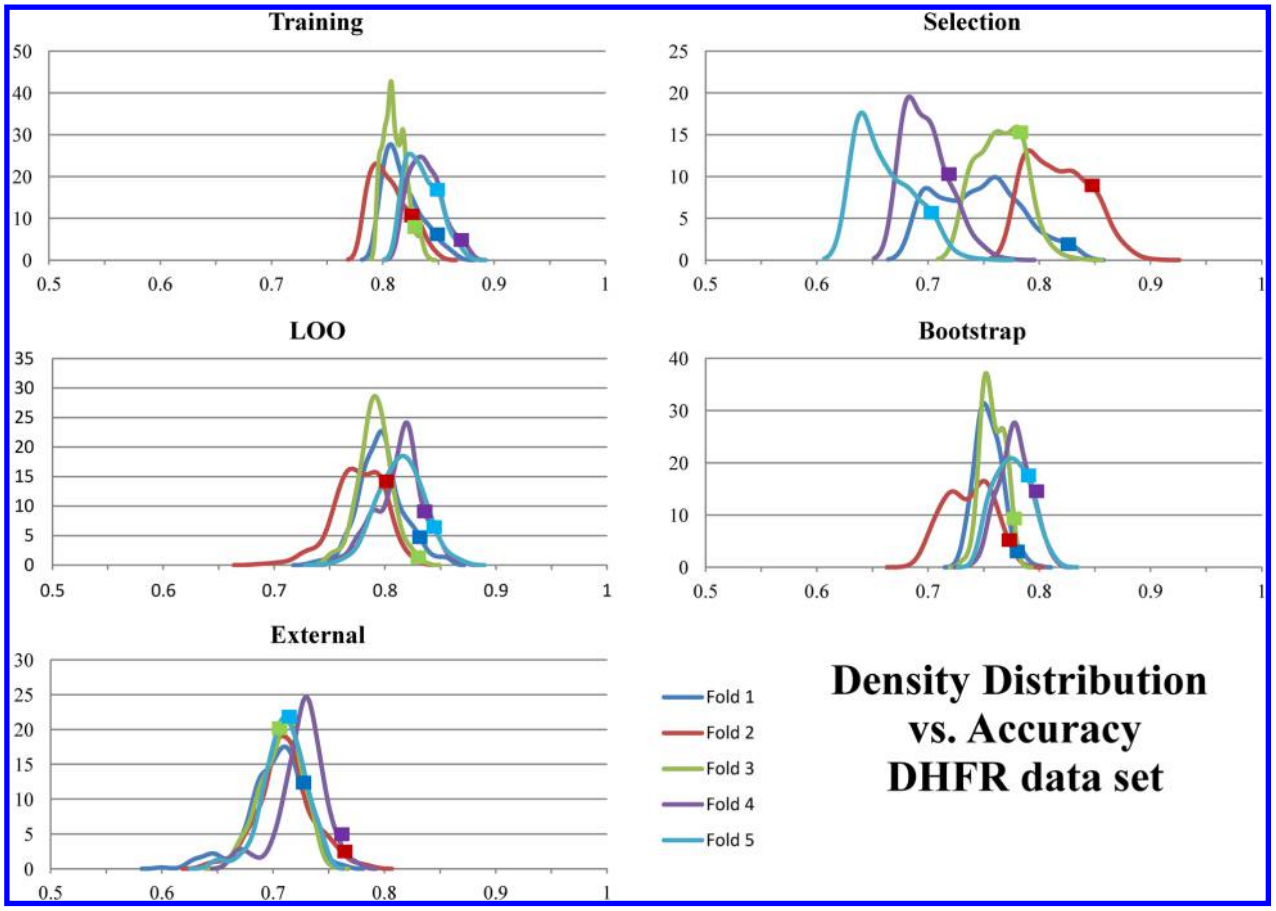


Figure 4. Density distribution estimation vs accuracy for the four decision makers used to rank and select the optimal Adaboost Ensemble. Here we represent the five data folds we used for the DHFR data set.

We report the extensive results obtained for all the five modeling folds of each data set as Supporting Information in Tables TS1 to TS5. In Table 2 we show a summary of these results that includes the following: the minimum and maximum values of the accuracy estimators of the optimal models we obtain with the GA(M)E-QSAR methodology across the five splitting folds of each data set. The results are summarized for the following: the optimal Adaboost Ensemble (A.E), the clustering derived Adaboost Meta-Ensemble (ME-AB), the Adaboost Meta-Ensemble trained using the classic greedy search strategy without model clustering (ME-AB NC), the Weighted Votes Meta-Ensemble (ME-Vot W), and the Unweighted Votes Meta-Ensemble (ME-Vot). In each column we present the minimum and maximum values for the following: (i) the size of the (Meta)Ensemble (**Size**); (ii) the accuracy, sensitivity, and specificity on the training, selection, and external data subsets shown as **Accuracy** (**Se**/**Sp**) in the columns identified by **Train**, **Sel.**, and **Ext.** respectively; and (iii) the minimum and maximum values obtained from the Leave-One-Out (**LOO**) and Bootstrap (**Boot**) cross-validations. Since the model selection process is carried out according to the classifier's global accuracy, the minimum and maximum values of Se and Sp are those corresponding to the models that have the worst and the best overall accuracies respectively.

It is also necessary to note that the reference classification accuracies we use for comparison are those of the optimal model reported on the original references. In contrast to our approach, in the case of the BZR, COX2, DHFR, and ER data sets, they use the external test set to select their optimal prediction models. On the other side, in our algorithm the optimal model selection process is based only on the training and selection data sets, and the external test subset is only used to evaluate the final generalization capabilities of this optimal model as has been proposed by Tropsha.[9]

From Table 2 it can be seen that there are fluctuations on the accuracy values obtained for each modeling fold when the optimal models selected using the procedure described in the Computational Methods section are analyzed. These results show that for the COX2 and DHFR data sets, the maximum accuracy in predicting the external data subset with the optimal Adaboost Ensemble outperforms the results previously reported; while for the BZR, ER, and DILI data sets the maximum accuracy achieved by the proposed methodology is close to what was reported in refs 52 and 60. The Meta-Ensemble approaches are also able to improve or perform as good as the previous modeling studies carried out on these data sets. If we now analyze the values of the mean performance of the individual Ensembles and Meta-Ensembles members (available in the Supporting Information section Tables TS1 to TS5), it can be seen that combining single variable LDA models in an Adaboost Ensemble is an effective strategy to build accurate and robust models from weak classifiers. Since the molecular descriptors that we used to validate the methodology here proposed differ from those used to model the BZR, COX2, DHFR, and ER data sets in the original reference, we also compared the performance of the Adaboost Ensembles trained using the GA search with some state of the art feature selection and classification techniques. This comparative study is shown later.

dx.doi.org/10.1021/ci300146h | *J. Chem. Inf. Model.* 2012, 52, 2366−2386

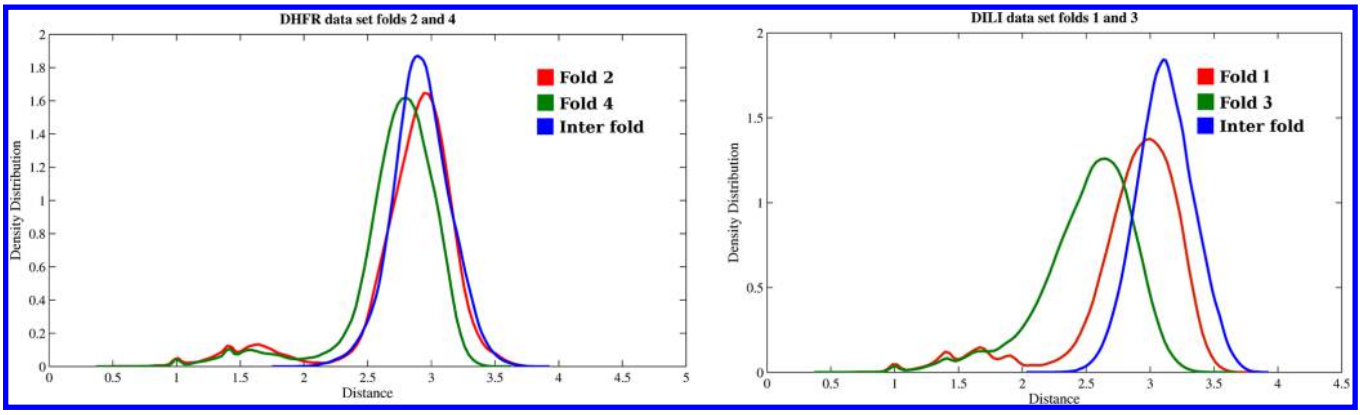


Figure 5. Comparison of the density distribution of distances between two data folds for DHFR and DILI data sets.

It is worth noting that in almost every trained model there is a difference between the accuracy in the prediction of the training and external data sets around 10−20%. If the heuristic rule between the number of adjustable parameters and the number of training samples is analyzed ((*#train samples*)/(*#adjustable parameters* + 1) > 4)), it can be seen that this relation is fulfilled by all the models and hence according to this rule there is a low probability of chance correlation. It should be taked into account that we combined in the GA search two strategies that have been demonstrated to be effective in controlling overfitting: the minimization of AIC and the use of 10-fold cross-validation. Besides, the accuracies achieved in predicting the external data sets are away from being random. Based on these observations, we consider that this difference between the prediction accuracies of the training and external sets can be regarded as a normal consequence of using one model to predict unseen data.

We compared the performance of the GA(M)E-QSAR algorithm when the single feature LDA models are filtered according to their overall accuracy instead of Sensitivity and Specificity and the classification error is used to compute AIC during the GA evolution. In the Supporting Information Table TS6 we provide the maximum and minimum values of the accuracy estimation parameters of the optimal models we obtain with the GA(M)E-QSAR methodology across the five modeling folds of each data set when the overall accuracy filter is used. The comparison of the results obtained with both filters reveals that, despite the minimum and the maximum accuracy of the models are slightly higher for the accuracy filter, the Se and Sp statistics are more unbalanced. This leads to models that, although having a better overall accuracy, are not suitable for virtual screening experiments since the results will be biased toward a particular class and hence there will exist the risk of having many false positives/negatives predictions. In the rest of the paper we focus our analysis in the results obtained when the Se/Sp filter is applied to the single feature LDA models.

The difference between the best and worst values of accuracy in predicting the training data for the Adaboost Ensembles across the five data folds is lower than the same difference for the selection and external data subsets in all the studied cases. It is also worth noting that there is less divergence in the prediction of the external data subset than in the prediction of the selection data for all the five data sets but the DILI one. To study whether these intermodeling folds fluctuations are a consequence of either the composition of the data subset used to train the GA, the stochastic nature of the GA, or the model ranking and selection process; we examined the density distribution estimation of the four decision makers we used to rank the Adaboost Ensembles, for all possible solutions resulting after the Adaboost Ensembles filtering step (see the Computational Methods Section) as well as that of the external data set predictions. In Figure 4 we summarize the model selection step for the optimal Adaboost Ensemble in the five folds of the DHFR data set; the square markers indicate the value of each decision maker for the top-ranked model. The equivalent figures for the other four data sets we used in this study are included in the Supporting Information section (FS1, FS2, FS3, and FS4), and the conclusions derived for the DHFR data set can be generalized to the remaining data sets.

From Figure 4 it is clear that the density distribution estimation of the accuracy in predicting the training sets is very similar for all the five data folds we investigated. As discussed before, by the end of the GA evolution it trends to explore solutions around a certain local minimum that can be different for each of the 25 times it is run. This together with the Adaboost Ensembles filtering step leads to models which prediction accuracies are constrained to a narrow range since the goal of the GA is to enrich the population with accurate solutions. Despite this being true, if the composition of the training subset influences the accuracy of the developed models, we would expect narrow distributions of it but with less overlapping than observed in Figure 4. This can be a consequence of the 60% overlapping in the training sets composition between any pair of data folds and the existence of statistical equivalent solutions despite training subsets composition. To fully clarify the influence of the data sets composition and the stochastic nature of the GA on the modeling process, we analyzed the density distribution estimation of the models' distance for two case studies: data folds 2 and 4 of the DHFR and 1 and 3 of the DILI data sets respectively. These data folds of these two data sets were selected because they are the ones showing more divergent interfold training accuracy distributions as shown in Figure 4 and FS4. Furthermore, in contrast to the DHFR data set behavior, the overlapping between the training set accuracy of folds 1 and 3 in the DILI data set is almost null (see **FS4**). The obtained density distribution estimations of the models' distance are presented in Figure 5 where the intra- and interfold distance distributions are plotted. It should be mentioned that to determine the density distribution of the interfold distances the intrafold distances pairs are excluded.

As can be seen from Figure 5, there are small peaks in the intrafold distances interval 1−2 that are a consequence of the presence of small groups of almost identical individuals in the population such as those which only differ by a mutated gene. The most interesting observation comes from the fact that the

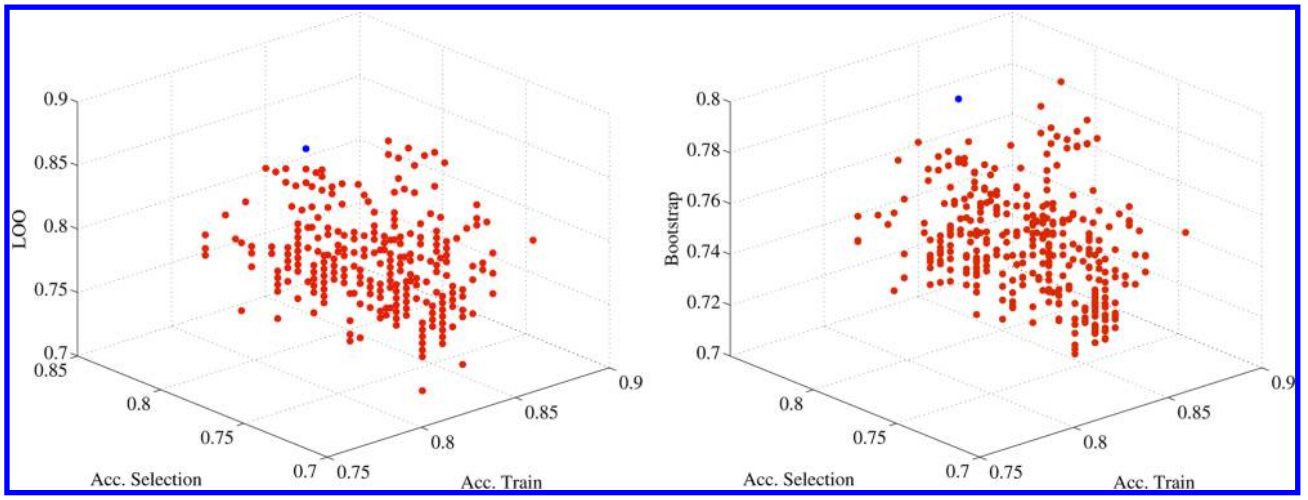


Figure 6. 3D scatter plot of the decision makers used to select the best Single Ensemble in the modeling fold 1 of the DHFR data set. The top-ranked model is highlighted blue.

intra- and interdistances density estimations are almost identical for the DHFR data set and have a high overlapping on the DILI system despite the fact that in this last scenario the accuracies of the prediction of the training set show no overlapping. What we can conclude from the analysis of the density distribution estimation of the accuracies in predicting the training sets and the models distance is that in the case of the BZR, COX2, DHFR, and ER data sets, the combination of the stochastic nature of the GA and the existence of multiple equivalent solutions in the search space is the main reason for the diversity of the solutions the algorithm finds. Furthermore, this diversity is only minimally influenced by the training/selection sets composition. Moreover, if we analyze the modeling folds 1 and 3 of the DILI data set it can be seen that despite the distribution of the accuracies for the training set are completely divergent and both subsets only share 55% of the training compounds, the model diversity show similar inter- and intradistances profiles, although the differences are bigger than in the rest of the data sets. Another factor that is related to the larger spread of the results obtained for the DILI data set is that it is more structurally diverse than the BZR, COX2, DHFR, and ER sets.

On the other hand, there is no overlapping between the selection data subsets, and hence the accuracy that each one is predicted with is dependent on its composition and how well the models from the GA evolutions can fit it. The density distribution for this data subset shows that despite the fact that the models produced by the GA optimization exhibit narrow intervals of accuracies in predicting the training data, not all of them will generalize in the same way. The robustness of the models, measured by the LOO and Bootstrap accuracies, has also a distribution similar to the training set accuracies, and this could suggest a correlation between these three decision makers. This correlation results in a lack of diversity on the decision makers used to rank the models, and hence they will produce biased consensus rankings. Actually this is not the case, as can be seen from the four decision makers correlation matrices provided in the Supporting Information Table TS7, and these decision makers can be used to obtain an appropriate consensus ranking.

If we now look at the models being selected as optimal (highlighted using square markers in Figure 4), we can see that for all the four decision makers, all the selected Adaboost Ensembles have values of the four of them on the tail or the second higher half of the density distribution plot. We also show in Figure 6 two 3D scatter plots of the four decision makers we use for the consensus ranking of the Adaboost Ensemble for the first modeling fold of the DHFR data set to show how this model selection strategy is able to identify a model combining fitting, robustness, and generalization capabilities. This last affirmation can be confirmed when the accuracy of the selected Adaboost Ensemble in predicting the external data subset is evaluated. It can be seen from Figure 4 that all the selected models are capable of predicting the external subset with accuracies higher than the media of its density distribution. It should also be noted the agreement on the distribution of accuracies on the external data set for the BZR, COX2, DHFR, and ER data sets despite the training/selection subsets composition. The same observation is valid for the DILI data folds 1 and 2 and data folds 3 to 5 respectively where at each data fold a different external subset is used. This last observation indicates that our methodology is robust and can produce similar results profiles when also the composition of the external data set that is used to measure the generalization of the selected model changes.

We should remark that the Adaboost Ensembles obtained from the GA optimization are very simple since they are a weighted sum of the output of single feature LDA models. In this way, the influence of each individual feature encoding the LDA models the Adaboost Ensemble is built from can be analyzed, and important information can be derived from it. Moreover, the scores provided by the Adaboost Ensembles can be used to rank the chemicals after a virtual screening experiment and hence aid in prioritizing the candidate molecules for further synthesis and experimental evaluation.

Regarding the performance of the Meta-Ensembles approaches, they tend to produce on average only minor improvements on the minimum and maximum accuracies in the prediction of the external data subsets when compared with the Adaboost Ensembles (see Table 2 and Supporting Information Tables TS1 to TS5). Something to take into account is that the Meta-Adaboost models are more robust than the Adaboost Ensembles as can be observed from the improvement of the LOO and Bootstrap accuracies which can be interpreted as that they could be preferably used. This last observation suggests that the Adaboost Meta-Ensembles are capable of adapting to the loss of information during training and overcome this by adjusting the models weights in such a way that the prediction capability can be maintained when there are less

training samples to learn from. The analysis of the results also reveals that the Adaboost algorithm can produce accurate and robust Meta-Ensembles without the need of a clustering process. One of the causes of the lower performance, in general, of the Adaboost Meta-Ensembles derived from the clustering process when compared to those obtained without clustering is that in the second scenario the most accurate model of the whole pool will be selected in the first Adaboost iteration since it is the one that will provide the lowest weighted error, while in the first one the centroids of each cluster will be combined in the Adaboost Meta-Ensemble and the optimal Adaboost Ensemble in general is not the cluster centroid. Moreover, the size of the Meta-Ensembles is highly variable since it will depend upon the composition of the set of models that are available for their construction as well as upon the balance between the ensembles size, their accuracy, and the degree of redundancy that a Meta-Ensemble of size n shares with another one formed by m > n Adaboost Ensembles. Another interesting conclusion that can be derived from the analysis of the Meta-Ensembles results is that the Weighted Voting scheme provides, in general, slightly better classification results than the Unweighted Voting one. This statement is supported by the fact that in 69% of the experiments Weighted Voting achieves better performance on the external set than the Unweighted Voting Meta-Ensembles and that they are smaller than the Unweighted ones in 67% of the studied classification experiments. Furthermore, for all five data sets the maximum external set classification rate is higher in the Weighted Voting Meta-Ensembles than in the Unweighted Voting Meta-Ensembles.

When a low generalization Adaboost Ensemble is selected as the optimal one after the GA evolution such as in the third modeling fold of the DHFR data set where only 68% of the external subset is correctly classified (see Supporting Information Table TS3), Meta-Ensembles can be particularly useful. In this example the Meta-Adaboost strategies with and without model clustering are able to increase by 3% and 6% respectively the performance of the optimal Adaboost Ensemble while their robustness also increases. The Meta-Voting schemes are also able to increase the Adaboost Ensemble performance on the external set by 5−6%. If we take a closer look at this case study, we can get more insights to the Meta-Ensembles modeling process. In Figure 7 we represent the density distribution estimation for the whole pool of models (**a**), the distances matrix of the representative models of the first 11 clusters (the maximum number of members of any clustering-based Meta-Ensemble) (**b**), and the matrix of distances between the three models selected by the Meta-Adaboost approach without model clustering (**c**).

Two main observations can be derived from Figure 7. First of all, the model clustering process guarantees diversity in the clustering derived Meta-Ensembles since the distance matrix of the representative models show pairwise distances around and greater than the central value of the distances' density distribution. This ensures that the information provided by each Meta-Ensemble member will be complemented by the rest of its members, and hence the construction of Meta-Ensembles with low levels of redundant information will be possible. The second observation is that despite the Adaboost Meta-Ensemble being built without clustering the models, the self-adapting capabilities of the Adaboost algorithm are sufficient for the selection of models that are informatively diverse.

We also investigated the ability of the models here developed in retrieving the most active compounds of the external data set.

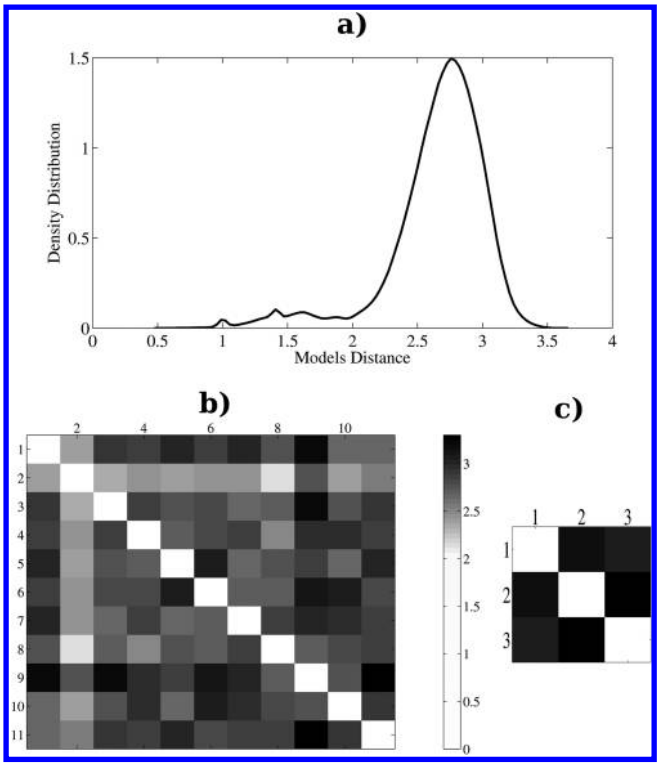


Figure 7. Density distribution (a); distance between first 11 clusters representative models (b); and pairwise distance between the three members of the Meta-Adaboost model obtained without model clustering (c) for the third modeling fold of the DHFR data set.

With that aim, we used the three data sets that have enzymatic inhibitory activity reported: BZR, COX2, and DHFR. In Figure 8 we show a comparison of the global Sensitivity and the Sensitivity when only the 25% most actives compounds of the external data set are considered for the BZR (a), COX2 (b), and DHFR (c) data sets. The numeric values for such comparison are presented in the Supporting Information Table TS8.

As can be seen from Figure 8, the models here developed are able to achieve a better accuracy in predicting the most active compounds present in the external data set than in predicting the whole actives subset in almost every experiment that we analyzed. This indicates that the misclassification probability of an active compound is higher for those with lower inhibitory potency. This observed behavior of the models derived from the GA(M)E-QSAR approach highlights their utility for virtual screening of databases of chemical compounds. It is also interesting to see that in some experiments such as the third and fourth modeling folds of the BZR and DHFR data sets respectively, the Meta-Ensemble-based methods can significantly increase the enrichment of the predicted active compounds with high potency chemicals. In the context of the development of a model for virtual screening, this last observation should also be considered when deciding whether the Adaboost Ensemble derived from the Genetic Algorithm optimization or a Meta-Ensemble is going to be selected as the final virtual screening tool.

Since we used molecular descriptors different from those used in the original source of the BZR, COX2, DHFR, and ER data sets, we compared the Adaboost Ensembles found by our approach with some state of the art feature selection and classification techniques. Given that SVM and LS-SVM classifiers are known to be less sensitive to overfitting than other machine

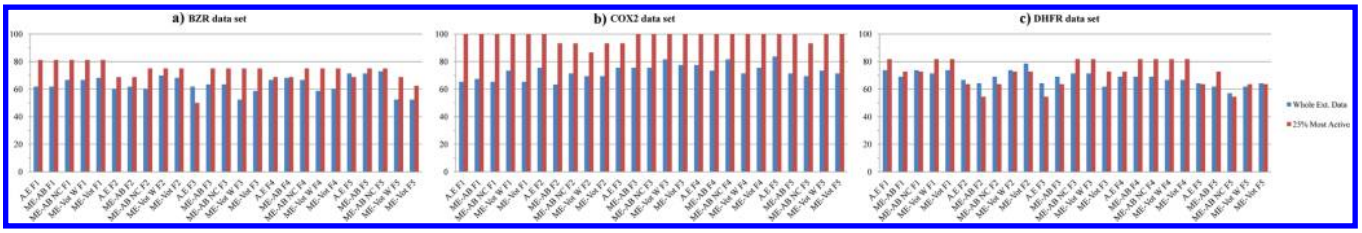


Figure 8. Comparative study of the global Sensitivity of the models and their Sensitivities when only the top 25% most actives compounds are analyzed for the BZR, COX2, and DHFR data sets. The calculations were perform for the Adaboost Ensemble (A.E), the clustering derived Adaboost Meta-Ensemble (ME-AB), the Adaboost Meta-Ensemble trained using the classic greedy search strategy without model clustering (ME-AB NC), the Weighted Votes Meta-Ensemble (ME-Vot W), and the Unweighted Votes Meta-Ensemble (ME-Vot).

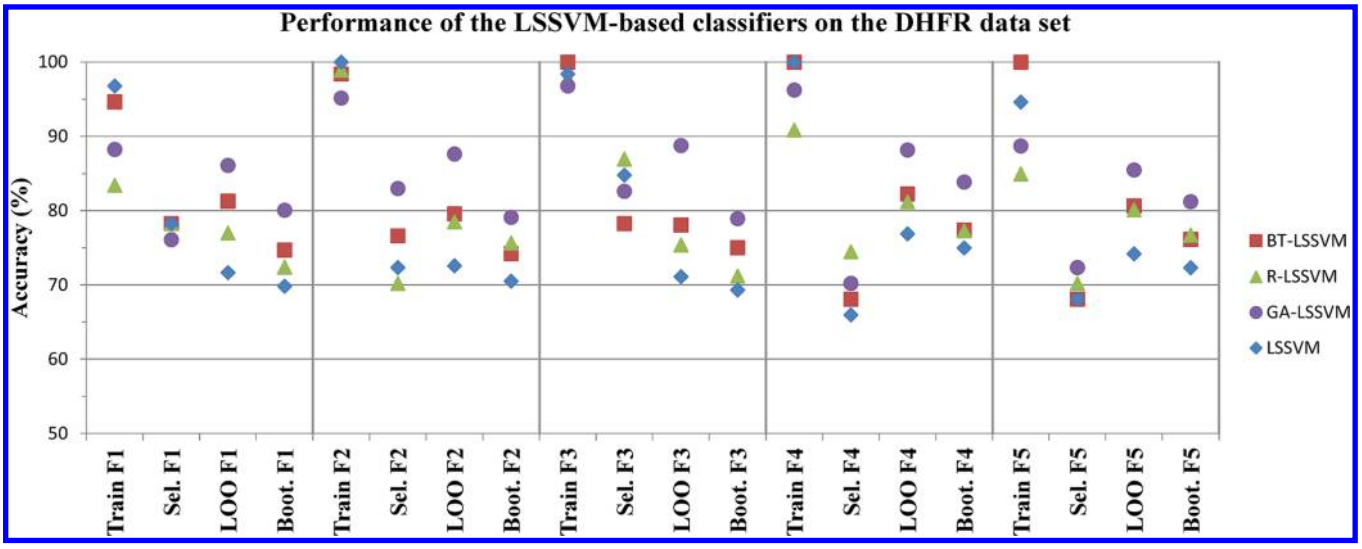


Figure 9. Comparison of the performance of the LS-SVM based classifiers for the five modeling folds of the DHFR data set. The accuracy in predicting the training data set (Train), the accuracy in the prediction of the selection data set (Sel.), and the Leave-One-Out (LOO) and Bootstrap (Boot) cross-validation accuracies are shown for the LS-SVM model trained without feature selection (LS-SVM) and for the LS-SVM models trained with features selected using Bagged Trees (BT-LS-SVM), Features Ranking (R-LS-SVM), and Genetic Algorithm (GA-LS-SVM).

learning methods, we trained SVM and LS-SVM classifiers without feature selection using the RBF kernel. We also evaluated the combination of Genetic Algorithms with SVM and LS-SVM classifiers. Finally, we performed feature selection using Bagged Trees and a Wilcoxon rank−sum based methodology and used the top 25 features derived from each method to train Adaboost Ensemble, SVM, LS-SVM, and LDA models. In summary, additionally to the Adaboost Ensemble derived from the GA(M)E-QSAR methodology, we tested 12 classifiers that were trained with and without feature selection. The feature selection strategies as well as the training methodology of each type of classifier were performed following the procedure described in the Computational Methods section. The performance parameters of each optimal classifier per modeling methodology for each of the 25 experiments (5 modeling folds per data set) are presented in the Supporting Information section Tables TS9 to TS13.

An overview of the results of the classifiers and feature selection methods shows that there is not a method that can clearly outperform the performance of the rest of them. The results of the 25 classification experiments that we ran highlight the fact that there is not such a thing like a perfect feature selection or classification technique that will perform optimally in the solution of every classification problem. However, it can be seen that the Adaboost Ensembles found with the GA(M)E-QSAR methodology can produce models with performance at the same level as those obtained using the feature selection and classification techniques used in the comparative study. Furthermore, since the Adaboost algorithm only involves the computation of scalar products without any classifier training (see Chart 3), the algorithm here proposed is much simpler than classical wrapper methods such as the widely used GA-SVM approach.

One general rule is that the two classification methods that we tested that do not require feature selection, SVM and LSSVM, despite fitting perfectly or almost perfectly the training data set, show performances on the prediction of the selection set and in the cross-validation experiments lower than those obtained when the same classifiers are combined with any of the feature selection methods we tested. Despite the values of the accuracy in predicting the selection data set and the LOO and Bootstrap cross-validation accuracies can be considered good, it is preferable to use feature selection techniques to obtain models with better quality. It can also be seen that the models derived from the features selected using the Genetic Algorithm optimization are, in general, more robust and show higher generalization capabilities than those derived from either the whole features or from the other two feature selection methods that we evaluated. This behavior is illustrated in Figure 9 for the LS-SVMbased classifiers obtained for the DHFR data set.

To get more insights into the performance of the different classifiers and feature selection techniques that we used, we carried out a consensus ranking process of each classifier to attempt to select an optimal modeling technique for each of the

dx.doi.org/10.1021/ci300146h | J. Chem. Inf. Model. 2012, 52, 2366−2386

classification experiments that we studied. It is known that not all machine learning methods are equally sensitive to overfitting and that the performance in cross-validation experiments is closer to the expected performance of a model when it is used to predict data with unknown response than the observed performance on training data. Besides, not all classification techniques can achieve the same balance between Sensitivity and Specificity, and it is desirable that, apart from yielding a high accuracy, classifiers used for virtual screening also provide a good intergroup classification performance. Based on these points and in the fact that we have previously selected the optimal model per modeling approach; which already combine fitting, robustness, and generalization capabilities; we focus the comparative analysis of the different modeling techniques solely on their robustness (measured as the cross-validated LOO and Bootstrap accuracies) and their generalization capabilities taking into account the accuracy in the prediction of the selection set and $(Sensitivity*Specificity)^{1/2}$ to consider the balance between the intergroup classification performances. In Table 3 we show the top 3 classifiers per experiment according to the results of the consensus ranking.

Table 3. Top 3 Modeling Approaches Per Experiment According to the Consensus Ranking-Based Selection[a]

		Ranking		
Target	Fold	1	2	3
BZR	1	GA-LSSVM	GA-SVM	R-LSSVM
	2	A.E	BT-LSSVM	BT-SVM
	3	BT-SVM	GA-LSSVM	BT-LSSVM
	4	GA-LSSVM	A.E	BT-LSSVM
	5	A.E	BT-LSSVM	R-LSSVM
COX2	1	A.E	GA-LSSVM	BT-LSSVM
	2	A.E	GA-SVM	R-SVM
	3	GA-LSSVM	BT-LDA	BT-LSSVM
	4	A.E	GA-LSSVM	GA-SVM
	5	GA-SVM	GA-LSSVM	A.E
DHFR	1	A.E	GA-SVM	GA-LSSVM
	2	A.E	GA-SVM	GA-LSSVM
	3	GA-LSSVM	GA-SVM	R-SVM
	4	A.E	GA-LSSVM	BT-SVM
	5	GA-LSSVM	A.E	BT-AB
ER	1	GA-SVM	BT-SVM	BT-LSSVM
	2	GA-LSSVM	GA-SVM	BT-LSSVM
	3	BT-SVM	GA-LSSVM	GA-SVM
	4	BT-SVM	GA-LSSVM	GA-SVM
	5	GA-LSSVM	GA-SVM	BT-SVM
DILI	1	GA-LSSVM	GA-SVM	A.E
	2	GA-SVM	R-SVM	GA-LSSVM
	3	A.E	GA-SVM	GA-LSSVM
	4	A.E	GA-SVM	GA-LSSVM
	5	GA-LSSVM	GA-SVM	A.E

[a]A.E: Adaboost Ensemble derived from the GA(M)E-QSAR approach. GA-*: Genetic Algorithm feature selection. BT-*: Bagged Tress feature selection. R-*: Ranking feature selection. SVM: Support Vector Machine. LSSVM: Least Square Support Vector Machine. A.B: Adaboost Ensemble. LDA: Linear Discriminant Analysis.

As shown in Table 3, the Adaboost Ensembles obtained through the Genetic Algorithm optimization process that we proposed is among the top 3 modeling approaches for 15 out of the 25 experiments we carried out. This value is lower than the 21 times that the GA-LS-SVM wrapper is in the top 3 techniques and close to the 17 times that the GA-SVM wrapper and the Bagged Trees-based classifiers are selected among the top 3 classification methodologies. More important, the Adaboost Ensemble resulting from the methodology we proposed in this study is chosen as the optimal classification approach for 10 out of the 25 experiments, while classifiers based on Support Vector Machines are selected as the optimal methodology in 15 experiments. Nevertheless, if we separately consider the SVM and LS-SVM based methods, the Adaboost Ensemble is ranked first is more experiments (10) than any classification approach and outperforms the GA-SVM and GA-LSSVM wrappers which are selected as the optimal classification method 9 and 3 times respectively. It should also be noted that for all the experiments the top classification approach is obtained using GA-based feature selection 22 times showing the search capabilities of Genetic Algorithms as feature selection technique. Our results also show the advantage of using the GA optimization to select the members of the Adaboost Ensemble over other feature selection methods such as Bagged Trees and Feature Ranking. One important observation in agreement with the results derived from the comparison of the LS-SVM based models when feature selection and no feature selection is used is that none of the two methods that we tested that does not require feature selection is chosen among the top 3 modeling techniques in any of the 25 classification experiments. If the performance of these top three classifiers per experiment (see the Supporting Information Table TS14) is analyzed, it can be seen that it is similar for all three classifiers with high values of accuracy in the prediction of the selection set and high LOO and Bootstrap cross-validation accuracies. More important, all of them are able to yield good performances on the external data set that is used only as the final evaluation criterion of the quality of the models. In Table 4 we show the performance statistics of each of the optimal model per experiment. We included the following for each experiment: the optimal classification strategy (Method), the modeling fold (Fold), the number of feature included in the model (Size), the accuracy in the prediction of the training set (Train(%)), the accuracy in the prediction of the selection set (Sel.(%)), the Leave-One-Out (LOO(%)), Bootstrap (Boot.(%)) cross-validation accuracies, and the accuracy in the prediction of the external set (Ext.).

One of the conditions that a QSAR model should fulfill according to the OECD guidelines is that it should have a defined applicability domain.[76] Here we investigated the percent of the external data set that is inside the applicability domain of the classifiers trained for each experiment. The calculation of the applicability domain was based in the ranges method as explained in the Computational Methods section. We include the percent of coverage of the external set by every model developed for each experiment in the Supporting Information Table TS15. We summarize these results in Table 5 where the minimum and maximum coverage (expressed in %) of the external set by the applicability domain defined for each classification approach applied to the data sets under study are shown.

As expected, the more features are included in the model, the narrower the applicability domain of the model is. This is a straightforward conclusion when the applicability domain defined by all features is compared with the respective feature selection methods. For the problems and classification methods here compared, it is preferable to use feature selection-based classifiers since they perform better than the two nonfeature selection classifiers that we tested (SVM and LS-SVM) as shown by the consensus selection of the optimal classification methodology per problem. However, we also think that for a

Table 4. Performance of the Optimal Model on Each of the Classification Experiments We Carried Out[a]

Method	Fold	Size	Train(%)	Sel.(%)	LOO(%)	Boot.(%)	Ext.(%)
				BZR			
GA-LSSVM	1	13	87 (92/81)	78 (83/72)	85	78	67 (68/66)
A.E	2	14	92 (94/91)	78 (71/84)	91	84	66 (60/71)
BT-SVM	3	25	97 (96/97)	81 (78/85)	81	75	78 (73/84)
GA-LSSVM	4	12	88 (84/91)	89 (95/83)	84	78	73 (68/77)
A.E	5	13	90 (90/90)	75 (82/68)	88	82	70 (71/69)
				COX2			
A.E	1	10	89 (93/86)	89 (75/96)	87	80	63 (65/62)
A.E	2	11	88 (88/88)	77 (57/90)	88	81	74 (76/72)
GA-LSSVM	3	14	93 (95/92)	78 (92/70)	91	87	74 (82/68)
A.E	4	6	87 (89/86)	77 (81/74)	86	80	74 (78/71)
GA-SVM	5	44	95 (95/95)	92 (81/100)	92	84	73 (67/76)
				DHFR			
A.E	1	13	85 (82/87)	83 (88/79)	83	78	72 (74/71)
A.E	2	15	83 (84/83)	85 (81/87)	80	77	76 (67/79)
GA-LSSVM	3	19	97 (97/97)	83 (71/88)	89	79	79 (71/82)
A.E	4	13	88 (89/87)	72 (61/79)	84	79	76 (69/78)
GA-LSSVM	5	25	89 (74/97)	72 (53/86)	85	81	75 (50/84)
				ER			
GA-SVM	1	44	96 (98/95)	80 (82/78)	95	90	77 (72/81)
GA-LSSVM	2	16	90 (94/87)	96 (95/97)	89	85	78 (72/82)
BT-SVM	3	6	93 (96/91)	89 (93/84)	82	79	76 (68/81)
BT-SVM	4	15	97 (99/96)	85 (86/84)	87	83	78 (61/90)
GA-LSSVM	5	17	95 (98/93)	83 (84/82)	93	88	75 (68/80)
				DILI			
GA-LSSVM	1	26	84 (86/82)	74 (75/73)	77	70	67 (72/62)
GA-SVM	2	89	81 (71/89)	73 (57/88)	77	71	66 (62/70)
A.E	3	9	75 (76/74)	65 (65/66)	73	68	57 (55/58)
A.E	4	9	71 (69/72)	70 (68/72)	70	64	59 (56/63)
GA-LSSVM	5	25	81 (76/85)	69 (76/64)	79	69	57 (46/66)

[a]A.E: Adaboost Ensemble derived from the GA(M)E-QSAR approach. GA-*: Genetic Algorithm feature selection. BT-*: Bagged Tress feature selection. SVM: Support Vector Machine. LSSVM: Least Square Support Vector Machine.

Table 5. Coverage of the External Set by the Applicability Domain Derived from Each Classification Approach[a]

Target	No FS	A.E	BT-AB	BT-SVM	BT-LSSVM	BT-LDA
BZR	38/51	70/93	68/87	77/90	77/94	72/86
COX2	34/43	81/91	79/99	58/79	56/76	69/83
DHFR	46/55	81/92	84/95	81/91	81/86	79/91
ER	46/53	89/92	92/100	84/95	86/97	84/98
DILI	47/48	90/97	93/98	81/96	92/96	88/96
Target	R-AB	R-SVM	R-LSSVM	R-LDA	GA-SVM	GA-LSSVM
BZR	78/100	74/95	73/95	75/90	62/80	86/90
COX2	70/98	63/80	68/94	61/86	40/68	74/87
DHFR	90/99	84/89	81/96	85/94	59/78	77/92
ER	89/99	89/93	86/91	86/92	71/79	83/88
DILI	94/98	89/98	85/96	89/98	53/68	88/95

[a]No FS: No feature selection performed.A.E: Adaboost Ensemble derived from the GA(M)E-QSAR approach. GA-*: Genetic Algorithm feature selection. BT-*: Bagged Tress feature selection. R-*: Ranking feature selection. SVM: Support Vector Machine. LSSVM: Least Square Support Vector Machine. A.B: Adaboost Ensemble. LDA: Linear Discriminant Analysis.

particular problem where the performance of a method that does not require feature selection is better than that of feature selection-based classifiers, it should be considered an option despite many possible active molecules can be discarded due to the narrow its applicability domain is. Another option to increase the coverage of the applicability domain of nonfeature selection classifiers can be to use a smaller number of input variables.

We would like to mention that, as seen from Table 5, the Adaboost Ensembles derived from the methodology that we propose here is able to achieve a high coverage of the external data by their applicability domains and that this behavior is closely related to the relative small number of LDA models selected by the Genetic algorithm to build these ensembles. Finally, we consider that the applicability domain of the models is a factor to be taken into account when deciding if it is worth using a Meta-Ensemble of Adaboost Ensembles for further use (i.e., for virtual screening porpoises) even when we previously showed that they are, in general, more robust than the single Adaboost Ensembles found during the Genetic Algorithm search.

CONCLUDING REMARKS

Here we presented a novel algorithm based on the combination of Genetic Algorithms, Adaboost Ensembles, and Adaboost and Voting Meta-Ensembles to be used for ligand based-drug design. The algorithm is computationally simple due to the Adaboost-based fitness function used to train the modified GA. This simplicity of the GA wrapper can be particularly exploited in the modeling of large data sets. We should remark that the developed methodology is fully automatic from the data preprocessing step to the selection of the optimal models and produces models with high accuracy in the prediction of the external data set that is

never used to train or select the optimal models. In the context of a virtual screening experiment, the models are also able to enrich the predicted active compounds subset with high potency chemicals.

We made an exhaustive evaluation of our algorithm and studied its stability and robustness. Despite the simplicity of the Adaboost algorithm, the quality of the models we obtained was comparable with the results previously reported for the data sets used to validate the methodology. Granted that there is not a unique classification approach that can perform well on every problem, we compared the Adaboost Ensembles derived from the GA(M)E-QSAR algorithm with some state of the art feature selection and classification techniques and showed that these Adaboost Ensembles can achieve the same level of accuracy, robustness, and generalization provided by other feature selection and classification methods.

However, the method suffers from the convergence to statistical equivalent local minimums as most of the machine learning-based feature selection methods used in QSAR modeling. In this sense, we showed that the use of Meta-Ensembles can, in some cases, improve the generalization of the optimal Adaboost Ensembles. The Adaboost Meta-Ensembles were also more robust than the Adaboost Ensembles as proved by the improvement in the LOO and Bootstrap values. As we discussed, the choice of an Adaboost Ensemble or a Meta-Ensemble as the final model is dependent on the quality of the models that are generated by the GA and how well they generalize to predict independent external data. Some other considerations when deciding whether to use a Meta-Ensemble or an Adaboost Ensemble is the coverage of the chemical search space by the applicability domain of the models and how effective is each method in retrieving the compounds with the highest potency. Since the time required to train and validate the four Meta-Ensemble types is equivalent to that required to complete one GA evolution, our advice is to train Single Adaboost Ensembles as well as the Meta-Ensembles and make a thorough evaluation of each model type to determine pros and cons before deciding which one will be used.

■ ASSOCIATED CONTENT

Supporting Information

The **Supporting Information** contains the detailed results of the GA(M)E-QSAR algorithm for all the data sets used to validate the algorithm in Tables **TS1 (BZR), TS2 (COX2), TS3-(DHFR), TS4(ER), and TS5(DILI)**. In this section are also provided the Minimum and Maximum values of the accuracy estimator parameters of the Adaboost Ensembles obtained with the GA(M)E-QSAR algorithm when the overall accuracy is used to filter the LDA models, as well as the goodness of fit estimator for AIC (**TS6**) and the correlation matrices of the four decision makers used to select the optimal Adaboost Ensemble for each target and modeling fold (**TS7**). We include as **Supporting Information** the comparison of the overall Sensitivity of the models with their Sensitivity when only the top 25% most active compounds are analyzed (**TS8**). The accuracy estimator parameters for the optimal model per modeling approach is presented in the **Supporting Information** Tables **TS9** (BZR data set), **TS10** (COX2 data set), **TS11** (DHFR data set), **TS12** (ER data set), and **TS13** (DILI data set). Besides, the accuracy estimator parameters of the top 3 classification approaches per experiment are summarized in the **Supporting Information** Table **TS14**. In the **Supporting Information** Table **TS15** we report the percent of coverage of the external data set by the Applicability Domain defined by the training data set for each model trained in every classification experiment. We also report as **Supporting Information** the plots of the Density Distribution Estimation vs Accuracy for the four Decision Makers used to rank and select the optimal Adaboost Ensemble for the BZR (**FS1**), COX2 (**FS2**), ER (**FS3**), and DILI (**FS5**)data sets. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*Phone: + 32 2 629 33 08. Fax: + 32 2 629 37 08. E-mail: yunierkis@gmail.com (Y.P.-C.), ann.nowe@vub.ac.be (A.N.).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank the Vlaams Supercomputer Centrum (VSC) for providing access to the computational resources needed for accomplishing all calculations. Pérez-Castillo Y. thanks the Flemish Interuniversity Council (VLIR) for financial support through the project: "Strengthening research and PhD formation in Computer Sciences and its applications" in the framework of the VLIR-UCLV collaborative program. Cabrera-Pérez M. A. thanks the Spanish Agency of International Cooperation for the Development (AECID) for financial support through the project "Montaje de un laboratorio de química computacional, con fines académicos y científicos, para el diseño racional de nuevos candidatos a fármacos en enfermedades de alto impacto social" (D/024153/09). The authors thank the Brussels Institute for Research and Innovation (INNOVIRIS) who partially funded this research.

■ REFERENCES

(1) Wilson, G. L.; Lill, M. A. Integrating structure-based and ligand-based approaches for computational drug design. *Future Med. Chem.* **2011**, *3* (6), 735−50.

(2) Mavromoustakos, T.; Durdagi, S.; Koukoulitsa, C.; Simcic, M.; Papadopoulos, M. G.; Hodoscek, M.; Grdadolnik, S. G. Strategies in the rational drug design. *Curr. Med. Chem.* **2011**, *18* (17), 2517−30.

(3) Favia, A. D. Theoretical and computational approaches to ligand-based drug discovery. *Front Biosci.* **2011**, *16*, 1276−90.

(4) Tuccinardi, T.; Ortore, G.; Santos, M. A. l.; Marques, S. r. M.; Nuti, E.; Rossello, A.; Martinelli, A. Multitemplate alignment method for the development of a reliable 3D-QSAR model for the analysis of MMP3 inhibitors. *J. Chem. Inf. Model.* **2009**, *49* (7), 1715−1724.

(5) Hajjo, R.; Grulke, C. M.; Golbraikh, A.; Setola, V.; Huang, X. P.; Roth, B. L.; Tropsha, A. Development, validation, and use of quantitative structure-activity relationship models of 5-hydroxytryptamine (2B) receptor ligands to identify novel receptor binders and putative valvulopathic compounds among common drugs. *J. Med. Chem.* **2010**, *53* (21), 7573−86.

(6) Tang, H.; Wang, X. S.; Huang, X. P.; Roth, B. L.; Butler, K. V.; Kozikowski, A. P.; Jung, M.; Tropsha, A. Novel inhibitors of human histone deacetylase (HDAC) identified by QSAR modeling of known inhibitors, virtual screening, and experimental validation. *J. Chem. Inf. Model.* **2009**, *49* (2), 461−76.

(7) Shen, M.; Béguin, C.; Golbraikh, A.; Stables, J. P.; Kohn, H.; Tropsha, A. Application of predictive QSAR models to database mining: identification and experimental validation of novel anticonvulsant compounds. *J. Med. Chem.* **2004**, *47* (9), 2356−2364.

(8) Doweyko, A. M. QSAR: dead or alive? *J. Comput.-Aided Mol. Des.* **2008**, *22* (2), 81−9.

(9) Tropsha, A. Best practices for QSAR model development, validation, and exploitation. *Mol. Inf.* **2010**, *29* (6−7), 476−488.

(10) Fourches, D.; Muratov, E.; Tropsha, A. Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research. *J. Chem. Inf. Model.* **2010**, *50* (7), 1189−204.
(11) Maggiora, G. M. On outliers and activity cliffs−why QSAR often disappoints. *J. Chem. Inf. Model.* **2006**, *46* (4), 1535.
(12) Golbraikh, A.; Tropsha, A. Beware of q2! *J. Mol. Graphics Modell.* **2002**, *20* (4), 269−76.
(13) Ivanciuc, O. Applications of Support Vector Machines in Chemistry. In *Reviews in Computational Chemistry*; John Wiley & Sons, Inc.: 2007; pp 291−400.
(14) Klon, A. E.; Lowrie, J. F.; Diller, D. J. Improved naive Bayesian modeling of numerical data for absorption, distribution, metabolism and excretion (ADME) property prediction. *J. Chem. Inf. Model.* **2006**, *46* (5), 1945−56.
(15) Zheng, W.; Tropsha, A. Novel variable selection quantitative structure−property relationship approach based on the k-nearest-neighbor principle. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (1), 185−94.
(16) Zhang, S.; Golbraikh, A.; Oloff, S.; Kohn, H.; Tropsha, A. A novel automated lazy learning QSAR (ALL-QSAR) approach: method development, applications, and virtual screening of chemical databases using validated ALL-QSAR models. *J. Chem. Inf. Model.* **2006**, *46* (5), 1984−95.
(17) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (6), 1947−58.
(18) Ivanciuc, O. Aquatic Toxicity Prediction for Polar and Nonpolar Narcotic Pollutants with Support Vector Machines. *Internet Electron. J. Mol. Des.* **2003**, *2*, 195−208.
(19) Hudelson, M. G.; Ketkar, N. S.; Holder, L. B.; Carlson, T. J.; Peng, C. C.; Waldher, B. J.; Jones, J. P. High confidence predictions of drug-drug interactions: predicting affinities for cytochrome P450 2C9 with multiple computational methods. *J. Med. Chem.* **2008**, *51* (3), 648−54.
(20) Tong, W.; Hong, H.; Fang, H.; Xie, Q.; Perkins, R. Decision forest: combining the predictions of multiple independent decision tree models. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (2), 525−31.
(21) Patel, J. Science of the Science, Drug Discovery and Artificial Neural Networks. *Curr. Drug Discovery Technol.* **2012**, Epub ahead of print, published online June 25, 2012.
(22) Ivanciuc, O. Weka machine learning for predicting the phospholipidosis inducing potential. *Curr. Top. Med. Chem.* **2008**, *8* (18), 1691−709.
(23) Ivanciuc, O. Artificial immune system prediction of the human intestinal absorption of drugs with AIRS (artificial immune recognition system). *Internet Electron. J. Mol. Des.* **2006**, *5*, 515−529.
(24) Shi, W. M.; Shen, Q.; Kong, W.; Ye, B. X. QSAR analysis of tyrosine kinase inhibitor using modified ant colony optimization and multiple linear regression. *Eur. J. Med. Chem.* **2007**, *42* (1), 81−6.
(25) Fernandez, M.; Caballero, J.; Fernandez, L.; Sarai, A. Genetic algorithm optimization in drug design QSAR: Bayesian-regularized genetic neural networks (BRGNN) and genetic algorithm-optimized support vectors machines (GA-SVM). *Mol. Diversity* **2011**, *15* (1), 269−89.
(26) Zhou, X.; Li, Z.; Dai, Z.; Zou, X. QSAR modeling of peptide biological activity by coupling support vector machine with particle swarm optimization algorithm and genetic algorithm. *J. Mol. Graphics Modell.* **2010**, *29* (2), 188−96.
(27) Al-Sha'er, M. A.; Taha, M. O. Elaborate Ligand-Based Modeling Reveals New Nanomolar Heat Shock Protein 90α Inhibitors. *J. Chem. Inf. Model.* **2010**, *50* (9), 1706−1723.
(28) Cheng, Z.; Zhang, Y.; Zhou, C. QSAR models for phosphoramidate prodrugs of 2'-methylcytidine as inhibitors of hepatitis C virus based on PSO boosting. *Chem. Biol. Drug Des.* **2011**, *78* (6), 948−59.
(29) Wen, J. H.; Zhong, K. J.; Tang, L. J.; Jiang, J. H.; Wu, H. L.; Shen, G. L.; Yu, R. Q. Adaptive variable-weighted support vector machine as optimized by particle swarm optimization algorithm with application of QSAR studies. *Talanta* **2011**, *84* (1), 13−8.
(30) Abbasitabar, F.; Zare-Shahabadi, V., Development predictive QSAR models for artemisinin analogues by various feature selection methods: A comparative study. *SAR QSAR Environ Res* **2011**.
(31) Goodarzi, M.; Freitas, M. P.; Jensen, R. Feature selection and linear/nonlinear regression methods for the accurate prediction of glycogen synthase kinase-3beta inhibitory activities. *J. Chem. Inf. Model.* **2009**, *49* (4), 824−32.
(32) Ivanciuc, O.; Ivanciuc, T.; Cabrol-Bass, D. QSAR for dihydrofolate reductase inhibitors with molecular graph structural descriptors. *J. Mol. Struct.: THEOCHEM* **2002**, *582* (1−3), 39−51.
(33) Ivanciuc, O. Drug Design with Artificial Intelligence Methods. In *Encyclopedia of Complexity and Systems Science*; Meyers, R. A., Ed.; Springer-Verlag: New York, 2009.
(34) Ivanciuc, O. Drug Design with Machine Learning. In *Encyclopedia of Complexity and System Science*; Meyers, R. A., Ed.; Springer-Verlag: New York, 2009.
(35) Duch W Fau - Swaminathan, K.; Swaminathan K Fau - Meller, J.; Meller, J. Artificial intelligence approaches for rational drug design and discovery. (1873−4286 (Electronic)).
(36) Gonzalez, M. P.; Teran, C.; Saiz-Urra, L.; Teijeira, M. Variable selection methods in QSAR: an overview. *Curr. Top. Med. Chem.* **2008**, *8* (18), 1606−27.
(37) Holland, J. H. *Adaptation in natural and artificial systems*; MIT Press: 1992; p 211.
(38) Cartwright, H. M. *Applications of Artificial Intelligence in Chemistry*; Oxford University Press, Inc.: 1994; p 96.
(39) Huang, J.; Fan, X. Why QSAR fails: an empirical evaluation using conventional computational approach. *Mol. Pharmaceutics* **2011**, *8* (2), 600−608.
(40) Zhang, Q.; Hughes-Oliver, J. M.; Ng, R. T. A Model-Based Ensembling Approach for Developing QSARs. *J. Chem. Inf. Model.* **2009**, *49* (8), 1857−1865.
(41) Zhu, H.; Tropsha, A.; Fourches, D.; Varnek, A.; Papa, E.; Gramatica, P.; Oberg, T.; Dao, P.; Cherkasov, A.; Tetko, I. V. Combinatorial QSAR modeling of chemical toxicants tested against Tetrahymena pyriformis. *J. Chem. Inf. Model.* **2008**, *48* (4), 766−84.
(42) Dutta, D.; Guha, R.; Wild, D.; Chen, T. Ensemble feature selection: consistent descriptor subsets for multiple QSAR models. *J. Chem. Inf. Model.* **2007**, *47* (3), 989−97.
(43) Fernandez, M.; Fernandez, L.; Caballero, J.; Abreu, J. I.; Reyes, G. Proteochemometric modeling of the inhibition complexes of matrix metalloproteinases with N-hydroxy-2-[(phenylsulfonyl)amino]-acetamide derivatives using topological autocorrelation interaction matrix and model ensemble averaging. *Chem. Biol. Drug Des.* **2008**, *72* (1), 65−78.
(44) Yoav, F.; Robert, E. S. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55* (1), 119−139.
(45) Chouaib, H.; Terrades, O. R.; Tabbone, S.; Cloppet, F.; Vincent, N., Feature selection combining genetic algorithm and Adaboost classifiers. In *ICPR*; IEEE: 2008; pp 1−4.
(46) Yanagimoto, H.; Omatu, S. Construction of a classifier using AdaBoost for information filtering. *Artificial Life Robotics* **2005**, *9* (2), 72−75.
(47) Dezhen, Z.; Kai, Y., Genetic Algorithm Based Optimization for AdaBoost. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 01*; IEEE Computer Society: 2008; pp 1044-1047.
(48) Chouaib, H.; Terrades, O. R.; Tabbone, S.; Cloppet, F.; Vincent, N. In *Feature selection combining genetic algorithm and Adaboost classifiers*, Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, 2008; 2008; pp 1-4.
(49) Dezhen, Z.; Kai, Y. In *Genetic Algorithm Based Optimization for AdaBoost*, CSSE '08: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, 2008; IEEE Computer Society: 2008; pp 1044-1047.
(50) Ran, L.; Jianjiang, L.; Yafei, Z.; Tianzhong, Z. Dynamic Adaboost learning with feature selection based on parallel genetic algorithm for image annotation. *Know.-Based Syst.* **2010**, *23* (3), 195−201.

(51) Yalabik, I.; Fatos, T. Y. V. In *A pattern classification approach for boosting with genetic algorithms*, Computer and information sciences, 2007. iscis 2007. 22nd international symposium on, 7−9 Nov. 2007, 2007; 2007; pp 1-6.
(52) Sutherland, J. J.; O'Brien, L. A.; Weaver, D. F. Spline-fitting with a genetic algorithm: a method for developing classification structure-activity relationships. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (6), 1906−15.
(53) Sutherland, J. J.; O'Brien, L. A.; Weaver, D. F. Pruned Receptor Surface Models and Pharmacophores for Three-Dimensional Database Searching. *J. Med. Chem.* **2004**, *47* (15), 3777−3787.
(54) Auer, J.; Bajorath, J. Emerging Chemical Patterns: A New Methodology for Molecular Classification and Compound Selection. *J. Chem. Inf. Model.* **2006**, *46* (6), 2502−2514.
(55) Evans, D. A.; Doman, T. N.; Thorner, D. A.; Bodkin, M. J. 3D QSAR Methods: Phase and Catalyst Compared. *J. Chem. Inf. Model.* **2007**, *47* (3), 1248−1257.
(56) Obrezanova, O.; Segall, M. D. Gaussian processes for classification: QSAR modeling of ADMET and target activity. *J. Chem. Inf. Model.* **2010**, *50* (6), 1053−61.
(57) Rathke, F.; Hansen, K.; Brefeld, U.; Muller, K. R. StructRank: a new approach for ligand-based virtual screening. *J. Chem. Inf. Model.* **2011**, *51* (1), 83−92.
(58) Santos-Filho, O. A.; Cherkasov, A. Using Molecular Docking, 3D-QSAR, and Cluster Analysis for Screening Structurally Diverse Data Sets of Pharmacological Interest. *J. Chem. Inf. Model.* **2008**, *48* (10), 2054−2065.
(59) Mahé, P.; Ralaivola, L.; Stoven, V.; Vert, J.-P. The Pharmacophore Kernel for Virtual Screening with Support Vector Machines. *J. Chem. Inf. Model.* **2006**, *46* (5), 2003−2014.
(60) Fourches, D.; Barnes, J. C.; Day, N. C.; Bradley, P.; Reed, J. Z.; Tropsha, A. Cheminformatics Analysis of Assertions Mined from Literature That Describe Drug-Induced Liver Injury in Different Species. *Chem. Res. Toxicol.* **2009**, *23* (1), 171−183.
(61) Talete *DRAGON (Software for Molecular Descriptor Calculation)*, 6.0; 2010.
(62) *MATLAB*, R2009a; The MathWorks Inc.: 2009.
(63) Akaike, H. In *Information theory and an extension of the maximum likelihood principle*, Second International Symposium on Information Theory, 1973; Petrov, B. N., Csaki, F., Eds.; Akadémiai Kiado: 1973; pp 267−281.
(64) Todeschini, R.; Consonni, V.; Pavan, M. A distance measure between models: a tool for similarity/diversity analysis of model populations. *Chemom. Intell. Lab. Syst.* **2004**, *70* (1), 55−61.
(65) Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence - Vol. 2*; Morgan Kaufmann Publishers Inc.: Montreal, Quebec, Canada, 1995; pp 1137−1143.
(66) Efron, B.; Tibshirani, R. J. *An Introduction to the Bootstrap*; Chapman & Hall: 1993.
(67) de Borda, J. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences* **1784**.
(68) Kemeny, J. G.; Snell, J. L. Preference Rankings - An Axiomatic Approach. In *Mathematical Models in the Social Sciences*; 1962; pp 9-23.
(69) Bogart, K. P. Preference structures II: distances between asymmetric relations. *SIAM J. Appl. Math.* **1975**, *29* (2), 254−262.
(70) Cook, W. D.; Kress, M.; Seiford, L. M. A general framework for distance-based consensus in ordinal ranking models. *Eur. J. Oper. Res.* **1997**, *96* (2), 392−397.
(71) Chang, C.-C.; Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Trans. Intelligent Systems Technol.* **2011**, *2* (27), 1 −27:27.
(72) Suykens, J. A. K.; Van Gestel, T.; De Brabanter, J.; De Moor, B.; Vandewalle, J. *Least Squares Support Vector Machines*; World Scientific: Singapore, 2002.
(73) Lehmann, E. L.; D'Abrera, H. J. M. *Nonparametrics: statistical methods based on ranks*; Springer: 2006.
(74) Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1* (6), 80−83.
(75) Jaworska, J.; Nikolova-Jeliazkova, N.; Aldenberg, T. QSAR applicabilty domain estimation by projection of the training set descriptor space: a review. *Altern. Lab. Anim.* **2005**, *33* (5), 445−59.
(76) OECD OECD Principles for the Validation, for Regulatory Purposes of (Quantitative) Structure-Activity Relationship Models. http://www.oecd.org/LongAbstract/0,3425,en_2649_34379_37849784_119669_1_1_1,00.html (accessed July 16, 2012).