

Simultaneous Feature Selection and Parameter Optimization using an Artificial Ant Colony

Journal:	<i>Journal of Chemical Information and Modeling</i>
Manuscript ID:	draft
Manuscript Type:	Article
Date Submitted by the Author:	n/a
Complete List of Authors:	O'Boyle, Noel; University of Cambridge, Department of Chemistry Palmer, David; University of Cambridge, Department of Chemistry Mitchell, John; University of Cambridge, Chemistry



Simultaneous Feature Selection and Parameter Optimization using an Artificial Ant Colony

Noel M. O'Boyle,[†] David S. Palmer and John B. O. Mitchell*

Unilever Centre for Molecular Science Informatics, Dept. of Chemistry, University of Cambridge, Lensfield Rd, Cambridge CB2 1EW, U.K.

*Corresponding author: email: jbom1@cam.ac.uk; fax: +44-1223-763076; phone: +44-1223-762983

[†]Current address: Cambridge Crystallographic Data Centre, 12 Union Rd, Cambridge, CB2 1EZ, U.K.

Abstract

The selection of descriptors that will allow a model to generalize well to unseen data is a crucial step in the development of predictive quantitative structure-activity relationship (QSAR) models. We present a novel feature selection algorithm, Winnowing Artificial Ant Colony (WAAC), that performs simultaneous feature selection and model parameter optimization. The winnowing procedure implemented in WAAC improves its ability to optimize the objective function by removing irrelevant descriptors. The algorithm was used to develop a QSAR model for a literature dataset of intrinsic solubilities using support vector machines (SVM) by optimizing the root mean squared error from 10-fold cross validation. Starting from an initial set of 144 descriptors, the best model contains 12 descriptors and has a root mean squared error of 0.93 Log S units on a holdout test set. The use of an objective function that explicitly penalizes the number of descriptors was also investigated, but was found to be unsuitable for use with the WAAC algorithm.

Introduction

The development of improved models to predict the activities (QSAR) or physical properties (QSPR) of small molecules is of key importance for identifying potential

1
2
3 lead compounds for drug development. Physical properties such as solubility and
4 permeability influence the ability of molecules to be adsorbed in the body and to
5 reach the desired target area. Models that predict so-called ADMET properties
6 (Adsorption, Distribution, Metabolism, Excretion and Toxicity) permit large libraries
7 of potential lead compounds to be screened in-silico, thus avoiding the cost associated
8 with the synthesis and experimental testing of molecules with undesirable properties.
9

10
11
12
13
14
15 QSAR (and QSPR) is based upon the idea, first proposed by Hansch,¹ that a
16 molecular property is dependent on physicochemical properties of the molecule.
17 These properties may be obtained from experiment or may be calculated, in which
18 case they are referred to as descriptors. Typically, when the factors that influence the
19 molecular property are not fully understood, a large number of descriptors are
20 calculated for each molecule in a training set. A model is then built using the
21 descriptors as features. A QSAR model for prediction must be able to generalize well
22 to give accurate predictions on unseen test data. Although it is true in general that the
23 more descriptors used to build a model, the better the model predicts the training set
24 data, such a model typically has very poor predictive ability when presented with
25 unseen test data. This is referred to as overfitting.² Feature selection refers to the
26 problem of selecting a subset of the descriptors which can be used to build a model
27 with optimal predictive ability.³ In addition to better prediction, the identification of
28 relevant descriptors can give insight into the factors affecting the property of interest.
29
30
31
32
33
34
35
36
37
38
39

40
41 The number of subsets of a set of n descriptors is $2^n - 1$. Unless n is small (< 20) it is not
42 feasible to test every possible subset, and the number of descriptors calculated by
43 cheminformatics software is usually much larger (CDK,⁴ MOE⁵ and Sybyl⁶ can
44 respectively calculate a total of 95, 182 and 248 1D and 2D descriptors). Feature
45 selection methods can be divided into two main classes: the filter approach and the
46 wrapper approach.^{3,7,8} The filter approach does not take into account the particular
47 model being used for prediction, but rather attempts to determine *a priori* which
48 descriptors are likely to contain useful information. Examples of this approach include
49 ranking descriptors by their correlation with the target value or by estimates of the
50 mutual information (based on information theory) between each descriptor and the
51 response. Another commonly used filter in QSAR is the removal of highly correlated
52
53
54
55
56
57
58
59
60

1
2
3 (or anti-correlated) descriptors.⁹ Liu¹⁰ presents a comparison of five different filters in
4 the context of prediction of binding affinities to thrombin. The filter approach has the
5 advantages of speed and simplicity, but the disadvantage that it does not explicitly
6 consider the performance of the model containing different features. Correlation
7 criteria can only detect linear dependencies between descriptor values and the
8 response, but the best performing QSAR models are often non-linear (support vector
9 machines (SVM), neural networks (NN) and random forests (RF), for example). In
10 addition, Guyon and Elisseeff show that very high correlation (or anti-correlation)
11 does not necessarily imply an absence of feature complementarity, and also that two
12 variables that are useless by themselves can be useful together.³
13
14
15
16
17
18
19
20
21

22 The wrapper approach conducts a search for a good feature selection using the
23 induction algorithm as a black box to evaluate subsets and calculate the value of an
24 objective function. The objective function should provide an estimate of how well the
25 model will generalize to unseen data drawn from the same distribution. The purpose
26 of the search is to find the feature selection that optimizes this value. The most well-
27 known deterministic wrapper is sequential forward selection¹¹ (SFS) which involves
28 successive additions of the feature that most improves the objective function to the
29 subset of descriptors already chosen. A related algorithm, sequential backwards
30 elimination¹² (SBE), successively eliminates descriptors starting from the complete
31 set of descriptors. Both of these algorithms suffer from the problem of 'nesting'. In the
32 case of SFS, nesting refers to the fact that once a particular feature is added it cannot
33 be removed at a later stage, even if this would increase the value of the objective
34 function. More sophisticated methods, such as the sequential forward floating
35 selection (SFFS) algorithm of Pudil et al.,¹³ include a backtracking phase after each
36 addition where variables are successively eliminated if this improves the objective
37 function. Wrapper methods specific to certain models have also been developed. For
38 example, the Recursive Feature Elimination algorithm of Guyon et al.¹⁴ and the
39 Incremental Regularized Risk Minimization of Fröhlich et al.¹⁵ are specific to models
40 built using support vector machines.
41
42
43
44
45
46
47
48
49
50
51
52
53
54

55
56 Stochastic wrappers attempt to deal with the size of the search space by incorporating
57 some degree of randomness into the search strategy. The most well known of these
58
59
60

1
2
3 algorithms is the genetic algorithm¹⁶ (GA), whose search procedure mimics the
4 biological process of evolution. A number of models are created randomly in the first
5 generation, the best of which (as measured by the objective function) are selected and
6 interbred in some way to create the next generation. A mutation operator is applied to
7 the new models so that random sampling of the local space occurs. Over the course of
8 many generations, the objective function is optimized. Genetic algorithms were first
9 used for feature selection in QSAR by Rogers and Hopfinger¹⁷ and are now used
10 widely.^{9,18,19} Other stochastic methods which have been used for feature selection in
11 QSAR are particle swarm optimization^{20,21} and simulated annealing.²²

12
13
14
15
16
17
18
19
20
21 An additional difficulty in the development of QSAR models is the fact that some
22 regression methods have parameters that need to be optimized to obtain the best
23 performance for a particular problem. The Support Vector Machine (SVM) is an
24 example of such a method. A SVM is a kernel-based machine learning method used
25 for both classification and regression,^{23,24,25} and which has shown very good
26 performance in QSAR studies.⁹ In ϵ -SVM regression, the algorithm finds a
27 hyperplane in a transformed space of the inputs that has at most ϵ deviation from the
28 output y values. Deviations greater than ϵ are penalized by multiplying by a cost value
29 C . The transformation of the inputs is carried out by means of kernel functions, which
30 allows nonlinear relationships between the inputs and the outputs to be handled by
31 this essentially linear method. For a particular problem and kernel, the values of C and
32 ϵ must be tuned.

33
34
35
36
37
38
39
40
41
42 Here we describe WAAC, Winoing Artificial Ant Colony, a stochastic wrapper for
43 feature selection and parameter optimization that combines simultaneous optimization
44 of the selected descriptors and the model parameters to create a model with good
45 predictive accuracy. This method does not require any pre-processing of the data apart
46 from removal of zero-variance and duplicate descriptors. The only requirement is that
47 allowed values of parameters of the models must be specified. As a result, this method
48 is suitable for use as an automatic generator of predictive models.

49
50
51
52
53
54
55
56 The WAAC algorithm is a novel stochastic wrapper derived from the modified Ant
57 Colony Optimization (ACO) algorithm of Shen et al.²⁶ Ant colony algorithms take
58
59
60

1
2
3 their inspiration from the foraging of ants whose cooperative behavior enables the
4 shortest path between nest and food to be found.²⁷ Ants deposit a substance called
5 pheromone as they walk, thus forming a pheromone trail. At a branching point, an ant
6 is more likely to choose the trail with the greater amount of pheromone. Over time as
7 pheromones evaporate, only those trails that have been reinforced by the passage of
8 many ants will retain appreciable amounts of pheromone, with the shortest trail
9 having the greatest amount of pheromone. In the end, all of the ants will travel by the
10 shortest trail. Artificial ant colony systems may be used to solve combinatorial
11 optimization problems by making use of the ideas of cooperation between
12 autonomous agents through global knowledge and positive feedback that are observed
13 in real ant colonies.²⁸

14
15
16
17
18
19
20
21
22
23
24 The following section describes the WAAC algorithm in detail. Next, we illustrate the
25 performance of the algorithm on the solubility dataset of Chen et al.^{29,30} in
26 combination with two different types of objective function. Finally we compare
27 WAAC to related algorithms and discuss its performance.

28 29 30 31 32 33 **WAAC Algorithm**

34
35
36 The structure of the algorithm is shown in Figure 1. An initialization phase is
37 followed by an optimization phase after which, if convergence is not achieved, a
38 winnowing procedure is applied so that the algorithm starts over with a reduced
39 feature space. Convergence is achieved when the same best model is found in two
40 consecutive optimization phases. An implementation of WAAC in R³¹ is available
41 from the authors on request.

42 43 44 45 46 47 48 49 **Ant colony**

50
51
52 The term 'ant colony' refers to the set of models that are moved around in feature and
53 parameter space from one iteration of the algorithm to the next. Each ant represents a
54 model; that is, it is associated with a particular feature selection as well as particular
55 values for the model (e.g. SVM) parameters. The set of descriptors is stored as a
56 binary fingerprint of length F (the number of descriptors), where a value of 1 for the
57
58
59
60

1
2
3 n^{th} bit indicates that the n^{th} descriptor is selected, and 0 indicates that it is not.
4
5

6
7 Allowed values for the model parameters must be specified in advance. That is, for
8 each parameter of the model, a range of discrete values is required. The parameter
9 values used by a particular ant are stored in a list of length P , where P is the number
10 of adjustable parameters of the model. It is generally worthwhile to do an exploratory
11 run of the algorithm to determine reasonable ranges for the parameter values. It is
12 important that the number of allowed values for each parameter is less than the
13 number of ants (preferably much less), to ensure that the parameter space is
14 adequately sampled.
15
16
17
18
19

20
21
22 The fitness of each model is measured using an objective function. The purpose of the
23 algorithm is to optimize the value of the objective function to find the best model. In
24 addition, each ant 'remembers' the best model and best fitness value it has found in
25 that optimization phase. This information is used during the optimization phase to
26 place the ants in favored regions of model space and again in the winnowing
27 procedure to remove regions of feature space that are not favored.
28
29
30
31
32

33
34 The number of ants in the colony remains constant throughout the algorithm. An
35 appropriate size for the ant population depends on the number of descriptors and the
36 extent of the interaction between them. Model space will be better sampled if more
37 ants are used, but the calculation time will also increase. However, since the feature-
38 selection space is of size $2^n - 1$, where n is the number of descriptors, the exact number
39 of ants is not expected to affect the ability of the ACO to find solutions. An ant
40 population of between 50 and 100 ants is recommended.
41
42
43
44
45
46
47
48

49 **Initialization phase**

50
51 The initial population of ants is randomly placed in feature space. The bits of the
52 binary fingerprints representing the feature selections are initialized to either 0 or 1
53 with equal probability, so that on average each ant corresponds to a model based on
54 approximately 50% of the descriptors. Conversely, each descriptor is initially selected
55 by approximately 50% of the ants. Each ant is also placed randomly in parameter
56
57
58
59
60

1
2
3 space; that is, for each ant, the initial parameter values are chosen at random from the
4 available values for each parameter.
5
6
7

8 9 **Optimization phase**

10 For each descriptor, a moving probability is calculated by taking the average of the
11 fraction of ants which have currently selected that descriptor and the fraction that
12 have selected that descriptor in their best model. At the start of the optimization
13 phase, the moving probabilities for all of the descriptors will be approximately equal
14 to 0.5 (since the best model will be the current model and each descriptor is selected
15 by approximately 50% of the ants). In the next iteration, this moving probability is
16 used to determine the chance that a particular ant will select a particular descriptor.
17
18
19
20
21
22
23

24 Similarly, for each parameter there is a moving probability associated with every
25 allowed value. These moving probabilities sum to unity (since each ant needs to select
26 exactly one allowed value for each parameter), and are calculated by taking the
27 average of the fraction of ants which have currently selected a particular allowed
28 value and the fraction of ants that have selected that value in their best model. At the
29 start of the optimization phase, each allowed value of a parameter will be selected by
30 approximately N/P ants where N is the number of ants, and P the number of allowed
31 values. For this reason it is important that the number of allowed values is less than or
32 equal to the number of ants.
33
34
35
36
37
38
39
40
41

42 At the start of the optimization phase, the ants move more or less randomly, as the
43 moving probabilities are essentially equal for all features and parameter values.
44 However, over the course of the optimization phase as particular descriptors are found
45 to occur frequently in the best models associated with the ants, due to positive
46 feedback these descriptors will be more likely to be chosen in subsequent iterations.
47 This global optimization procedure is combined with local optimization due to the
48 influence of the current positions of the ants on the moving probabilities. Note that the
49 ants do not move about relative to their position in a previous iteration; rather, their
50 subsequent location in feature space is determined by the best and current feature
51 selections of all of the ants.
52
53
54
55
56
57
58
59
60

1
2
3
4
5 The length of the optimization phase should be sufficient to allow the objective
6 function to start to converge to an optimum value. However, it is not necessary to
7 allow the optimization phase to proceed much further as the winnowing procedure
8 and subsequent reinitialization provide more rapid improvements.
9

10 11 12 13 14 15 **Winnowing procedure**

16
17 If the optimization phase is left to run indefinitely, convergence occurs as the
18 descriptors chosen in the best models reinforce themselves until the moving
19 probabilities become either 0 or 1 and all models are identical. However, although the
20 resulting models will contain the descriptors found in the best models, only some of
21 those descriptors may actually contribute positively to the performance of those
22 models. Since broad sampling of the search space is no longer carried out when the
23 optimization is starting to converge, a reinitialization procedure on a smaller search
24 space is required.
25
26
27
28
29
30

31
32 Winnowing reduces the search space by retaining descriptors that have been chosen
33 by at least 20% of the ants in their best models, while removing the rest. Parameter
34 values are reinitialized randomly. Some descriptors may be retained that do not
35 improve the models as discussed above, but the subsequent reinitialization of the ants
36 on the smaller search space, will allow the optimization phase to identify better
37 models which exclude that descriptor.
38
39
40
41
42

43
44 Note that no information is carried from one optimization procedure to the next. In
45 particular, memory of previous best models does not guide future searching. This
46 means that the randomly initialized models in the new optimization phase are always
47 poorer than the best models of the previous phase, but the reduction in the size of the
48 feature space means that the performance of the model quickly recovers and matches
49 or improves on earlier performance.
50
51
52
53
54
55
56
57
58
59
60

Methods

Dataset

To test the performance of the WAAC algorithm in developing a robust QSAR module, we used the dataset of Chen et al., which contains intrinsic solubility data for 321 molecules.^{29,30} The intrinsic solubility of a molecule is the equilibrium concentration of the neutral form of the molecule in a saturated solution. We removed two molecules (terfenadine and propranolol) as their reported intrinsic solubilities disagreed by more than one log S unit with another literature dataset.³² MOE⁵ was used to calculate 161 1D and 2D descriptors for each molecule. The 319 molecules were divided randomly into a training set of 210 molecules and a holdout test set of 109 molecules. Where 90% or more of the molecules in the training set had identical values for a descriptor, that descriptor was removed. This was a requirement for performing 10-fold cross validation (see below) as it removed the possibility that a model would be built where one of the descriptors had zero variance, which causes an error with the SVM implementation we used. After this procedure, 144 descriptors remained.

The dataset and descriptor values are available as Supporting Information. Note that the names of the cycloalkane-spirobarbiturates have been changed in accordance with IUPAC guidelines, and that the molecular weight of L-DOPA has been corrected.

Objective function

The goal of the WAAC algorithm is to find the feature subset and parameter values that will give the best predictive accuracy for a model based on given training data. After a model is chosen, it is possible to measure the predictive accuracy on a hold-out test set from the same original dataset as the training data. However, during the course of the optimization, the algorithm needs to be guided by an objective function that will give an estimate of the predictive accuracy of a particular model.

Here we examine the performance of the WAAC algorithm in combination with two different objective functions: RMSE(cv) and a modified Lack-of-Fit (LOF) function.

Here, $RMSE(cv)$ is the root mean squared error from 10-fold cross validation. The training set is split equally into 10 hold-out sets, each of which is predicted using a model trained on the other 90%. The root mean square of the prediction error is then calculated. We have used different random splits of the training data each time to reduce overfitting effects. Cross validation has been widely used to guide previous feature selection algorithms.^{10,33} The RMSE from 10-fold cross validation is much better for this purpose than that from leave-one-out cross validation, in which the hold-out sets each contain a single molecule, leading to overoptimistic results by overfitting to the training data.³

The $RMSE(cv)$ does not contain an explicit cost function for the number of descriptors in a model. In fact, adding a nonsense descriptor or a descriptor highly-correlated to an existing explanatory descriptor to the model does not necessarily decrease the $RMSE(cv)$.³⁴ Since one of the purposes of feature selection is to exclude irrelevant variables, a cost function penalizing the addition of descriptors is often desirable. Several cost functions have been devised for linear models such as the Friedman's Lack-of-Fit function.³⁵ Here we use a modification of the Lack-of-Fit function ($mLOF$), adapted to more strongly penalize additional descriptors:

$$mLOF = \begin{cases} \frac{RSS}{\left(1 - \frac{dp}{M}\right)^2} & dp < M \\ \frac{RSS}{\left(1 - \frac{M-1}{M}\right)^2} & dp \geq M \end{cases} \quad \text{Equation 1}$$

where p is the number of descriptors in the model, M the number of molecules in the training set, RSS is the sum of the squares of the residuals, and d is a smoothing parameter whose value, 9, was chosen to pick models which, on average, have the same number of descriptors as the best model from the $RMSE(cv)$ runs.

Model

The *svm* method in the *e1071* package in R³¹ was used to perform ϵ regression with a radial basis function. A range of allowed parameter values for the SVM were chosen based on a preliminary run: values for C from 1 to 5 inclusive in steps of 0.2, and

values of ϵ from 0.01 to 0.2 inclusive in steps of 0.01.

Results

Using the training data, the WAAC algorithm was used to build two predictive SVM models for solubility using the RMSE(cv) objective function and the mLOF function, respectively. A colony of 50 ants was used, and the algorithm was run for 1500 iterations with winnowing every 100 iterations. For comparison, the algorithm was run for the same length without any winnowing.

Each experiment was performed 10 times with different random seeds. For each repetition, the model with the lowest value of the objective function was chosen from among the best models found in each optimization phase. To reduce the possibility of finding by chance a model which had an optimal value of the objective function but poor predictive ability, the final selected model was chosen from among the 10 repetitions using an alternative criterion. That is, we chose the model with the fewest number of parameters in the case of the RMSE(cv), and that with the lowest mean RMSE(cv) (from 100 random splits) in the case of the mLOF.

The best model found using each objective function is shown in Table 1, along with the RMSE(cv) on the training set (the mean of 100 evaluations with different random folds) and the RMSE on the holdout test set. The two models agree only on three descriptors: SlogP, SlogP_VSA4 and radius. The model built using the RMSE(cv) as objective function has a much better RMSE(cv) (0.786 versus 0.976), as might be expected, but in addition has a better RMSE on the test set (0.934 versus 1.064). The training set and test set predictions for these models are shown in Figure 2.

Figure 3 shows the value of the objective function (either RMSE(cv) or mLOF) for the best model at each iteration for the WAAC algorithm compared to a single optimization phase without any winnowing. For the RMSE(cv) objective function, winnowing not only shortens the time required to converge, but also finds better models. The effect is even more pronounced when using the mLOF function, as without winnowing the algorithm never decreases the number of descriptors

1
2
3 sufficiently to move into the $dp < M$ regime (see Equation 1) thus scoring very poorly.
4
5 The same random seeds were used for corresponding repetitions of the experiments,
6
7 so that the effect observed is not due to different initial models.
8
9

10 In order to assess how the WAAC algorithm compares to the most common stochastic
11 wrapper for feature selection, a genetic algorithm (GA) was implemented in R³¹ (see
12 Appendix A for details) and used to optimize the feature selection (Figure 4a). The
13 default parameter values for the *svm* module in the R package were used ($\epsilon = 0.1$ and
14 $C = 1$). This was compared to results for the WAAC algorithm without performing
15 parameter optimization, and using SVM parameters of $\epsilon = 0.1$ and $C = 1$ (Figure 4b).
16
17 The mean of the minimum values for RMSE(cv) from 10 repetitions was 0.86 for the
18 genetic algorithm, compared to 0.81 when using WAAC. The importance of model
19 parameter optimization can be seen when comparing Figures 3a and 4b. By
20 performing simultaneous feature selection and parameter optimization, the value of
21 the objective function is considerably improved.
22
23
24
25
26
27
28
29
30
31

32 **Discussion**

33
34 Of the two models presented in Table 1, the 12 descriptor model selected by the
35 RMSE(cv) objective function has better generalized performance than the 11
36 descriptor model selected using the mLOF function as measured by the lower RMSE
37 for the prediction of the holdout test set. The 12 descriptors chosen when using the
38 RMSE(cv) objective function may be attributed to the following properties:
39 *hydrophobicity* – SlogP, SlogP_VSA4; *hydrogen bonding* – a_acc; *partially charged*
40 *surface area* – PEOE_PC+, PEOE_VSA+2, PEOE_VSA+3, PEOE_VSA_FPOS;
41 *molecular size* – radius, VadjMa; *others* – a_nN, a_nS, chiral_u. The thermodynamic
42 equilibrium established in a saturated solution may be considered as three separate
43 terms: breakdown of the crystal lattice, creation of a cavity in the solvent, and
44 solvation of the solute molecules.³⁶ At the risk of oversimplifying the underlying
45 physical chemistry, we can rationalize the choice of descriptors in terms of this
46 model. In the absence of other factors, molecules with larger values of SlogP or
47 SlogP_VSA4 will be more hydrophobic and less soluble. An increase in molecular
48 size is expected to increase the energy required for cavity formation in the solvent and
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3 hence decrease solubility. The influence of hydrogen bonds on solubility is less clear
4 as increasing hydrogen bonding increases the interactions with water but can also be
5 expected to increase the energy required to break down the crystal lattice. An increase
6 of partial charge on the molecular surface will tend to increase the strength of solute-
7 water interactions, but perhaps will also increase the lattice energy.
8
9

10
11
12
13 The first use of artificial ant systems for variable selection in QSAR was the
14 ANTSELECT algorithm of Izrailev and Agrafiotis.³⁷ The ANTSELECT algorithm
15 involves the movement of a single ant through feature space. Initially equal weights
16 are assigned to each descriptor. The probability of the ant choosing a particular
17 descriptor in the next iteration is the weight for that descriptor divided by the sum of
18 all weights. After the fitness of the model is assessed, all of the weights are reduced
19 by multiplying by $(1-\rho)$, where ρ is the evaporation coefficient. The weights of those
20 descriptors selected in the current iteration are then increased by a constant multiple
21 of the fitness score. Gunturi et al.³⁸ used a modification of the ANTSELECT
22 algorithm in a recent study of human serum albumin binding affinity in which the
23 number of features selected was fixed *a priori* and, in addition, could not include
24 descriptors that had a correlation coefficient greater than 0.75.
25
26
27
28
29
30
31
32
33
34

35
36 Since the ANTSELECT algorithm uses only a single ant, it cannot make use of one of
37 the most important features of ant colony algorithms, collective intelligence. Instead,
38 premature convergence will occur due to positive reinforcement of models that have
39 performed well earlier in the local search. In addition, the search space will be poorly
40 covered. Although the authors recommend that the algorithm should be repeated
41 several times to minimize the likelihood of convergence to a poor local minimum, the
42 use of an ant colony is a much more robust solution.
43
44
45
46
47
48

49
50 Shen et al.²⁶ presented an ACO algorithm that differed from ANTSELECT in several
51 ways. Their algorithm, which they called a modified ACO, is similar to our WAAC
52 algorithm in that it involves a colony of ants, each of which remembers its best model
53 and score, as well as its current model and score. In Shen et al.'s algorithm, for every
54 descriptor there are both positive and negative weights. The probability that an ant
55 will choose a particular descriptor is given by the positive weight for that descriptor
56
57
58
59
60

1
2
3 divided by the sum of the positive and negative weights. After every iteration, the
4 weights are reduced by multiplying by $(1-\rho)$ as for ANTSELECT. The positive weight
5 for a particular descriptor is increased by the sum of the fitness scores of all ants in
6 the current iteration that have selected it, as well as the fitness scores of the best
7 models of all ants that have selected it in that model. Similarly, the negative weight
8 for a particular descriptor is decreased by an amount based on the fitness scores of
9 models that have not selected it.

10
11
12 Both the modified ACO and ANTSELECT determine probabilities by summing
13 weights based on fitness scores. However, we observed that as convergence is
14 achieved the fitness of the ant models in a particular iteration differ very little from
15 each other. Thus, WAAC uses the fraction of the number of ants that have chosen a
16 particular descriptor rather than a function of the fitness of the ants that have chosen
17 that feature. Another problem with the use of weights is that they tended to increase
18 monotonically over the course of the algorithm whereas the sum of the number of ants
19 has a clear bound. In addition, WAAC uses a value for ρ of 1, that is, complete
20 evaporation. Values less than 1 were found to delay convergence without any
21 corresponding improvement in the result. This makes sense when we consider that the
22 evaporation parameter is supposed to help strike a balance between exploitation of
23 information on previous models (global search) and exploration of local feature space
24 (local search). However, this aspect is already included in Shen et al.'s algorithm and
25 WAAC by the influence of the best models (global search) and current models (local
26 search) on the moving probabilities. As a result of this simpler approach, the moving
27 probabilities now have a meaningful interpretation: the probability of choosing a
28 particular descriptor in the next iteration is equal to the fraction of ants that have
29 chosen that descriptor in their current/best model.

30
31
32 The choice of what objective function to use should take into account the nature of the
33 algorithm. If an objective function is chosen which does not explicitly penalize the
34 number of descriptors but only does so implicitly, irrelevant descriptors may
35 accumulate in the converged model. When using such a function, the winnowing
36 procedure implemented in WAAC plays an important role in removing these
37 descriptors after the optimization phase by initiating a new search of a reduced feature
38 space.

space which makes it less likely that irrelevant descriptors will be selected.

In addition, the variance of the objective function should be taken into account. Since n -fold cross validation involves the random splitting of the data into n folds of equal size, repeated evaluations of the RMSE(cv) will vary somewhat if the data is not very homogeneous. Near the end of each optimization phase, the majority of ants converge to the same feature selection and parameter values, causing the same model to be repeatedly evaluated. Since for each ant the best score is retained, the value of the objective function tends towards the optimistic tail of the distribution of values of the RMSE(cv). For our data, when the RMSE(cv) was recalculated for each of highest scoring models from the 10 repetitions of the WAAC algorithm by taking the mean of 100 repeated evaluations of the RMSE(cv), the value for the objective function (0.724) was found to be overoptimistic by about 0.06 Log S units. We believe that this will not affect the result of the feature selection or parameter optimization as it only occurs once the majority of the ants' models have converged. A more sophisticated treatment would avoid re-evaluating a model that had already been scored but instead would just use the previous value. Alternatively, the mean of multiple evaluations of the RMSE(cv) for a particular model could be used, although this would decrease the speed of the algorithm by the same multiple.

An alternative is to use an objective function that explicitly penalizes the number of descriptors. In this case, it is necessary to adjust the cost term based on some *a priori* knowledge of the number of descriptors desired in the model. Functions like the mLOF used here and the objective function used by Shen et al. quickly force models into a reduced feature space by favoring models with fewer descriptors. However, the moving probabilities used to choose descriptors will be misleading as they will largely be based on those descriptors present in models with fewer descriptors rather than those with the best predictive ability. As a result, descriptors with good predictive ability may be removed by chance.

It should be noted an objective function that simply optimizes a measure of fit to the training data is not a suitable choice for the development of a model with predictive ability. Optimizing the RMSE on the training data, RMSE(train), or optimizing the R^2

1
2
3 value, will produce an overfitted model that fits the training data exceptionally well
4 but performs poorly on unseen data.
5
6
7

8 9 **Conclusion**

10 The key elements to developing an effective QSAR model for prediction are accurate
11 data, relevant descriptors and an appropriate model. Where there is no *a priori*
12 information available on relevant descriptors, some form of feature selection needs to
13 be performed.
14
15
16
17
18

19 We have presented WAAC, an extension of the modified ACO algorithm of Shen et
20 al.,²⁶ which can perform simultaneous optimization of feature selection and model
21 parameters. In addition, the moving probabilities used by the algorithm are easily
22 interpreted in terms of the best and current models of the ants, and our winnowing
23 procedure promotes the removal of irrelevant descriptors. Although we have only
24 examined here the optimization of SVM models with WAAC, our algorithm can be
25 used in conjunction with a broad range of machine learning and regression methods.
26
27
28
29
30
31
32
33

34 We have also shown that the choice of an objective function is not independent from
35 the choice of a feature selection method. For use with WAAC, we recommend an
36 objective function such as the RMSE(cv) that does not explicitly penalize the number
37 of descriptors.
38
39
40
41
42

43 **Acknowledgements**

44 We thank the BBSRC (NMOB & JBOM - grant BB/C51320X/1) and Pfizer (DSP –
45 through the Pfizer Institute for Pharmaceutical Materials Science) for funding, and
46 Unilever for supporting the Centre for Molecular Science Informatics. NMOB thanks
47 Dr. Jen Ryder, Daniel Almonacid and Dr. Avril Coghlan for helpful comments on the
48 manuscript.
49
50
51
52
53
54
55
56
57
58
59
60

Supporting Information Available

The descriptor values for the training and test sets are available as text files in R format. In addition, the full dataset is available as an MDL SD file. This material is available free of charge *via* the Internet at <http://pubs.acs.org>.

Appendix A

Genetic algorithm

50 chromosomes were randomly initialized so that each chromosome on average corresponded to a model based on half of the descriptors. A selection operator chose 10 chromosomes using tournament selection with tournaments of size 3. Once selected, that chromosome was removed from the pool for further selection. A crossover operator was applied to the selected chromosomes, as a single-point crossover between randomly selected (with replacement) chromosomes yielding a pair of children in each case. Each child was subject to a mutation operator which, for a given bit on a chromosome, had a probability of 0.04 of flipping it. The process of crossover and mutation was repeated until 50 offspring were created. The next generation was then formed by the 25 best chromosomes in the original population along with the best 25 of the offspring.

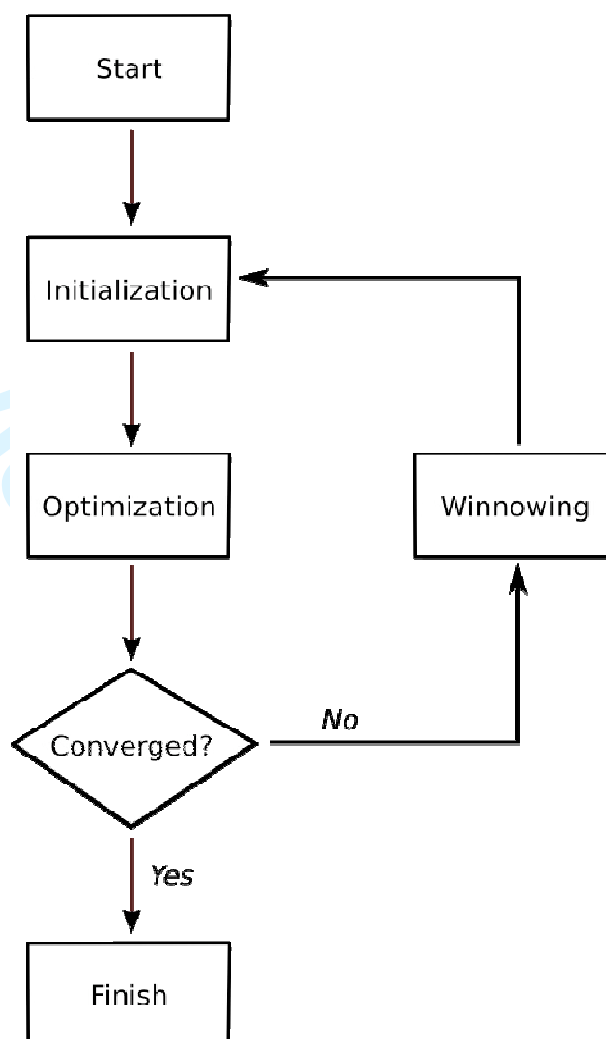
Figures

Figure 1 – Outline of the WAAC algorithm

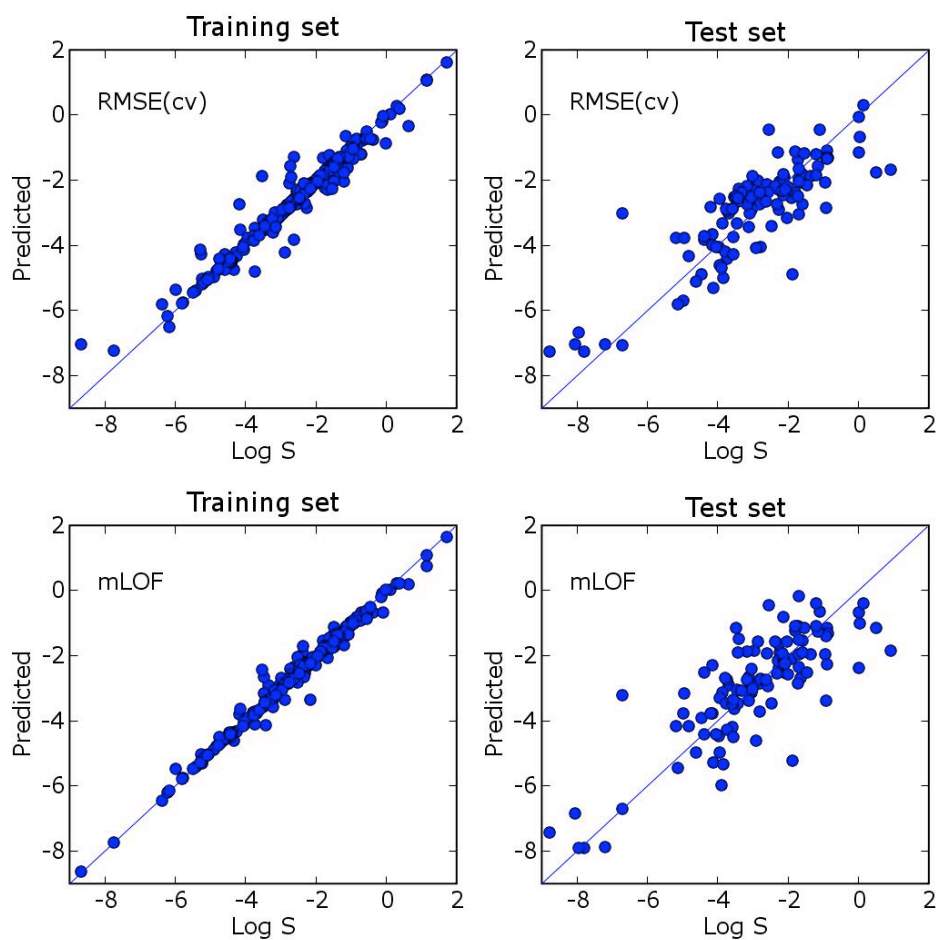


Figure 2 – The prediction of the test and training set data for the best model from 10 iterations of the WAAC algorithm using the RMSE(cv) as the objective function (top) and the mLOF function (bottom). The line $x = y$ is shown for comparison.

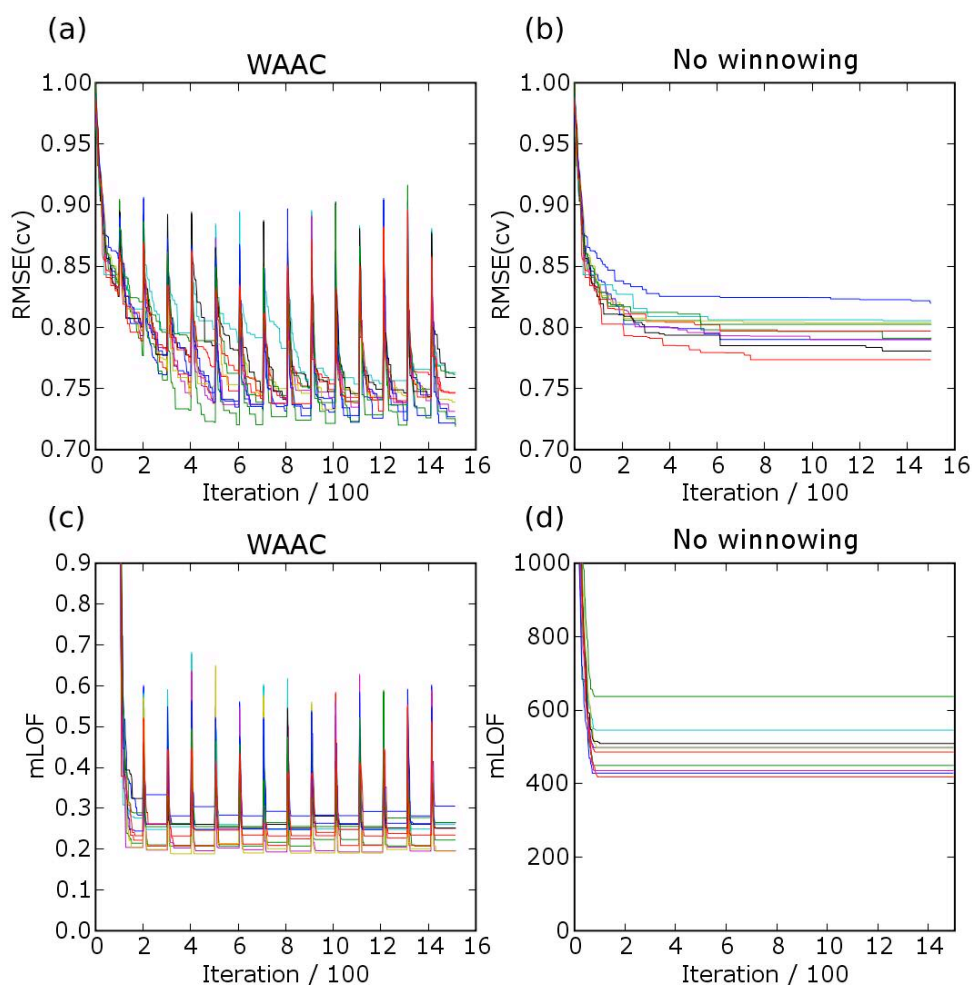


Figure 3 – The value of the objective function is shown for the best model found in a particular iteration of the WAAC algorithm, with (on the left), and without (on the right), winnowing. Ten repetitions of the algorithm are shown, with corresponding repetitions starting from the same initial random seed. For (a) and (b), the objective function is the RMSE(cv); for (c) and (d), mLOF is used.

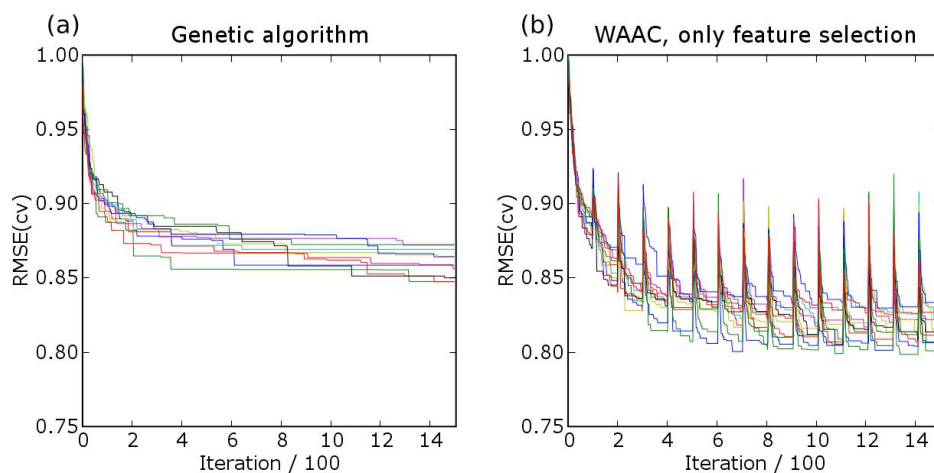


Figure 4 – The value of the RMSE(cv) objective function for the best model found in a particular iteration of a genetic algorithm and WAAC (without parameter optimization). Ten repetitions of each algorithm are shown, with corresponding repetitions starting from the same initial random seeds as used for the experiments shown in Figure 3.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Tables

Table 1

<i>Descriptors</i>	<i>RMSE(cv)</i>	<i>mLOF</i>
radius	x	x
GCUT_PEOE_0		x
GCUT_PEOE_1		x
chiral_u	x	
a_nN	x	
a_nS	x	
VAdjMa	x	
PEOE_PC+	x	
PEOE_RPC		x
PEOE_VSA+2	x	
PEOE_VSA+3	x	
PEOE_VSA-0		x
PEOE_VSA_FPOS	x	
lip_don		x
a_acc	x	
vsa_acc		x
SlogP	x	x
SlogP_VSA4	x	x

<i>Descriptors</i>	<i>RMSE(cv)</i>	<i>mLOF</i>
SMR_VSA3		x
SMR_VSA7		x
cost	3.8	5
epsilon	0.04	0.03
number of descriptors	12	11
mean of 100 RMSE(cv)	0.786	0.976
RMSE(test)	0.934	1.064

References

1. Hansch, C.; Maloney, P. P.; Fujita, T.; Muir, R. M. Correlation of biological activity of phenoxyacetic acids with Hammett substituent constants and partition coefficients. *Nature* **1962**, *194*, 178-180.
2. Hawkins, D. M. The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1-12.
3. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157-1182.
4. Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493-500.
5. MOE v2007.04. Chemical Computing Group Inc., Montreal, Quebec, Canada, **2006**. <http://www.chemcomp.com>
6. SYBYL 7.1. Tripos Inc., 1699 Hanley Road, St. Louis, MO 63144, **2006**. <http://www.tripos.com>
7. John, G. H.; Kohavi, R.; Pfleger, K. Irrelevant features and the subset selection problem. *Machine learning: Proc. of the 11th Intern. Conf.* **1994**, 121-129.
8. Kohavi, R.; John, G. H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273-324.
9. Dudek, A. Z.; Arodz, T.; Gálvez, J. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Comb. Chem. High Through. Screen.* **2006**, *9*, 213-228.
10. Liu, Y. A comparative study on feature selection methods for drug discovery. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1823-1828.
11. Whitney, A. W. A direct method of nonparametric measurement selection. *IEEE Trans. Comput.* **1971**, *20*, 1100-1103.
12. Marill, T.; Green, D. M. On the effectiveness of receptors in recognition systems. *IEEE Trans. Inform. Theory* **1963**, *9*, 11-17.

13. Pudil, P.; Novovičová, J.; Kittler, J. Floating search methods in feature selection. *Patt. Recog. Lett.* **1994**, *15*, 1119-1125.
14. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* **2002**, *46*, 389-422.
15. Fröhlich, H.; Wegner, J. K.; Zell, A. Towards optimal descriptor subset selection with support vector machines in classification and regression. *QSAR Comb. Sci.* **2004**, *23*, 311-318.
16. Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Kluwer Academic Publishers: Boston, MA, 1989.
17. Rogers, D.; Hopfinger, A. J. Application of genetic function approximation to quantitative structure-activity relationships and quantitative structure-property relationships. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 854-866.
18. Wegner, J. K.; Zell, A. Prediction of aqueous solubility and partition coefficient optimized by a genetic algorithm based descriptor selection method. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1077-1084.
19. von Homeyer, A. In *Handbook of Chemoinformatics*; Gasteiger, J., Ed.; Wiley-VCH: Weinheim, 2003; Vol. 3, Section IX, Chapter 1.6, pp 1239-1280.
20. Agrafiotis, D. K.; Cedeno, W. Feature selection for structure-activity correlation using binary particle swarms. *J. Med. Chem.* **2002**, *45*, 1098-1107.
21. Lin, W.-Q.; Jiang, J.-H.; Shen, Q.; Shen, G.-L.; Yu, R.-Q. Optimized block-wise variable combination by particle swarm optimization for partial least squares modeling in quantitative structure-activity relationship studies. *J. Chem. Inf. Model.* **2005**, *45*, 486-493.
22. Guha, R.; Jurs, P. C. Development of linear, ensemble and nonlinear models for the prediction and interpretation of the biological activity of a set of PDGFR inhibitors. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2179-2189.
23. Vapnik, V. N. *The nature of statistical learning theory*. Springer Verlag: New York, 1995.
24. Hastie, T.; Tibshirani, R.; Friedman, J. *The elements of statistical learning: data mining, inference, and prediction*. Springer: New York, 2001.

- 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
25. Smola, A. J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199-222.
 26. Shen, Q.; Jiang, J.-H.; Tao, J.-C.; Shen, G.-L.; Yu, R.-Q. Modified Ant Colony Optimization Algorithm for Variable Selection in QSAR Modeling: QSAR Studies of Cyclooxygenase Inhibitors. *J. Chem. Inf. Model.* **2005**, *45*, 1024-1029.
 27. Goss, S.; Aron, S.; Deneubourg, J. L.; Pasteels, J. M. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* **1989**, *76*, 579-581.
 28. Dorigo, M.; Di Caro, G.; Gambardella, L. M. Ant algorithms for discrete optimization. *Artif. Life* **1999**, *5*, 137-172.
 29. Chen, X.-Q.; Cho, S. J.; Li, Y.; Venkatesh, S. Prediction of aqueous solubility of organic compounds using a quantitative structure-property relationship. *J. Pharm. Sci.* **2002**, *91*, 1838-1852.
 30. Rytting, E.; Lentz, K. A.; Chen, X.-Q.; Qian, F.; Venkatesh, S. Aqueous and cosolvent solubility data for drug-like organic compounds. *AAPS J.* **2005**, *7*, E78-E105.
 31. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, **2006**. <http://www.R-project.org>
 32. Bergstrom, C. A. S.; Wassvik, C. M.; Norinder, U.; Luthman, K.; Artursson, P. Global and local computational models for aqueous solubility prediction of drug-like molecules. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1477-1488.
 33. Waller, C. L.; Bradley, M. P. Development and validation of a novel variable selection technique with application to multidimensional quantitative structure-activity relationship studies. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 345-355.
 34. Hawkins, D. M.; Basak, S. C.; Mills, D. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 579-586.
 35. Friedman, J. H. Multivariate Adaptive Regression Splines. *Ann. Stat.* **1991**, *19*, 1-67.
 36. Yalkowsky, S. H. *Solubility and solubilization in aqueous media*. Oxford

1
2
3 University: Oxford, 1999.
4

5
6 37. Izrailev, S.; Agrafiotis, D. K. Variable selection for QSAR by artificial ant
7 colony systems. *SAR QSAR Environ. Res.* **2002**, *13*, 417-423.
8

9
10 38. Gunturi, S. B.; Narayanan, R.; Khandelwal, A. In silico ADME modelling
11 2: Computational models to predict human serum albumin binding affinity
12 using ant colony systems. *Bioinorg. Med. Chem.* **2006**, *14*, 4118-4129.
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60