Drug Design, Artificial Intelligence Methods in

OVIDIU IVANCIUC
Department of Biochemistry and Molecular Biology,
University of Texas Medical Branch, Galveston, USA

Article Outline

Glossary

Ant colony optimization Ant colony optimization (ACO) is an agent-based algorithm procedure inspired by the function of ant colonies and their search for the optimum path to food sources. The virtual agents are called artificial ants or ants, and the optimization problem is represented as a trail-and-error search for the optimum path on a weighted graph. The pheromone that is deposited by ants on the trail is represented as weights for graph components (vertices or edges). Each ant generates a solution by moving on the graph and by selecting the next step based on the pheromone level. The pheromone level is updated after each cycle (when all ants found a solution) by adding a pheromone quantity proportional to the quality of the solutions to which it belongs.

Antigen An antigen is a molecule (chemical compound, protein or polysaccharide) that induces an immune response. Each pathogen contains specific antigens that are recognized by the immune system. The antigen region that is recognized by the immune system is called an epitope.

Antibody An antibody (or immunoglobulin) is a protein used by the immune system to identify bacteria, viruses and other pathogens or foreign molecules. The antibody region that binds antigens is extremely variable, thus allowing the immune system to recognize a large diversity of pathogens. The ability to recognize antigens is improved through successive cycles of antigen presentation, antibody cloning, and hypermutation of the variable region of the antibody.

Artificial immune systems Artificial immune systems (AIS) represent a class of optimization algorithms inspired by the components and mechanisms of the biological immune system. AIS simulate the learning and memory capabilities of the immune system to develop computational algorithms for pattern recognition, function optimization, classification, process control, and intrusion detection.

Genetic algorithms Genetic algorithms (GA) solve high-dimensional problems through a Darwinian evolution of a population of individuals, in which each individual (chromosome) represents a possible solution. Depending on the type of the optimization problem, chromosomes may represent the solution in a binary, continuous, or hybrid encoding. Each chromosome has a fitness value that measures the quality of the solution. A population of parents evolves to a generation of children by crossover and mutation.

Particle swarm optimization Swarm intelligence (SI) represent a group of distributed intelligence algorithms that solve optimization problems by applying processes inspired by swarming, herding, and flocking of various species. Particle swarm optimization (PSO) simulates the swarming behaviors observed in swarms of bees, flocks of birds, or schools of fish. PSO considers a swarm of particles that start from a random position and have a random velocity. At each step a particle moves to a new position that is determined by its own experience (the best past position) and by the memory of the best particle in the swarm. PSO may be applied to both binary and continuous optimization problems, and its main strength is a fast convergence.

Quantitative structure-activity relationships Quantitative structure-activity relationships (QSAR) represent regression models that define quantitative correlations between the chemical structure of molecules and their physical properties (boiling point, melting point, aqueous solubility), chemical properties and reactivities (chromatographic retention, reaction rate), or biological activities (cell growth inhibition, enzyme inhibition, lethal dose). The fundamental hypotheses of QSAR is that similar chemicals have similar properties, and small structural changes result in small changes in property values. The general form of a QSAR equation is $P(i) = f(\mathbf{SD}_i)$, where $P(i)$ is a physical, chemical, or biological property of compound i, $\mathbf{SD}_i$ is a vector of structural descriptors of i, and f is a mathematical function such as linear regression, partial least squares, artificial neural networks, or support vector machines. A QSAR model for a property P is based on a dataset of chemical compounds

with known values for the property P, and a matrix of structural descriptors computed for all chemicals. The learning (training) of the QSAR model is the process of determining the optimum parameters of the regression function f. After the training phase, a QSAR model may be used to predict the property P for novel compounds that are not present in the learning set of molecules.

Structural descriptor A structural descriptor (SD) is a numerical value computed from the chemical structure of a molecule, which is invariant to the numbering of the atoms in the molecule. Structural descriptors may be classified as constitutional (counts of molecular fragments, such as rings, functional groups, or atom pairs), topological indices (computed from the molecular graph), geometrical (volume, surface, charged-surface), quantum (atomic charges, energies of molecular orbitals), and molecular field (such as those used in CoMFA, CoMSIA, or CoRSA).

Structure-activity relationships Structure-activity relationships (SAR) represent classification models that can discriminate between sets of chemicals that belong to different classes of biological activities, usually active/inactive towards a certain biological receptor. The general form of a SAR equation is $C(i) = f(\mathbf{SD}_i)$, where $C(i)$ is the activity class of compound i (active/inactive, inhibitor/non-inhibitor, ligand/non-ligand), $\mathbf{SD}_i$ is a vector of structural descriptors of i, and f is a classification function such as k-nearest neighbors, linear discriminant analysis, random trees, random forests, Bayesian networks, artificial neural networks, or support vector machines.

Definition of the Subject

Drug design and development represents a complex and expensive process that is based on the creative application of scientific results from various disciplines, including genomics, chemistry, biology, computational chemistry, pharmacology, toxicology, and clinical studies. The average cost of bringing a new drug to market is currently around US$800 million, with a large part of the cost coming from chemical compounds that fail in different stages of development. Computational simulation of biochemical processes may guide the drug discovery process through reliable in silico models of biochemical properties (aqueous solubility, octanol-water partition, intestinal absorption, blood-brain barrier transport, excretion), prediction of enzyme-ligand interactions, simulations of cells, tissues and organisms. In this chapter we review the most important applications of artificial intelligence in structure-activity relationships (SAR) and quantitative structure-activity relationships (QSAR). These techniques are used in different stages of drug design, including large scale screening of chemical libraries, optimization of protein-ligand interactions, modeling the drug transport through membranes, prediction of drug metabolism, mutagenicity, and carcinogenicity. The common goal of artificial intelligence applications in computer-assisted drug design is to identify the best candidates in each step, which may eventually lead to reduced costs for the development new drugs.

Introduction

Biology is a rich source of inspiration for developing algorithms that solve complex problems by emulating mechanisms and functions of biological systems. Well-known examples of biologically inspired algorithms are artificial neural networks, genetic algorithms, ant colony optimization, DNA computing, particle swarm optimization, and artificial immune systems.

Evolutionary algorithms represent a family of stochastic methods that solve optimization problems by evolving solutions based on Darwinian evolution and concepts of DNA genetics (for details on GA and evolutionary algorithms, see "*Genetic and Evolutionary Algorithms and Programming*")CE2. The main algorithms from this class are genetic algorithms (GA), genetic programming (GP), and evolutionary programming (EP). The major principles of genetic algorithms were developed by Holland [1], and then further developed by Goldberg [2]. Many applications of chemoinformatics and computational chemistry have a large search space that must be explored to locate the solution. Usually, the brute-force grid search approach cannot be applied but for small systems, and various stochastic methods were developed to find near-optimal solutions. Several examples of high-dimensional problems are the prediction of the biopolymer structure from sequence (peptides, proteins, DNA, RNA), protein-protein docking, protein-ligand docking, conformational search, geometry optimization, design of chemical libraries, and design of chemical compounds with special physico-chemical and biological properties. For other GA applications in chemistry and biology see the reviews by Jones [3], Terfloth [4], and von Homeyer [5]. The most important GA applications in drug development are reviewed in Sect. "Genetic Algorithms".

Dorigo and co-workers developed the ant colony optimization (ACO) algorithm to mimic the foraging behavior of some ant species [6,7,8,9,10]. The main feature modeled

CE2 Please confirm change.

Editor's or typesetter's annotations (will be removed before the final TEX run)

in ACO is the ability of an ant population to find the shortest path to a food source using as guide the pheromone trace that is deposited on the path explored by each individual ant. The pheromone accumulates on paths explored more frequently by ants, which indicates that the paths are shorter routes to the food source. ACO has numerous applications, mainly in combinatorial optimization, when their ability to explore large solutions spaces is a clear advantage. For theoretical details and applications of agent based simulation, see ► Agent Based Modeling and Simulation, ► Agent Based Modeling Formalisms, Mathematics of, ► Agent Based Modeling and Agent Based Modeling Platforms, Design of, ► Agent Based Modeling and Artificial Life, and ► Agent Based Modeling and System Biology. In Sect. "Ant Colony Optimization" we present an overview of ACO applications in drug design.

The particle swarm optimization (PSO) algorithm proposed by Kennedy and Eberhart is inspired by the social behavior of large groups of individuals, such as bird flocking, fish schooling, and animal herding [11]. Each individual of the group, represented as a particle that moves with a particular velocity through the search space, is a solution for the optimization problem. The movement of each particle is determined by the best position visited by the particle, and the best position found by the group. The balance between a local and a global search is introduced by weighting the attraction of the best solution of the particle and the best solution of the swarm (for more details on the PSO algorithms, see ► Swarm Intelligence and ► Multi-agent Systems: Swarms). PSO converges fast and may be used with success to explore high dimensional spaces. The algorithm is simple, with a small number of parameters, and the large number of variants proposed in the literature is a sign of the great interest and vigorous research in this field [12,13]. Swarm intelligence algorithms are used in drug design for diverse application, including gene expression [14], enzyme-inhibitor docking [15], selection of structural descriptors for QSAR models [16], QSAR with support vector machines optimized with PSO [17], and modeling enzyme inhibitors with artificial neural networks trained with PSO [18]. The most important PSO applications in drug discovery are presented in Sect. "Particle Swarm Optimization".

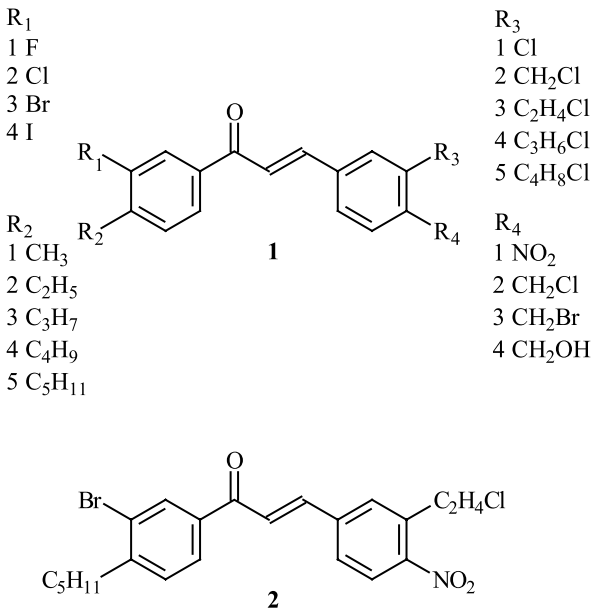
The immune system protects an organism against infection by identifying and killing pathogens. Recognition cells known as B-cells and T-cells identify the pathogens that enter into the human body. Receptors situated on the surface of the B-cells and T-cells recognize and bind proteins and protein fragments from pathogens, thus forming high affinity antigen-antibody complexes. The learning and memory capabilities of the biological immune system are used in a novel class of machine learning algorithms, the artificial immune systems (AIS) [19,20,21,22,23,25,26,27] (for further details on AIS see ► Immunecomputing. The major AIS algorithms and the most important applications are presented in several books and conference proceedings: *Artificial Immune Systems and Their Applications* edited by Dasgupta [28]; *Artificial Immune Systems: A New Computational Intelligence Approach* by de Castro and Timmis [29]; *Immunocomputing: Principles and Applications*, by Tarakanov, Skormin, and Sokolova [30]; *Immunity-Based Systems* by Ishida [31]; *Artificial Immune Systems: ICARIS 2003* edited by Timmis, Bentley and Hart [32]; *Artificial Immune Systems: ICARIS 2004* edited by Nicosia, Cutello, Bentley, and Timmis [33]; *Artificial Immune Systems: ICARIS 2005* edited by Jacob, Pilat, Bentley, and Timmis [34]; *Artificial Immune Systems: ICARIS 2006* edited by Bersini and Carneiro [35].

AIS models were successfully applied to biological and medical problems, such as classification of gene expression data [36,37,38], identification of breast cancer [39], diagnosis of lung cancer [40], recognition of ECG arrhythmia [41], and interpretation of carotid artery Doppler signals [42]. Protein structure prediction starting from the amino acids sequence is a difficult and computationally intensive task, which was investigated with AIS for models based on Dill's hydrophobic-hydrophilic lattice approach [43] and with three-dimensional models [44]. In Sect. "Artificial Immune Systems" we present a review of the AIS applications in drug design and toxicology.

Genetic Algorithms

Compared with other families of artificial intelligence algorithms, evolutionary algorithms are by far the most popular, with the largest number of publications and with the most diverse applications. GA methods are applied with success to solve diverse drug design problems, such as protein-ligand docking [45], structure-based drug design [46], global optimization of QSAR models based on artificial neural networks [47], computer-aided molecular design [48,49], design of combinatorial libraries [50], and feature selection in QSAR models [51,52]. All these problems are difficult to solve thorough a brute force approach due to the huge search space, but GA are very efficient in finding their global optimum with modest computational resources.

Evolutionary algorithms are applied with success in computer-aided molecular design [48,49] to generate novel molecules with prescribed physical, chemical, or biological properties. In pharmaceutical applications,



Drug Design, Artificial Intelligence Methods in, Figure 1
General formula for a family of chemical compounds (1) that may be encoded with chromosomes having four elements, and an example of molecule (2) from this family

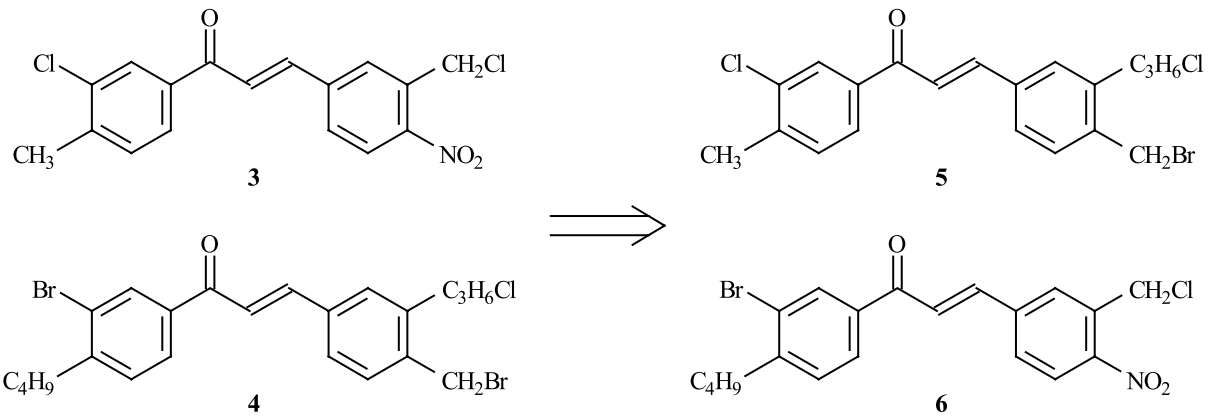
molecular design is focused on discovering chemical structures that can satisfy all requirements of a successful drug, such as affinity and selectivity for the biological target, and good ADME-Tox (absorption, distribution, metabolism, excretion, and toxicity) properties. The most important part of any molecular design application is a proper encoding of the molecular structure into a chromosome. A straightforward translation of chemicals may be achieved if the molecule can be partitioned into a constant skeleton and a series of substituents, such as the family of chemical compounds **1** (Fig. 1) that has four substituents R_1, R_2, R_3, and R_4. Each molecule from this family may be encoded by a chromosome with four elements ($/R_1/R_2/R_3/R_4/$), with each element recording the index of the respective substituent. Each substitution position has a set of allowed substituents encoded with numbers. For example, compound **2** is represented by the chromosome /3/5/3/1/.

Using this molecular encoding, one can easily define the GA operations of crossover and mutation. The crossover operation involves the exchange of substituents between two parent molecules. For example, parent molecules **3** and **4** generate child molecules **5** and **6** by exchanging substituents R_3 and R_4 (Fig. 2).

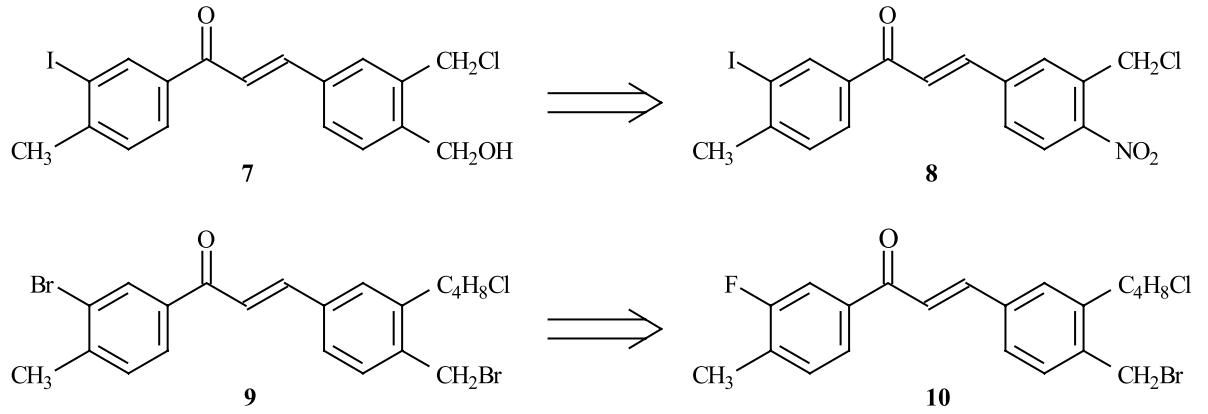
The chemical space is also explored with the substituent mutation, as shown in Fig. 3: parent molecule **7** generates child molecule **8** by mutating R_4, and parent molecule **9** generates child molecule **10** by mutating R_1. These examples demonstrate the encoding and evolution of chemical structures in combinatorial libraries of chemical compounds [53,54]. The progress in combinatorial chemistry [55,56], virtual screening of chemical libraries, and high throughput techniques dramatically increased the chemical space that can be explored in the quest for molecules with special properties (peptides, nucleic acids, catalysts, pesticides, drugs). Due to the huge chemical space that can be generated through combinatorial chemistry, it is rarely possible to perform an exhaustive synthesis of all possible chemical species. Instead, GA implementations are used to guide the chemical synthesis towards regions containing molecules with target properties [57,58].

An efficient class of reactions that may generate large combinatorial libraries is the Ugi multicomponent reaction (MCR) [59]. Ugi MCRs are one-pot reactions in which three reactants (Fig. 4; U-3CR), four reactants (Fig. 5; U-4CR), or more reactants are converted into the corresponding product without separation and purification of the intermediates. The diversity of chemical structures generated thorough MCR reactions comes from the diversity of the groups R from reactants. Using available chemicals, one can design chemical libraries that are too large to synthesize. Instead, a sample of the combinatorial library is synthesized and evaluated in biological assays, followed by an in silico exploration based on GA models [60]. If each reactant type in an U-3CR is a set of 1000 different chemical compounds, then the complete library has 10^9 distinct molecular structures. Similarly, an U-4CR library generated from four sets of 1000 chemicals each consists of 10^{12} distinct compounds. It becomes apparent that the vast chemical space available through combinatorial synthesis is too large even for the in silico exploration, which explains why evolutionary algorithms are used to guide the chemical synthesis.

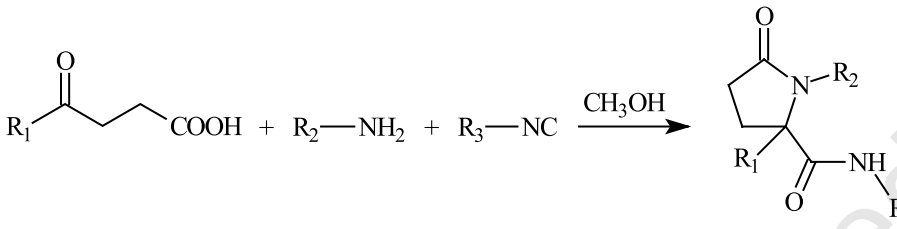
The GA translation of MCR reactions and other combinatorial libraries is straightforward, and in many cases the in silico exploration of the chemical space may be performed with standard GA software. This approach has limitations, because the size of the chemical space is fixed by the initial sets of reactants, and the common skeleton remains constant during the simulation. Graph-based GA models solve these limitations by generating molecular structures that are not programmed in the starting building blocks. Such GA systems have crossover and mutation procedures that operate directly on the molecular graph [61], and define a chromosome structure capable to encode a molecular graph [62]. The graph-based GA sys-



Drug Design, Artificial Intelligence Methods in, Figure 2
Example of molecule crossover: parent molecules 3 and 4 generate child molecules 5 and 6 by exchanging substituents R_3 and R_4 (see molecule 1)

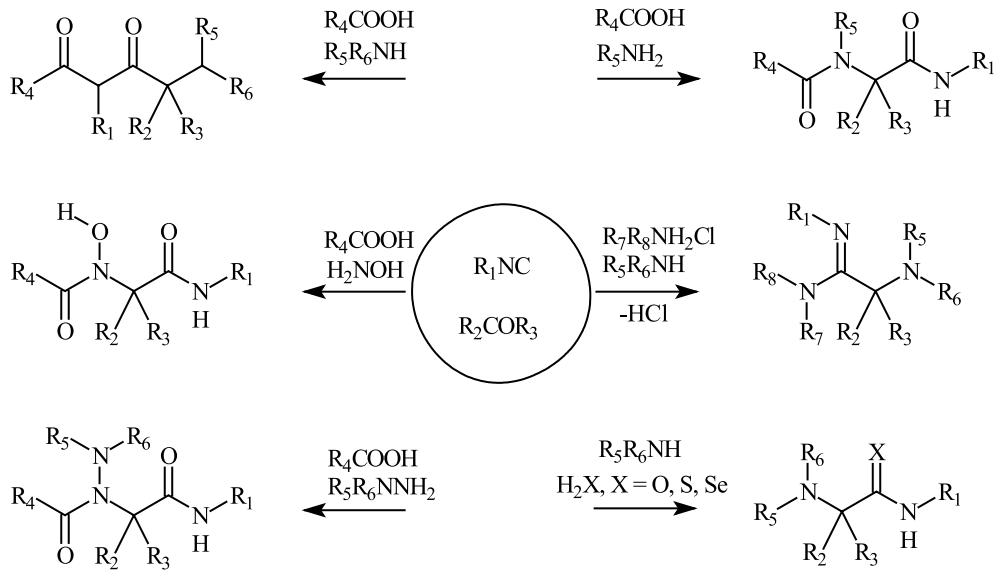


Drug Design, Artificial Intelligence Methods in, Figure 3
Examples of molecule mutation: parent molecule 7 generates child molecule 8 by mutating R_4, and parent molecule 9 generates child molecule 10 by mutating R_1

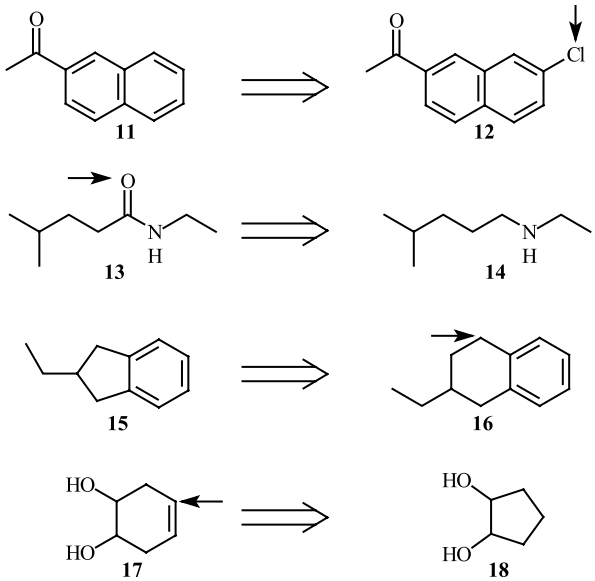


Drug Design, Artificial Intelligence Methods in, Figure 4
Example of Ugi 3-component reactions (U-3CR)

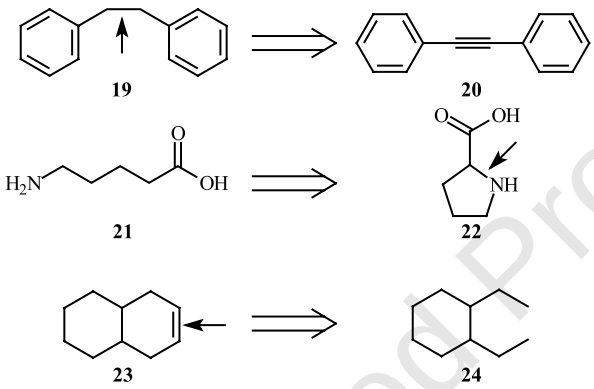
tem proposed by Brown et al. introduces novel crossover and mutation operations for molecular graphs [62] in order to solve the inverse QSAR problem, i. e., to design new chemicals starting from structure-activity models [63]. Four mutations operate on atoms (graph nodes), namely append, prune, insert, and delete (Fig. 6; the site of the transformation is indicated with an arrow). The append mutation adds an atom and its chemical bond to the molecular graph (**11**→**12**). The connecting atom is selected at random from the set of atoms in the molecule



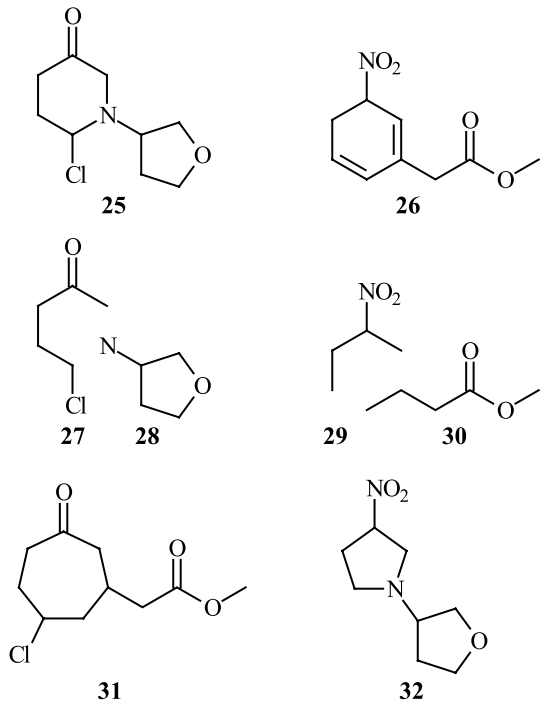
Drug Design, Artificial Intelligence Methods in, Figure 5
Examples of Ugi 4-component reactions (U-4CR)

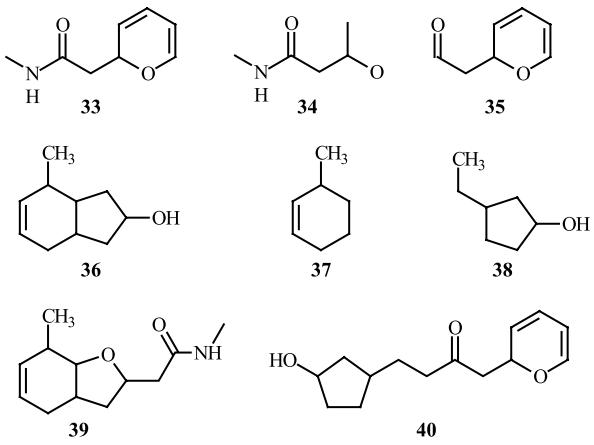


Drug Design, Artificial Intelligence Methods in, Figure 6
Examples of the node mutation operators: 11→12, append, 13→14, prune, 15→16, insert, and 17→18, delete



Drug Design, Artificial Intelligence Methods in, Figure 7
Examples of the edge mutation operators: 19→20, substitute, 21→22, add, and 23→24, delete





Drug Design, Artificial Intelligence Methods in, Figure 8
Example of the multiple crossover: 25 and 26, parent molecules; 27 and 28, disconnected subgraphs of 25; 29 and 30, disconnected subgraphs of 26; 31, child molecule generated from subgraphs 27 and 30; 32, child molecule generated from subgraphs 29 and 28

Drug Design, Artificial Intelligence Methods in, Figure 9
Example of the subgraph crossover: the first parent molecule 33 and its two induced connected subgraphs 34 and 35; the second parent molecule 36 and its two induced connected subgraphs 37 and 38; the first child molecule 39 generated from subgraphs 34 and 37; the second child molecule 40 generated from subgraphs 35 and 38

that have available valences. The type of the connecting bond is randomly selected from the possible types for the two atoms. The prune mutation removes a terminal atom from the molecular graph (**13**→**14**). The insert mutation selects a bond in the molecular graph, cuts it and inserts a molecular fragment between the two disconnected atoms (**15**→**16**). The molecular fragment is selected from a library, and may consist of a single atom or a more complex subgraph. Additional tests are performed to ensure that the final chromosome (molecular graph) is a valid chemical structure. The delete mutation selects an atom at random, removes it and reconnects the molecular graph (**17**→**18**). The edge mutations operate on the set of edges in a chromosome (Fig. 7; the site of the transformation is indicated with an arrow). The substitute mutation selects randomly an edge and then replaces changes its type to another bond type (**19**→**20**). The mutation result must correspond to a correct chemical structure. The add mutation adds a new bond between two atoms (**21**→**22**), thus making possible the generation of cyclic structures. Finally, the delete mutation deletes a bond that was randomly selected (**23**→**24**). The resulting chromosome must represent a connected molecular graph. Two crossover mutations are defined for molecular graphs, i. e., multiple crossover and subgraph crossover. The multiple crossover starts from two parent molecules, then each parent molecule is disconnected into two subgraphs, and finally, two child molecules are generated by swapping subgraphs from the parent molecules (Fig. 8). Parent molecule **25** generates subgraphs **27** and **28**, and parent molecule **26** generates subgraphs **29** and **30**. The crossover operation generates child molecule **31** from subgraphs **27** and **30**, and then assembles child molecule **32** from subgraphs **29** and **28**. In the subgraph crossover a connected subgraph is selected in each parent molecule, and then the subgraphs are combined to obtain the first child molecule (Fig. 9). The combination of the two fragments tries to retain the topology of the two subgraphs. In the second step a different subgraph is induced in each parent molecule, and the two subgraphs form the second child molecule. The parent molecule **33** generates the induced connected subgraphs **34** and **35**; and the second parent molecule **36** generates the induced connected subgraphs **37** and **38**. The first child molecule **39** is obtained by combining subgraphs **34** and **37**, and the second child molecule **40** is obtained from subgraphs **35** and **38**. The main advantage of the graph-based GA system is its ability to explore chemical structures that are not related to the starting molecules, and to discover novel chemical topologies.

Virtual Screening of Chemical Libraries

QSAR models are very useful tools for the identification of structural features that determine various molecular properties, and may even suggest the mechanism of action for biochemical processes. Thus, QSAR models start from structure and correlate descriptors with molecular properties. Once a QSAR model is established, an inverse process becomes possible, namely setting a target value for a molecular property and then finding all possible chemical structures that might exhibit that property value, within a certain range of variation. This process in called inverse QSAR, and it represents an important step in optimizing the drug-like properties of chemical compounds. Lewis proposed an inverse QSAR strategy that may assist medicinal chemists in deciding how to optimize a library of chemical compounds [64]. The starting point is a dataset of chemical compounds with a molecular property, and a corresponding QSAR model. The inverse QSAR strategy involves an iterative application of several steps, namely generation of new structures, structure filtering based on synthetic feasibility or undesired properties, and QSAR filtering. The first step generates a new chemical library by applying simple chemical transformations to the molecules from the initial dataset. Examples of such transformations are modification of the bond order, adding or removing an atom, adding or removing a fragment, or changing C to N or O. The second step filters molecules that have nonspecific reactivity, such as electrophiles, nucleophiles, acylating agents, or redox systems. Synthetic feasibility rules are used to eliminate compounds that are difficult to synthesize or those that are expensive. Finally, QSAR models are used to select candidates for chemical synthesis. The inverse QSAR strategy developed by Lewis was tested for a combinatorial library of 150 inhibitors of human carbonic anhydrase II, that was used to develop a MLR genetic function approximation QSAR, as implemented in Cerius^2. The best QSAR model is based on five structural descriptors: TS3

$$\begin{aligned} \mathrm{pIC}_{50} &= 7.5 - 0.6\mathrm{PHI} - 5.7\mathrm{Jurs\text{-}RPCG} + 0.2\mathrm{SdsN} \\ &\quad + 1.7\mathrm{NaaS} + 0.001\mathrm{Vm} \\ n &= 150 r^2 = 0.81 q^2_{LOO} = 0.80 F = 127 \end{aligned}$$

where PHI is the molecular flexibility index, Jurs-RPCG is the charge of the most positive atom divided by the total positive charge, SdsN is the E-state index for sp^2 N, NaaS is the electrotopological count for aromatic S, and Vm is the molecular volume inside the contact surface. This QSAR was used as the starting point for performing automated property optimization.

Ant Colony Optimization

The classical ACO algorithm was successfully modified and adapted in numerous variants to solve specific problems from chemistry and drug design. By far the most important application is represented by the feature selection for QSAR models [65,66]. Several ACO implementations were tested in diverse QSAR models, including multi-linear regression, artificial neural networks, and regression trees. Clustering is routinely used to discover novelty in large chemical datasets, based on structural similarities measured by molecular descriptors. Since similar chemicals usually have similar properties, clustering may suggest groups of molecules that interact with the same biological target. Shelokar et al. proposed a clustering algorithm based on ACO assignment of objects in clusters [67]. Many biochemical problems require optimization of continuous variables, whereas the classical ACO implementation optimizes discrete variables. He et al. demonstrated an ACO extension to continuous variables that may be applied to identify optimum parameters for QSAR models [68]. Korb and co-workers introduced a new protein-ligand docking algorithm, PLANTS (Protein-Ligand ANT System), which uses ACO to find a minimum energy conformation for the protein-ligand complex [69]. Compared with docking algorithms based on GA, PLANTS is faster and finds a larger number of good solutions.

Izrailev and Agrafiotis used an ACO approach to identify the best regression tree models in QSAR [65]. Each ant represents a regression tree, and the pheromone trail is obtained from a reference tree that represents the topological union of all ant trees simulated. The ACO selection of regression trees was evaluated for three QSAR datasets, namely the antifilarial activity of antimycin analogues, the binding affinities of ligands to benzodiazepine/GABA$_A$ receptors, and the inhibition of dihydrofolate reductase by pyrimidines. Each simulation generated 2000 ant trees and then the tree with the best cross-validation predictions was selected as solution. For all three QSAR datasets the ant tree results were significantly better than those obtained with recursive partitioning and with random trees. Using the same three QSAR datasets, Izrailev and Agrafiotis proposed an ACO procedure (ANTSELECT) for feature selection in artificial neural networks QSAR [66]. A number of 100 independent ANTSELECT simulations were performed for each QSAR dataset, with each simulation containing a population of 2000 ants. Structural descriptors are represented as graph vertices, and an ant generates a path by visiting a number of vertices. All vertices on the path represent the selected structural descriptors that are subsequently used as input to an artificial neu-

TS3 Please check the label of the following equations.

Editor's or typesetter's annotations (will be removed before the final TEX run)

ral network. Features that give good QSAR models receive a larger quantity of pheromones, thus having greater chances to be selected in subsequent iterations. The QSAR results indicate that the ANTSELECT algorithm provides good solutions if the simulations use a sufficient number of ants to evaluate all features in different combinations. A second requirement is to have a pheromone accumulation that distinguishes between good and bas features. Artificial neural networks are sensitive to the input features, and ANTSELECT provides sets of descriptors that result in models with good predictive power.

Nonsteroidal antiinflammatory drugs (NSAID) treat inflammation and pain by inhibiting both cyclooxygenase-1 and cyclooxygenase-2 (COX2). NSAID have serious side effects, such as gastrointestinal ulceration and bleeding, but the observation that acute and chronic inflammation correlates with higher levels of COX2 prompted several drug design studies to identify selective COX2 inhibitors. Shen proposed a novel ACO procedure for feature selection in a QSAR study of 42 COX2 inhibitors [70]. Starting from 85 structural descriptors, the simulation used 100 ants and 200 iterations to select 3 descriptors for the optimum model. The ACO procedure selected a better set of descriptors, compared with a selection made with an evolutionary algorithm.

The drug binding to human serum albumin (HSA) determines its bioavailability, pharmacokinetics, and therapeutic effect. Many drugs are transported by HAS, but only the free drug has pharmacological effect. Gunturi et al. modeled the HAS binding of 94 diverse drugs starting from a pool of 327 structural descriptors [71]. Since the number of descriptors is too large for a multi-linear regression QSAR, an ACO procedure was implemented to select those features that determine HSA binding. The ACO solutions were cross-validated, and the best QSAR equations with five and six descriptors were selected as final models. The importance of each descriptor was evaluated by the frequency of selection in QSAR models, and it was found that HAS binding depends on hydrophobic interactions, solubility, size, and shape.

Tyrosine kinases are enzymes that transfer a phosphate group from ATP to a tyrosine residue in a protein. These enzymes have important functions in diverse cellular processes, such as metabolism, differentiation, growth, apoptosis. Shi et al. developed QSAR models for inhibitors of the epidermal growth factor receptor (EGFR), a cell-surface receptor from the tyrosine kinase family [72]. Mutations affecting EGFR expression or activity could result in cancer. The structure of the 61 EGFR inhibitors was encoded with 50 structural descriptors, and ACO was used to select relevant groups of descriptors. The ant population had 100 individuals trained for 200 iterations. The analysis of the descriptors selected with higher frequency by ants reveals the importance of electronic indices, and suggest that electron-donating groups increase the activity of these EGFR inhibitors.

The ability to distinguish between foreign and self proteins is one of the most important characteristics of the immune system. The major histocompatibility complex (MHC) molecules bind short peptides resulting from intracellular processing of foreign and self proteins. The MHC molecule loaded with the peptide migrates to the cell surface where it interacts with T-cell receptors. There are two classes of MHC molecules: (a) MHC class I, which binds peptides derived from endogenously expressed proteins and (b) MHC class II, which binds peptides derived mainly from exogenous or transmembrane proteins. Karpenko et al. devised a novel procedure to predict peptides that bind to MHC II, by using ACO to identify the optimum alignment of a set of variable length peptides [73]. The multiple alignment of all peptides is then utilized to compute a position specific scoring matrix. This matrix assigns different weights to each position and amino acid type, and provides a score for each peptide. Finally, the score is compared with a threshold to determine if the peptide binds or not to MHC II. The predictive power of the scoring matrix was demonstrated on several benchmark datasets, showing that the novel algorithm may be useful to design peptides that bind to MHC II and that may be used in vaccine development.

Major advances in proteomics are a result of significant technological advances in protein purification and mass spectrometry. Another critical component is the automated and reliable protein identification from mass spectrometric data. To improve the protein identification process, Hernandez et al. devised a heuristic algorithm that addresses the difficulties of the current methods, such as poor performance for large databases or for low quality data [74]. The new method based on ACO matches theoretical peptide sequences from a database with a structured representation of the source MS/MS spectrum. Tested with a set of 721 MS/MS spectra, the ACO-based procedure showed success rate of 88.9%, demonstrating that the artificial ants may perform an efficient exploration of the search space.

Particle Swarm Optimization

Particle swarm algorithms are used in diverse biochemistry and drug design applications, to solve problems that require binary or real value optimization. Among the advantages of using PSO in optimization one should count

the simple algorithm that translates into small and effective software, fast convergence, small population, and low number of iterations. PSO is applied with success to difficult problems, such as feature selection for gene expression data [14,75], identification of the global minimum geometry of chemical compounds [76], enzyme-inhibitor docking [15], QSAR [16], and protein motif discovery [77].

PSO is an effective replacement of GA for the global optimization of protein-ligand geometry in docking studies. Several PSO modifications of the most popular docking program, AutoDock, were proposed in the literature. The Tribe-PSO algorithm was used in AutoDock to identify the best protein-ligand geometry [78]. In Tribe-PSO the population is divided into several subpopulations or tribes. Each tribe has the same structure and evolution mechanism as the basic PSO model. In the first phase, each tribe evolves independent of the other tribes and converges to an optimum solution. In the second phase the tribes exchange information regarding the best solution from each tribe, and in the third phase all particles are united into a single population that evolves as a classical PSO model towards the final solution. In a comparative test involving 100 protein-ligand complexes from PDB, over 90% are docked better with Tribe-PSO than with AutoDock. Another PSO modification of AutoDock is SODOCK, which combines the basic PSO model with a local search for the best particle [79]. Compared with four docking methods (GOLD, DOCK, FlexX, and AutoDock) for a set of 37 PDB protein-ligand complexes, SODOCK obtained an average RMSD of 2.29 Å, whereas all other docking programs had an RMSD higher than 3 Å. In a related implementation, PSO@AUTODOCK, AutoDock is combined with a PSO variant that allows larger movements in the search space [15]. Significant improvement is obtained for 12 out of the 37 test complexes, compared with the SODOCK predictions.

Feature selection is an important step in QSAR and in virtual screening of chemical libraries, because almost all QSAR models are sensitive to the presence of irrelevant descriptors. Another benefit of feature selection is the identification of structural descriptors that may explain the mechanism of a particular structure-activity relationship. Agrafiotis and Cedeño used a binary PSO to select descriptors for a QSAR based on multilayer feed-forward artificial neural networks (MLF ANN) [16]. The real value PSO model may also be used for feature selection, as shown for QSAR models based on k-nearest neighbors kernel regression [80]. The target of the PSO model was to find the optimum weight (situated in the range $[0, 1]$) for each structural descriptor. The features wit the largest weights were selected in the QSAR model.

In a comparative study for 42 cyclooxygenase inhibitors, Lü et al. found that binary PSO is superior to GA for feature selection in multi-linear regression (MLR) QSAR [81]. Shen et al. showed that the partial least-squares (PLS) QSAR model could be improved by using structural descriptors selected with a binary PSO [82]. Another approach to feature selection is the optimized block-wise variable combination (OBVC) method that combines a descriptor selection guided by PSO followed by PLS modeling of the data [83,84]. Instead of selecting each descriptor independent of the other descriptors, OBVC operates with groups of descriptors. The size and composition of each group of descriptors is optimized with PSO. OBVC was evaluated in QSAR models for the carcinogenic potency of aromatic amines [83] and for inhibitors of lung carcinoma cells [84]. OBVC was also tested for a QSAR dataset consisting of 37 ligands of the $\alpha 6$ benzodiazepine receptor, and more than 70 structural descriptors (topological, geometric, and quantum indices) [85]. Comparative tests show that OBVC exceeds the predictions obtained with MLR, PLS, and hierarchical PLS. OBVC may suggest several combinations of descriptors with comparable prediction statistics, and can assist the discovery of the most important structural descriptors.

PSO is used also to modify and improve QSAR models, such as the piecewise modeling by particle swarm algorithm (PMPSO) which is a QSAR based on piecewise linear models [86]. PMPSO may be useful for dataset with high structural diversity, when a single linear model for all compounds might not be the best option. A minimum spanning tree model is used to cluster all compounds, and then PSO is applied to divide the tree in predictive piecewise linear models. PMPSO was applied with good results for angiotensin II antagonists. A variant of this QSAR model is the piecewise hypersphere modeling by particle swarm optimization (PHMPSO) which clusters similar compounds in subsets defined as hyperspheres [87]. The position and size of the hyperspheres are optimized with PSO, and then a QSAR model is fitted for the compounds in each hypersphere. PHMPSO was tested with good results for dihydrofolate reductase inhibitors, epidermal growth factor receptor inhibitors, and benzodiazepine receptor ligands [88]. Another PSO-modified algorithm is the optimized sample-weighted PLS (OSWPLS) which uses PSO to weight each object (chemical compound in QSAR) from the training dataset [89]. The weight determines the importance of each object, and the target of the PSO step is to minimize the error of the calibration model.

Training a neural network QSAR consists of (a) finding the best network topology (number of hidden neurons and distribution of the connections between neu-

rons), and (b) optimization of the connection weights. PSO is very efficient in optimizing the ANN weights, as shown in a QSAR study of inhibitors of platelet-derived growth factor receptor phosphorylation [18]. The versatility of swarm algorithm is practical in a global optimization of ANN QSAR, namely finding the best topology and set of weights. Shen proposed a hybrid use of PSO in training a MLF ANN, namely a binary PSO to determine the optimum network topology and a continuous PSO to find the optimum connection weights [90]. Extensive tests showed that this combination converges quickly and may avoid the overfitting of the learning dataset of chemicals.

The training process of a radial basis function artificial neural network (RBF ANN) consists of selecting the network topology, finding the centers and widths of the RBF neurons, and computing the connection weights between the hidden and output layers. A hybrid particle swarm optimization (HPSO) was used by Zhou et al. to train a RBF network for drug design studies [91]. In the HPSO algorithm, a discrete PSO is used to optimize the network topology, whereas a continuous PSO is used to optimize the network parameters. The new QSAR approach was tested with a dataset of 40 inhibitors of murine P388 leukemia cells and over 70 Cerius^2 descriptors. The HPSO network has the highest predictions: PLS, $r = 0.664$; RBF with parameters optimized with PSO, $r = 0.838$; RBF with parameters optimized with K-means, $r = 0.852$; RBF optimized with HPSO, $r = 0.894$. A similar trend was found for a second QSAR test, performed with 72 cyclooxygenase-2 inhibitors: RBF optimized with PSO, $r = 0.894$; RBF optimized with K-means, $r = 0.903$; RBF optimized with HPSO, $r = 0.921$. The experimental evidence suggests that the hybrid PSO optimization of RBF-ANN has a fast convergence to predictive QSAR models.

Zhou et al. proposed a novel version of nonlinear partial least-square method that is based on structural descriptors transformed by an artificial neural network [92]. The structural descriptors represent the ANN input, whereas the output signals from the neurons in the hidden layer represent the non-linear input for PLS. The ANN weights are trained with PSO. The novel non-linear QSAR model was tested with good results for two datasets, namely 53 antitumor agents, and 52 benzodiazepine receptor ligands.

As shown in the QSAR studies reviewed here, PSO is an efficient method to optimize linear and non-linear structure-activity models. A fast convergence to the global minimum depends on the parameters that control the population size, number of iterations, and weights to update the velocity of each particle. Choosing the best parameters that control a PSO model is a meta-optimization problem that was solved by Meissner et al. with the optimized particle swarm optimization (OPSO) model, in which the control parameters are optimized by a meta-swarm [93]. Although OPSO is more complex than a classical PSO because it contains swarms within a swarm, the system converges fast to good QSAR models. OPSO was tested for the prediction of the blood-brain barrier permeation coefficient with a MLF neural network.

Support vector machines (SVM) represent a class of versatile models that can produce non-linear classification or regression QSAR equations [94]. PSO can be efficiently applied to select the best structural descriptors for SVM models, as demonstrated in a QSAR for P-glycoprotein substrates [17]. The mathematical formalism of SVM was adapted by Lin et al. for the training of MLF ANN [95]. The parameters of the hybrid method SVM-ANN were optimized with PSO, and the new QSAR model was compared with other two algorithms, namely back-propagation ANN (BP-ANN) and ANN optimized with PSO (PSO-ANN). These methods were compared for a dataset of 111 dihydrofolate reductase inhibitors and for another set of 85 cyclooxygenase-2 inhibitors. The results show that SVM-ANN models have better prediction statistics, and that the PSO procedure converges fast to optimum parameters. A similar QSAR model was developed based on radial basis function ANN [96], by defining a nonlinear SVM model (RBF-SVM) representing a kernel transform based on RBF ANN optimized with PSO. QSAR models obtained for inhibitors of HIV-1 reverse transcriptase demonstrate that RBF-SVM provides better predictions compared to BP-ANN and SVM.

Artificial Immune Systems

The mechanisms and functions of the biological immune system were used as an inspiration for many AIS algorithms, such as the artificial immune network (aiNet) [97,98], the hierarchical artificial immune network (HaiNet) [37], the artificial immune recognition system (AIRS) [99,100,101,102], the clonal selection algorithm (CLONALG) [103,104], the clonal selection classification system (CSCA) [105], IMMUNOS-81 [106], and IMMUNOS-99 [107]. The pattern recognition capabilities of the artificial immune systems may be applied in modeling structure-activity relationships for drug design or for the computational screening of chemical libraries. In the following sections we review several SAR models obtained with AIRS, CLONALG, CSCA, and IMMUNOS. All AIS models were computed with Weka [108].

AIRS – Artificial Immune Recognition System

The AIRS machine learning algorithm developed by Watkins, Timmis, and Boggess is an efficient and popular pattern recognition adaptation of AIS [99,100,101,102]. Brownlee tested AIRS for a wide range of classification problems [109], confirming its utility as a supervised learning classifier. The main characteristics of AIRS are briefly reviewed below.

An antigen is represented as an n-dimensional vector $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, where each structural descriptor x_i is a real number ($x_i \in R$ for $i = 1, 2, \ldots, n$), and an associated class $y = \{+1, -1\}$. An identical encoding is used for antibodies. An artificial recognition ball (ARB) represents a B-cell, and consists of an antibody, a number of resources, and a stimulation value. The similarity between an ARB and an antigen is measured by the stimulation value. The number of resource from an AIRS model is limited, and ARBs compete for their allocation. Resources are allocated to the most stimulated ARBs by removing them from the least stimulated ARBs, and ARBs without resources are eliminated from the cell population. The ARB population is trained during several cycles of competition for limited resources. In each cycle of ARB training, the best ARB classifiers generate mutated clones that enhance the antigen recognition process, whereas the ARBs with insufficient resources are removed from the population. After training, the top ARB classifiers are selected as memory cells. Finally, the memory cells are used to classify novel antigens (patterns).

The drug design applications reviewed here were obtained with AIRS2, an improved version of AIRS [110]. The AIRS2 algorithm consists of the following steps [109]:

(1) Initialization. In the first phase of the algorithm the system is prepared for the learning process. The training data are normalized between 0 and 1. The Euclidean distance is computed for all pairs of antigens, and then the affinity Af is determined as the ratio between the distance and the maximum distance. The affinity threshold AT is computed as the average affinity for all antigens in the training set. The memory cell pool is populated with randomly selected antigens. At the end of the AIRS algorithm, the memory cell pool represents the recognition ARBs used as classifiers.

(2) Train for all antigens. The AIRS algorithm trains a classifier by passing only once over the entire population of training antigens.

(2.1) Antigen presentation. Each training antigen is presented to the memory cell pool, and each memory cell receives a stimulation value St, $\mathrm{St} = 1 - Af$. The memory cells with the largest stimulation values are selected, and a number of mutated clones are created and added to the ARB pool. The number of clones NC generated is computed with the formula:

$$\mathrm{NC} = \mathrm{St} \times \mathrm{CR} \times \mathrm{HR}$$

where the clonal rate CR and the hypermutation rate HR are user defined parameters.

(2.2) Competition for limited resources. During this iterative process the algorithm selects those ARBs that have the best recognition capabilities, while optimally allocating the resources to the best ARBs. For each antigen the process trains only those ARBs from the same class with the antigen.

(2.2.1) Perform competition for resources. The total number of resources is a user defined parameter that limits the number of ARBs.

(2.2.1.1) Stimulation. The selected antigen is presented to all ARBs and the stimulation is computed for each cell in the ARB pool.

(2.2.1.2) Normalization. The ARB stimulation values NSt are normalized.

(2.2.1.3) Allocate limited resources. The amount of resources Rs allocated to each ARB is computed from the normalized stimulation NSt and the clonal rate CR:

$$\mathrm{Rs} = \mathrm{NSt} \times \mathrm{CR}$$

The ARB pool is sorted in the descending order of allocated resources Rs and then resources are removed from the ARB situated at the end of the list until the sum of all allocated resources is lower than the total number of resources.

(2.2.1.4) Remove ARBs with insufficient resources. The ARBs with zero resources are removed from the pool.

(2.2.2) Continue with (2.3) if the stop condition is satisfied. The stop condition for the ARB refinement is met when the average normalized stimulation is higher than a user defined stimulation threshold.

(2.2.3) Generate mutated clones of surviving ARBs. The number of clones generated for each ARB is:

$$\mathrm{NC} = \mathrm{St} \times \mathrm{CR}$$

where St is the stimulation against the antigen, and CR is the clonal rate. The clones undergo a process of hypermutation, during which the elements of the **x** vector are randomly modified to increase the antigen recognition.

(2.2.4) Go to (2.2.1)

(2.3) Memory cell selection. In this step, new ARB classifiers are evaluated for inclusion in the memory cell

pool. An ARB is inserted into the memory cell pool if its stimulation value is higher than that of the existing best matching memory cell. The existing best matching memory cell is then removed if the affinity between the candidate ARB and the existing memory cell is less than a cut-off value CutOff computed with the formula:

$$\text{CutOff} = \text{AT} \times \text{ATS}$$

where the affinity threshold AT was computed during the initialization phase, and the affinity threshold scalar ATS is a user defined parameter.

(3) Classification. At the end of the training phase, the memory cell pool represents the AIRS classifier. The classification is performed with a *k*-nearest neighbor method, in which the *k* best matches to a prediction pattern are identified and the predicted class is determined with a majority vote. The parameter *k* is user defined, and may be optimized to maximize the prediction performances.

AIRS was applied with success in several drug design structure-activity relationships that are reviewed here. The classification performance of the AIRS algorithm depends on eight user defined parameters: affinity threshold scalar, clonal rate, hypermutation rate, number of nearest neighbors, initial memory cell pool size, number of instances to compute the affinity threshold, stimulation threshold, and total resources. To illustrate the influence of these parameters, we show the variation of the prediction statistics with the affinity threshold scalar. The statistical indices reported for each AIRS model are: TP_p, true positive in prediction (number of compounds from class +1 classified in class +1); with: FN_p, false negative in prediction (number of compounds from class +1 classified in class −1); with: TN_p, true negative in prediction (number of compounds from class −1 classified in class −1); with: FP_p, false positive in prediction (number of compounds from class −1 classified in class +1); with: Se_p, prediction selectivity; with: Sp_p, prediction specificity; with: Ac_p, prediction accuracy; with: MCC_p, prediction Matthews correlation coefficient.

Torsade de pointes (TdP) is a polymorphic ventricular arrhythmia that may be caused by drugs that induce the prolongation of the QT interval [111]. QT prolongation and TdP may be caused by a large number of drugs, such as antiarrhythmics CE4, antihistamines, antimicrobials, antidepressants, and antipsychotics. The drug design and development costs may be significantly reduced if, along with other ADME/Tox filters, chemical compounds that have the potential to induce torsade de pointes are eliminated as early as possible. AIRS was applied with success to classify 349 drugs into a subset of 106 drugs that induce torsade de pointes and a subset of 243 drugs that do not induce torsade de pointes [112]. The chemical structure was described with five linear solvation energy relationships (LSER) descriptors, and the prediction of the AIRS models was evaluated with the ten fold (leave-10%-out) cross-validation. The with: MCC_p variation with ATS (Table 1) shows that the best predictions are obtained with low values of ATS, in this case ATS = 0.05 (Ac = 0.845, MCC = 0.632). After several steps of optimizations involving the remaining seven parameters, the best AIRS model (Ac = 0.860, MCC = 0.671) has better predictions than 11 other machine learning algorithms.

Drug Design, Artificial Intelligence Methods in, Table 1
AIRS prediction statistics for TdP SAR models based on LSER descriptors and computed for various values of the affinity threshold scalar ATS

ATS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
0.01	76	30	213	30	0.7170	0.8765	0.8281	0.5935
0.05	78	28	217	26	0.7358	0.8930	0.8453	0.6323
0.10	78	28	206	37	0.7358	0.8477	0.8138	0.5710
0.30	71	35	210	33	0.6698	0.8642	0.8052	0.5369
0.50	63	43	203	40	0.5943	0.8354	0.7622	0.4333
0.70	55	51	198	45	0.5189	0.8148	0.7249	0.3394
0.90	49	57	198	45	0.4623	0.8148	0.7077	0.2872

In a related study, AIRS was applied to the classification of 361 drugs (85 induce torsade de pointes, and 276 do not induce torsade de pointes) based on 159 structural indices computed from the molecular structure [113]. The ATS parameter has a significant influence on the AIRS predictions (Table 2a). A series of fivefold (leave-20%-out) cross-validation tests shows that MCC increases from 0.2173 for ATS = 0.01, peaks at 0.2795 for ATS = 0.09, and then decreases to 0.1604 for ATS = 0.95. To investigate the effect of feature selection on the AIRS prediction quality, Weka was used to reduce the number of features to 13 with the combination SubsetEvaluation and BestFirst. Feature selection significantly improves the TdP predictions (Table 2b), with the best predictions obtained for ATS = 0.15 (MCC = 0.356). These results suggest that feature selection should be explored in order to increase the AIRS prediction power.

A good intestinal absorption is a major requirement for oral drugs [114,115], and various computational models were proposed as fast, reliable, and inexpensive in silico methods to assess the intestinal permeability of a chemical compound before synthesis [116,117]. The oral absorption of a drug is influence by a large number of variables,

CE4 Please confirm change.

Editor's or typesetter's annotations (will be removed before the final TEX run)

Drug Design, Artificial Intelligence Methods in, Table 2
AIRS prediction statistics for TdP SAR models based on 2D/3D descriptors and computed for various values of the affinity threshold scalar ATS

ATS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 159 structural descriptors								
0.01	39	46	213	63	0.4588	0.7717	0.6981	0.2173
0.05	42	43	213	63	0.4941	0.7717	0.7064	0.2484
0.09	43	42	218	58	0.5059	0.7899	0.7230	0.2795
0.15	40	45	204	72	0.4706	0.7391	0.6759	0.1924
0.30	40	45	207	69	0.4706	0.7500	0.6842	0.2039
0.50	36	49	203	73	0.4235	0.7355	0.6620	0.1470
0.70	36	49	201	75	0.4235	0.7283	0.6565	0.1396
0.95	38	47	201	75	0.4471	0.7283	0.6620	0.1604
(b) 13 structural descriptors								
0.01	40	45	236	40	0.4706	0.8551	0.7645	0.3327
0.05	42	43	235	41	0.4941	0.8514	0.7673	0.3484
0.09	40	45	227	49	0.4706	0.8225	0.7396	0.2885
0.15	47	38	226	50	0.5529	0.8188	0.7562	0.3558
0.30	30	55	238	38	0.3529	0.8623	0.7424	0.2336
0.50	24	61	243	33	0.2824	0.8804	0.7396	0.1894
0.70	29	56	246	30	0.3412	0.8913	0.7618	0.2668
0.95	30	55	235	41	0.3529	0.8514	0.7341	0.2182

Drug Design, Artificial Intelligence Methods in, Table 3
AIRS prediction statistics for HIA SAR models computed for various values of the affinity threshold scalar ATS

ATS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 159 structural descriptors								
0.01	105	26	33	32	0.8015	0.5077	0.7041	0.3174
0.04	107	24	33	32	0.8168	0.5077	0.7143	0.3364
0.09	107	24	34	31	0.8168	0.5231	0.7194	0.3506
0.15	107	24	28	37	0.8168	0.4308	0.6888	0.2640
0.30	100	31	30	35	0.7634	0.4615	0.6633	0.2287
0.50	105	26	25	40	0.8015	0.3846	0.6633	0.1997
0.70	105	26	25	40	0.8015	0.3846	0.6633	0.1997
0.95	105	26	25	40	0.8015	0.3846	0.6633	0.1997
(b) 21 structural descriptors								
0.01	113	18	41	24	0.8626	0.6308	0.7857	0.5064
0.04	114	17	42	23	0.8702	0.6462	0.7959	0.5300
0.09	113	18	39	26	0.8626	0.6000	0.7755	0.4796
0.15	111	20	40	25	0.8473	0.6154	0.7704	0.4727
0.30	116	15	30	35	0.8855	0.4615	0.7449	0.3885
0.50	121	10	30	35	0.9237	0.4615	0.7704	0.4500
0.70	120	11	28	37	0.9160	0.4308	0.7551	0.4090
0.95	123	8	28	37	0.9389	0.4308	0.7704	0.4495

such as drug formulation and stability, aqueous solubility, contents of the gastrointestinal tract, residence time in the intestine, intestinal metabolism, rate of passive intestinal permeability, carrier-mediated influx, and active efflux via transporters. The human intestinal absorption(HIA) of 196 drugs (131 drugs that penetrate the human intestine, and 65 drugs that do not penetrate the intestine) was modeled with the AIRS algorithm [118]. The AIRS classifiers were obtained with 159 structural descriptors from five classes, namely constitutional, topological indices, electrotopological state indices, quantum descriptors, and geometrical indices. The influence of the ATS parameter in L20%O cross-validation was investigated for values between 0.01 and 0.95 (Table 3a). As in previous experiments, MCC increases from 0.3174 for ATS = 0.01 to a maximum of 0.3506 for ATS = 0.09, and then decreases to 0.1997 for ATS = 0.95. After optimizing all eight parameters, the best predictions of the AIRS algorithm (Ac = 0.735, MCC = 0.406) are higher than those obtained with seven other machine learning algorithms, namely Bayesian network, naïve Bayes classifier, updateable naïve Bayes classifier, logistic regression, Gaussian radial basis function network, decision tree with naïve Bayes classifiers at the leaves, and random tree. In a feature selection experiment (SubsetEvaluation and BestFirst) the number of structural descriptors was reduced to 21, which improved considerably the AIRS predictions [119]. The results obtained for the ATS parameter (Table 3b) show a significant increase across the entire range of values, with a maximum of 0.53 for ATS = 0.04.

P-glycoprotein (Pgp) is responsible for the low cellular accumulation of anticancer drugs, for reduced oral absorption, for low blood-brain barrier penetration, and in hepatic, renal, or intestinal elimination of drugs. Computational methods for the identification of Pgp substrates are useful drug design tools for the early elimination of potential Pgp substrates [120,121]. The immune system classifier AIRS was used to discriminate between 116 Pgp substrates and 85 Pgp nonsubstrates [122]. The SAR models were computed from 159 structural descriptors and the prediction power was estimated with L20%O cross-validation. Low values for the ATS parameter give better predictions, with the highest predictions obtained for ATS = 0.03 (Table 4a). The AIRS model optimized for all eight parameters (Ac = 0.702, MCC = 0.380) is better than five machine learning algorithms (alternating decision tree, Bayesian network, logistic regression with ridge estimator, random tree, and fast decision tree learner), demonstrating that Pgp substrates may be successfully recognized with AIRS. A feature selection step reduces the number of structural descriptors from 159 to 15, and increases the SAR performances over the entire range of ATS values (Table 4b) [119]. The best predictions are obtained

Drug Design, Artificial Intelligence Methods in, Table 4
AIRS prediction statistics for Pgp SAR models computed for various values of the affinity threshold scalar ATS

ATS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 159 structural descriptors								
0.01	85	31	45	40	0.7328	0.5294	0.6468	0.2671
0.03	85	31	48	37	0.7328	0.5647	0.6617	0.3009
0.07	80	36	46	39	0.6897	0.5412	0.6269	0.2320
0.15	78	38	51	34	0.6724	0.6000	0.6418	0.2709
0.30	80	36	42	43	0.6897	0.4941	0.6070	0.1863
0.50	80	36	44	41	0.6897	0.5176	0.6169	0.2092
0.70	80	36	43	42	0.6897	0.5059	0.6119	0.1978
0.95	80	36	43	42	0.6897	0.5059	0.6119	0.1978
(b) 15 structural descriptors								
0.01	86	30	62	23	0.7414	0.7294	0.7363	0.4668
0.03	85	31	59	26	0.7328	0.6941	0.7164	0.4241
0.07	85	31	54	31	0.7328	0.6353	0.6915	0.3681
0.15	88	28	58	27	0.7586	0.6824	0.7264	0.4403
0.30	87	29	57	28	0.7500	0.6706	0.7164	0.4199
0.50	81	35	56	29	0.6983	0.6588	0.6816	0.3544
0.70	78	38	58	27	0.6724	0.6824	0.6766	0.3509
0.95	80	36	56	29	0.6897	0.6588	0.6766	0.3455

Drug Design, Artificial Intelligence Methods in, Table 5
AIRS prediction statistics for BZR SAR models computed for various values of the affinity threshold scalar ATS

ATS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 75 structural descriptors								
0.01	63	19	52	29	0.7683	0.6420	0.7055	0.4137
0.05	64	18	53	28	0.7805	0.6543	0.7178	0.4385
0.10	62	20	52	29	0.7561	0.6420	0.6994	0.4008
0.20	60	22	57	24	0.7317	0.7037	0.7178	0.4356
0.25	65	17	56	25	0.7927	0.6914	0.7423	0.4867
0.30	62	20	56	25	0.7561	0.6914	0.7239	0.4485
0.50	63	19	56	25	0.7683	0.6914	0.7301	0.4611
0.95	63	19	56	25	0.7683	0.6914	0.7301	0.4611
(b) 16 structural descriptors								
0.01	64	18	57	24	0.7805	0.7037	0.7423	0.4857
0.05	65	17	57	24	0.7927	0.7037	0.7485	0.4985
0.10	57	25	55	26	0.6951	0.6790	0.6871	0.3742
0.20	62	20	65	16	0.7561	0.8025	0.7791	0.5591
0.25	60	22	62	19	0.7317	0.7654	0.7485	0.4974
0.30	61	21	61	20	0.7439	0.7531	0.7485	0.4970
0.50	59	23	62	19	0.7195	0.7654	0.7423	0.4854
0.95	59	23	61	20	0.7195	0.7531	0.7362	0.4728

with ATS = 0.01 (Ac = 0.736 and MCC = 0.467), but the variation of the prediction statistics is not monotonous with ATS, and no simple rule can be extracted to guide further experiments.

Another successful application of AIRS in drug design was reported for the identification of benzodiazepine receptor (BZR) ligands [123]. The structure of the 163 BZR ligands was encoded with 75 structural descriptors, and AIRS classifiers were trained to discriminate between 82 high affinity ligands (class +1, pIC_{50} between 8.92 and 7.80) and 81 low affinity ligands (class −1, pIC_{50} between 7.77 and 5). A scan of the ATS values (Table 5a) shows that the best predictions are obtained for ATS = 0.025 (Ac = 0.7423 and MCC = 0.4867). The feature selection step further reduces the number of structural descriptors to 16 (Table 5b), and results in better predictions (best ATS = 0.20, with Ac = 0.7791 and MCC = 0.5591).

Numerous organic chemicals are environmental pollutants, and a considerable number of studies are dedicated to the computational prediction of their mechanism of aquatic toxicity (MOA). The reliable prediction of MOA has major applications in selecting the appropriate QSAR model, to identify chemicals with similar toxicity mechanism, and in extrapolating toxic effects between different species and exposure regimens [124,125]. The immune system AIRS was applied for the MOA prediction of 187 chemicals (143 non-polar narcotics, and 44 polar narcotics) [126]. The chemical structure was described with five LSER descriptors, and the AIRS predictions were evaluated with the ten fold cross-validation. The ATS parameter was modified between 0.01 and 0.95 (Table 6), and the best predictions were obtained for low ATS values (0.01, 0.02, and 0.05), namely Ac = 0.9519 and MCC = 0.8674. Based on the high prediction rates obtained with AIRS, such models may be used to identify the aquatic toxicity mechanism and to select the appropriate computational model for new chemical compounds.

Drug Design, Artificial Intelligence Methods in, Table 6
AIRS prediction statistics for MOA SAR models computed for various values of the affinity threshold scalar ATS

ATS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
0.01	138	5	40	4	0.9650	0.9091	0.9519	0.8674
0.02	138	5	40	4	0.9650	0.9091	0.9519	0.8674
0.05	138	5	40	4	0.9650	0.9091	0.9519	0.8674
0.15	135	8	39	5	0.9441	0.8864	0.9305	0.8120
0.30	139	4	39	5	0.9720	0.8864	0.9519	0.8653
0.50	138	5	39	5	0.9650	0.8864	0.9465	0.8514
0.70	137	6	39	5	0.9580	0.8864	0.9412	0.8379
0.95	137	6	39	5	0.9580	0.8864	0.9412	0.8379

CLONALG – Clonal Selection Algorithm

An AIS algorithm that gives a central role to the clonal selection theory is CLONALG, proposed by de Castro and Von Zuben [103,104]. CLONALG implements several mechanisms of the clonal selection: training of a group of memory cells; identification and cloning of the antibodies with the highest recognition power; death of the antibodies with low recognition power; cloning and hypermutation of the antibodies with high recognition power; evaluation and replacement of the clones; generation and preservation of antibody diversity. The CLONALG algorithm, as implemented by Brownlee, consists of the following steps [105]:

(1) Initialization. The CLONALG algorithm starts by generating a pool of N antibodies, which is subsequently partitioned into the memory antibody pool (MAP) and the remaining antibody pool (RAP). MAP contains m antibodies, and at the end of the training process it will represent the solution of the CLONALG classifier. RAP contains the remaining antibodies, $r = N - m$, and it has the role of adding additional diversity during the learning phase.

(2) Train antibodies. The main part of the CLONALG algorithm is an iterative process of exposing the system to all antigens from the training set for a number of G generations (iterations).

(2.1) Train for each antigen. Repeat steps (2.2)–(2.9) for all antigens in the training set. In each generation, an antigen is selected for training once and only once.

(2.2) Antigen selection. For each generation, an antigen is randomly selected without replacement from the entire pool of antigens.

(2.3) Affinity calculation. The selected antigen interacts with all antibodies, and the affinity is calculated for the interaction between the antigen and every antibody in the system. The affinity measures the similarity between an antigen and an antibody, and is based on the Euclidean distance between the vectors of structural descriptors that characterize the antigen and the antibody.

(2.4) Select antibodies. The antibodies are ranked according to their decreasing affinity towards the antigen, and the top n antibodies are selected for further processing.

(2.5) Clone antibodies. All n antibodies selected in the previous step are cloned proportionally with their affinity. The number of clones computed for an antibody that is ranked ith according to its affinity, with $i \in [1, n]$, is

$$\mathrm{Nc} = \left\lfloor \frac{\mathrm{CF} \times N}{i} + 0.5 \right\rfloor$$

where CF is the clonal factor. The total number of clones generated for the entire system of n antibodies is:

$$\mathrm{NC} = \sum_{i=1}^{n} \mathrm{Nc} \,.$$

(2.6) Affinity maturation. The clones enter the process of affinity maturation, during which random mutations are performed onto each clone in order to increase its affinity towards the antigen. The degree of affinity maturation is inversely proportional to the initial affinity, namely the lower the initial affinity the greater the mutation rate is.

(2.7) Evaluate clones. All clones are exposed to the antigen to compute their affinity.

(2.8) Select candidates. The antibodies with the highest affinity are selected to replace antibodies from MAP that have lower affinities.

(2.9) Replacement. The RAP group of antibodies is ranked according to the decreasing affinity towards the antigen, and the set of s antibodies with the lowest affinity is replaces with random antibodies.

(3) Classification. After training the system for G generations, the MAP group of antigens represents the solution of the CLONALG classifier.

The CLONALG machine learning was tested with success in drug design applications, namely recognition of glycogen phosphorylase B inhibitors, classification of benzodiazepine receptor ligands, and identification of polar and nonpolar narcotic pollutants. To illustrate the effect of the user defined parameters on the prediction performance of CLONALG, we show the influence of the clonal factor CF on the L20%O cross-validation statistics. The clonal factor is a scaling factor, with values between 0 and 1, that determines the number of clones generated for each selected antibody. Low values for CF result in a local search, whereas for high values the algorithm generates a larger number of clones that may explore a wider region and result in a higher diversity.

CLONALG in drug development for the recognition of glycogen phosphorylase B (GPB) inhibitors, based on a set of 66 compounds and 70 structural descriptors [127]. The subset of active compounds contains 33 chemicals (class +1, pKi between 6.8 and 2.5), whereas the subset of inactive compounds contains the remaining 33 chemicals

Drug Design, Artificial Intelligence Methods in, Table 7
CLONALG prediction statistics for GPB SAR models computed for various values of the clonal factor CF

CF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 70 structural descriptors								
0.01	20	13	15	18	0.6061	0.4545	0.5303	0.0613
0.05	22	11	17	16	0.6667	0.5152	0.5909	0.1839
0.08	22	11	14	19	0.6667	0.4242	0.5455	0.0937
0.15	20	13	17	16	0.6061	0.5152	0.5606	0.1217
0.25	23	10	15	18	0.6970	0.4545	0.5758	0.1562
0.50	23	10	18	15	0.6970	0.5455	0.6212	0.2453
0.65	24	9	16	17	0.7273	0.4848	0.6061	0.2186
0.95	22	11	14	19	0.6667	0.4242	0.5455	0.0937
(b) 2 structural descriptors								
0.01	21	12	22	11	0.6364	0.6667	0.6515	0.3032
0.05	21	12	23	10	0.6364	0.6970	0.6667	0.3339
0.08	21	12	23	10	0.6364	0.6970	0.6667	0.3339
0.15	21	12	23	10	0.6364	0.6970	0.6667	0.3339
0.25	21	12	23	10	0.6364	0.6970	0.6667	0.3339
0.50	21	12	22	11	0.6364	0.6667	0.6515	0.3032
0.65	21	12	23	10	0.6364	0.6970	0.6667	0.3339
0.95	21	12	22	11	0.6364	0.6667	0.6515	0.3032

Drug Design, Artificial Intelligence Methods in, Table 8
CLONALG prediction statistics for BZR SAR models computed for various values of the clonal factor CF

CF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 75 structural descriptors								
0.01	56	26	48	33	0.6829	0.5926	0.6380	0.2767
0.05	57	25	50	31	0.6951	0.6173	0.6564	0.3134
0.10	58	24	51	30	0.7073	0.6296	0.6687	0.3380
0.20	56	26	52	29	0.6829	0.6420	0.6626	0.3252
0.45	56	26	53	28	0.6829	0.6543	0.6687	0.3374
0.60	62	20	52	29	0.7561	0.6420	0.6994	0.4008
0.85	57	25	47	34	0.6951	0.5802	0.6380	0.2773
0.95	63	19	47	34	0.7683	0.5802	0.6748	0.3550
(b) 16 structural descriptors								
0.01	42	40	53	28	0.5122	0.6543	0.5828	0.1682
0.05	57	25	53	28	0.6951	0.6543	0.6748	0.3498
0.10	53	29	55	26	0.6463	0.6790	0.6626	0.3255
0.20	57	25	54	27	0.6951	0.6667	0.6810	0.3620
0.45	64	18	52	29	0.7805	0.6420	0.7117	0.4267
0.60	61	21	52	29	0.7439	0.6420	0.6933	0.3880
0.85	51	31	55	26	0.6220	0.6790	0.6503	0.3014
0.95	57	25	55	26	0.6951	0.6790	0.6871	0.3742

(class −1, pK_i between 2.4 and 1.3). The prediction performance depends on the number of clones generated, controlled by the values of CF (Table 7a), with best results for CF = 0.50 (Ac = 0.6212 and MCC = 0.2453), whereas low and high values for CF result in lower predictions. A feature selection step drastically reduces the number of structural descriptors from 70 to 2, while the model prediction increases (Ac = 0.6667 and MCC = 0.3339) for several CF values (Table 7b).

The CLONALG immune system was tested for the classification of 163 benzodiazepine receptor (BZR) ligands (82 high affinity ligands and 81 low affinity ligands) which are encoded with 75 structural descriptors [123]. The clonal factor was modified between 0.01 and 0.95 (Table 8a). The prediction MCC increases from 0.2767 for CF = 0.01, peaks at 0.4008 for CF = 0.60, and then decreases to 0.2888 for CF = 0.90. These results indicate that too few or too many clones are detrimental to the antigen recognition. The number of structural descriptors can be significantly reduced to 16 by feature selection (Table 8b), which also results in a slight increase of the prediction quality (MCC = 0.4267 for CF = 0.45). The optimum CF is situated in the middle of the range of CF values, similarly with the results obtained for the identification of GPB inhibitors.

The mechanism of toxic action of polar and nonpolar narcotic pollutants may be efficiently identified with

Drug Design, Artificial Intelligence Methods in, Table 9
CLONALG prediction statistics for MOA SAR models computed for various values of the clonal factor CF

CF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
0.01	98	16	73	3	0.8596	0.9605	0.9000	0.8052
0.05	100	14	72	4	0.8772	0.9474	0.9053	0.8116
0.10	103	11	75	1	0.9035	0.9868	0.9368	0.8763
0.15	105	9	68	8	0.9211	0.8947	0.9105	0.8141
0.30	104	10	72	4	0.9123	0.9474	0.9263	0.8503
0.55	110	4	69	7	0.9649	0.9079	0.9421	0.8791
0.70	102	12	73	3	0.8947	0.9605	0.9211	0.8427
0.90	105	9	73	3	0.9211	0.9605	0.9368	0.8720

CLONALG classifiers [128]. The dataset consists of 190 compounds (114 nonpolar pollutants, class +1; 76 polar pollutants, class −1), with each chemical characterized by five structural descriptors, namely the octanol-water partition coefficient, the energy of the highest occupied molecular orbital, the energy of the lowest unoccupied molecular orbital, the most negative partial charge on any non-hydrogen atom in the molecule, and the most positive partial charge on a hydrogen atom. The prediction MCC has no clear-cut variation with CF (Table 9), but the optimum is still in the middle of the range, as in previous studies, with MCC = 0.8791 for CF = 0.55.

CSCA – Clonal Selection Classification System

The clonal selection classification system, developed by Brownlee, is formulated as a function optimization procedure that maximizes the number of patterns correctly classified and minimizes the number of patterns incorrectly classified [105]. Unlike the AIRS algorithm, in which the system is exposed only once to the set of antigens, CSCA is trained for several generations, and during a generation the entire set of antibodies is exposed to all antigens. The computational steps of the CSCA algorithm are shown in the following diagram:

(1) Initialization. The CSCA algorithm starts by generating a set of N antibodies.
(2) Training. Repeat the training of all antibodies for G generations (iterations).
(2.1) Selection and pruning. The entire group of antibodies is exposed to the antigen set and a fitness score is computed for each antibody. Then all antibodies are selected and the following three evaluation rules are applied to each antibody:
(2.1.1) Remove from the selected set all antibodies with a misclassification score of zero.
(2.1.2) Antibodies that have zero correct classification and misclassification higher than zero are reassigned to the class of the majority. Fitness is recalculated.
(2.1.3) Remove from the selected set and from the base antibody population all antibodies with a fitness scoring lower than a threshold.
(2.2) Cloning and mutation. The selected set of antibodies is cloned and mutated.
(2.3) Insert new antibodies. Insert the clones generated into the main antibody population. A number of n randomly selected antigens from the antigen set are inserted into the main antibody population, where n is the number of antibodies selected in step (2.1).
(3) Final pruning. The antibody population is exposed to the entire antigen population, fitness scores are computed for each antibody, and pruning of antibodies is performed as described in step (2.1.3).
(4) Select classifier. The final antibody population represents the CSCA classifier. To classify a new pattern, the classification antibodies are exposed to the pattern, then the k most similar (highest affinity) antibodies are selected and a majority vote assigns the class of the pattern.

The artificial immune system CSCA was applied in several virtual screening studies, namely identification of estrogen receptor ligands, recognition of dihydrofolate reductase inhibitors, classification of angiotensin converting enzyme inhibitors, detection of benzodiazepine receptor ligands, and SAR for thermolysin inhibitors. To demonstrate the influence of the user defined parameters on the CSCA predictions, we present the influence of the clonal scale factor CSF, tested in L20%O cross-validation. CSF is used to increase or decrease the number of clones generated for each antibody, and has a default value of one. Low values for CSF promote a low diversity of solutions, whereas high CSF values increase the diversity of the recognition cells.

CSCA was applied for the classification of 232 chemical compounds into estrogen receptor (ER) ligands (131 chemicals, class +1) and compounds that do not bind to the estrogen receptor (101 chemicals, class −1) [129]. The chemical structure was represented with 312 topological indices computed with Molconn-Z. The clonal scale factor was modified between 0.1 and 4 (Table 10a), with the best predictions obtained for CSF = 2 (Ac = 0.6207 and MCC = 0.2057), but with no clear trend apparent for the values that give the best predictions. For example, the next best predictions are obtained for CSF = 0.1 (MCC = 0.1935), whereas the lowest predictions are obtained with CSF = 0.7 (MCC = 0.0416). To investigate the influence of feature selection on the classification abilities of CSCA, 29 structural descriptors were selected with SubsetEvaluation and BestFirst, which results in slightly better predictions for a much lower value of CSF (CSF = 0.1, Ac = 0.6336, MCC = 0.2508; Table 10b).

Dihydrofolate reductase (DHFR) inhibitors may be efficiently identified with CSCA, as was demonstrated for a dataset of 397 chemicals (198 compounds in class +1, pIC_{50} between 9.81 and 6.08; 199 compounds in class −1, pIC_{50} between 6.06 and 3.30) [130]. CSCA classifiers computed with 70 structural descriptors are used to evaluate the effect of the clonal scale factor on the prediction accuracy. Based on the structure of the CSCA algorithm, it should be expected that higher CSF values are useful in identifying better solutions, because more clones are generated, and the system explores a wider diversity of solutions. However, for dihydrofolate reductase inhibitors, the highest predictions are obtained for CSF = 0.2 (Ac = 0.5945 and MCC = 0.1935; Table 11a). Also, for high CSF values, between 0.7 and 3, MCC decreases markedly. A dramatic increase of the CSCA model quality is obtained with a feature selection that reduces the set of structural descriptors to 5 (Table 11b). The best predictions are obtained for a much higher CSF value, namely CSF = 3, with Ac = 0.7834 and MCC = 0.5670. Further tests should be performed with other SAR datasets in order to find the optimum CSF values for various drug screening experiments.

Drug Design, Artificial Intelligence Methods in, Table 10
CSCA prediction statistics for ER SAR models computed for various values of the clonal scale factor CSF

CSF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 312 structural descriptors								
0.1	98	33	44	57	0.7481	0.4356	0.6121	0.1935
0.3	92	39	35	66	0.7023	0.3465	0.5474	0.0519
0.5	97	34	37	64	0.7405	0.3663	0.5776	0.1149
0.7	92	39	34	67	0.7023	0.3366	0.5431	0.0416
1.0	108	23	29	72	0.8244	0.2871	0.5905	0.1326
2.0	112	19	32	69	0.8550	0.3168	0.6207	0.2057
3.0	99	32	31	70	0.7557	0.3069	0.5603	0.0698
4.0	106	25	30	71	0.8092	0.2970	0.5862	0.1238
(b) 29 structural descriptors								
0.1	91	40	56	45	0.6947	0.5545	0.6336	0.2508
0.3	99	32	45	56	0.7557	0.4455	0.6207	0.2119
0.5	97	34	42	59	0.7405	0.4158	0.5991	0.1651
0.7	94	37	47	54	0.7176	0.4653	0.6078	0.1887
1.0	98	33	47	54	0.7481	0.4653	0.6250	0.2226
2.0	98	33	38	63	0.7481	0.3762	0.5862	0.1338
3.0	96	35	41	60	0.7328	0.4059	0.5905	0.1466
4.0	98	33	42	59	0.7481	0.4158	0.6034	0.1738

Drug Design, Artificial Intelligence Methods in, Table 11
CSCA prediction statistics for DHFR SAR models computed for various values of the clonal scale factor CSF

CSF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 70 structural descriptors								
0.1	130	68	92	107	0.6566	0.4623	0.5592	0.1212
0.2	138	60	98	101	0.6970	0.4925	0.5945	0.1935
0.5	129	69	92	107	0.6515	0.4623	0.5567	0.1159
0.7	130	68	87	112	0.6566	0.4372	0.5466	0.0961
1.0	119	79	98	101	0.6010	0.4925	0.5466	0.0940
2.0	142	56	87	112	0.7172	0.4372	0.5768	0.1608
3.0	115	83	99	100	0.5808	0.4975	0.5390	0.0786
4.0	132	66	89	110	0.6667	0.4472	0.5567	0.1167
(b) 5 structural descriptors								
0.1	166	32	144	55	0.8384	0.7236	0.7809	0.5656
0.2	155	43	142	57	0.7828	0.7136	0.7481	0.4975
0.5	151	47	150	49	0.7626	0.7538	0.7582	0.5164
0.7	149	49	148	51	0.7525	0.7437	0.7481	0.4963
1.0	152	46	151	48	0.7677	0.7588	0.7632	0.5265
2.0	154	44	150	49	0.7778	0.7538	0.7657	0.5317
3.0	158	40	153	46	0.7980	0.7688	0.7834	0.5670
4.0	148	50	151	48	0.7475	0.7588	0.7531	0.5063

Drug Design, Artificial Intelligence Methods in, Table 12
CSCA prediction statistics for ACE SAR models with 12 structural descriptors computed for various values of the clonal scale factor CSF

CSF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
0.1	45	12	50	7	0.7895	0.8772	0.8333	0.6692
0.3	43	14	49	8	0.7544	0.8596	0.8070	0.6175
0.5	47	10	48	9	0.8246	0.8421	0.8333	0.6668
0.7	44	13	49	8	0.7719	0.8596	0.8158	0.6340
0.9	47	10	50	7	0.8246	0.8772	0.8509	0.7027
2.0	46	11	43	14	0.8070	0.7544	0.7807	0.5622
3.0	46	11	49	8	0.8070	0.8596	0.8333	0.6676
4.0	46	11	48	9	0.8070	0.8421	0.8246	0.6495

Another set of experiments with CSCA involved the classification of 114 angiotensin converting enzyme (ACE) inhibitors (57 compounds in class +1, pIC_{50} between 9.94 and 6.41; 57 compounds in class −1, pIC_{50} between 6.37 and 2.14) [131]. The chemical structure was encoded with 56 structural descriptors, and the CSF influence was evaluated for 16 values between 0.1 and 4. For all but one CSF values the CSCA classifiers give the same prediction indices, with Ac = 0.8684 and MCC = 0.7510. The CSCA insensitivity to the CSF variation is unexpected, and more experiments are necessary to fully understand this behavior. A feature selection step decreases the pool of structural descriptors to 12 (Table 12), with a slight decrease in the prediction statistics (CSF = 0.9, Ac = 0.8509, MCC = 0.7027). Usually, feature selection provides a smaller set of structural descriptors that increase the predictions of artificial immune systems. The exception encountered for ACE inhibitors should be further investigated to identify possible explanations and better feature selection procedures.

The CSCA immune system was evaluated for the discrimination of 163 benzodiazepine receptor (BZR) ligands (82 high affinity ligands and 81 low affinity ligands) [123]. Starting from a set of 75 structural descriptors, CSF was modified between 0.1 and 4 (Table 13a), with the best results obtained for CSF = 0.7 (Ac = 0.6994 and MCC = 0.3988). A small improvement of the CSCA predictions is obtained by reducing the pool of descriptors to 16 by feature selection (Table 13b). Although the model improvement is not big (Ac = 0.7055 and MCC = 0.4166 for CSF = 2), feature selection is still important because the CSCA model can be computed faster, and the selected descriptors may suggest which molecular features influence the biological activity.

CSCA was also tested for a dataset of 76 thermolysin (THER) inhibitors (38 compounds in class +1, pK_i be-

Drug Design, Artificial Intelligence Methods in, Table 13
CSCA prediction statistics for BZR SAR models computed for various values of the clonal scale factor CSF

CSF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 75 structural descriptors								
0.1	55	27	57	24	0.6707	0.7037	0.6871	0.3746
0.3	52	30	58	23	0.6341	0.7160	0.6748	0.3513
0.5	52	30	58	23	0.6341	0.7160	0.6748	0.3513
0.7	57	25	57	24	0.6951	0.7037	0.6994	0.3988
1.0	48	34	65	16	0.5854	0.8025	0.6933	0.3971
2.0	48	34	59	22	0.5854	0.7284	0.6564	0.3169
3.0	54	28	54	27	0.6585	0.6667	0.6626	0.3252
4.0	58	24	52	29	0.7073	0.6420	0.6748	0.3501
(b) 16 structural descriptors								
0.1	56	26	55	26	0.6829	0.6790	0.6810	0.3619
0.3	62	20	51	30	0.7561	0.6296	0.6933	0.3890
0.5	53	29	57	24	0.6463	0.7037	0.6748	0.3506
0.7	60	22	54	27	0.7317	0.6667	0.6994	0.3993
1.0	61	21	45	36	0.7439	0.5556	0.6503	0.3050
2.0	65	17	50	31	0.7927	0.6173	0.7055	0.4166
3.0	58	24	55	26	0.7073	0.6790	0.6933	0.3865
4.0	58	24	51	30	0.7073	0.6296	0.6687	0.3380

Drug Design, Artificial Intelligence Methods in, Table 14
CSCA prediction statistics for THER SAR models computed for various values of the clonal scale factor CSF

CSF	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 64 structural descriptors								
0.1	24	14	26	12	0.6316	0.6842	0.6579	0.3162
0.5	24	14	26	12	0.6316	0.6842	0.6579	0.3162
1.0	24	14	26	12	0.6316	0.6842	0.6579	0.3162
2.0	24	14	26	12	0.6316	0.6842	0.6579	0.3162
2.5	19	19	27	11	0.5000	0.7105	0.6053	0.2154
3.0	24	14	26	12	0.6316	0.6842	0.6579	0.3162
3.5	25	13	26	12	0.6579	0.6842	0.6711	0.3422
4.0	24	14	26	12	0.6316	0.6842	0.6579	0.3162
(b) 10 structural descriptors								
0.1	20	18	27	11	0.5263	0.7105	0.6184	0.2410
0.5	19	19	31	7	0.5000	0.8158	0.6579	0.3328
1.0	20	18	24	14	0.5263	0.6316	0.5789	0.1588
2.0	21	17	32	6	0.5526	0.8421	0.6974	0.4124
2.5	19	19	27	11	0.5000	0.7105	0.6053	0.2154
3.0	21	17	31	7	0.5526	0.8158	0.6842	0.3819
3.5	13	25	32	6	0.3421	0.8421	0.5921	0.2127
4.0	16	22	32	6	0.4211	0.8421	0.6316	0.2901

tween 10.17 and 5.55; 38 compounds in class −1, pK_i between 5.16 and 0.52) and 64 structural descriptors [132]. For 14 out of 16 CSF values tested in this experiment, the CSCA classifiers have identical predictions, with Ac = 0.6711 and MCC = 0.3162. The best predictions are obtained for CSF = 3.5, with slightly higher prediction statistics, namely MCC = 0.3422 (Table 14a). Feature selection reduces the number of descriptors to 10, which results in a minor improvement (CSF = 2, Ac = 0.6974, MCC = 0.4124, Table 14b).

IMMUNOS

Carter developed the IMMUNOS-81 artificial immune systems as an instance based classifier with some similarity to k-nearest neighbor classifiers [106]. Brownlee extended this algorithm by adding elements from other AIS classifiers, such as cloning and hypermutation, to obtain IMMUNOS-99 [107]. A brief description of the IMMUNOS-99 consists of the following steps:

(1) Initialization. The training group of antigens is divided into groups based on class label.

(2) Train B-cell groups. The final IMMUNOS classifier consists of a B-cell population for each class represented in the training set of antigens. Each B-cell population is generated and trained independent of the other B-cell populations. Steps (2.1) and (2.2) are repeated C times, where C is the number of antigen classes.

(2.1) Create B-cell population. Generate a B-cell population for the antigen class under training. A fraction of the antigen population from that class is used as seed for the B-cell population.

(2.2) Training. Train the B-cell class for G generations (iterations).

(2.2.1) Expose population. The B-cell population is exposed to all antigens from all classes, and an affinity value is computed for each B-cell/antigen comparison. A rank-based scoring is established for each B-cell.

(2.2.2) Compute fitness. A fitness index is computed for each B-cell, based on the rank scores for antigens in the same class and the rank scores for antigens in all other classes. B-cells that recognize better antigens from the same class have fitness score higher than one, whereas B-cells that recognize better antigens from other classes have fitness score lower than one.

(2.2.3) Pruning. A user-defined parameter, between [0, 1], sets the minimum fitness score of a B-cell. All B-cells with fitness scores lower than this threshold are removed from the population.

(2.2.4) Affinity maturation. After pruning, the B-cell population contains only cells that can identify antigens from the same class. To improve the B-cell recog-

nition ability, the system undergoes an affinity maturation process based on cloning and hypermutation.

(2.2.4.1) Order population. The B-cell population is ordered in the descending order of the fitness scores.

(2.2.4.2) Generate clones. Each B-cell is cloned proportional to its fitness rank. The rank ratio for a B-cell is:

$$r_i = \frac{\text{rank}}{S}$$

where r_i is the rank ration of the ith B-cell, rank is the actual index of the B-cell in the ordered sequence, rank $\in [1, S]$, and S is the total number of B-cell in the population (class). The number of clones generated for each B-cell is:

$$NC_i = \left\lfloor \frac{r_i}{\sum_{j=1}^{S} r_j} N + 0.5 \right\rfloor$$

where N is the total number of antigens in the same class.

(2.2.4.3) Mutate clones. The clones are mutated by the inverse of the B-cell rank ratios. As a result of this procedure, clones of B-cells with higher ranks undergo small mutations, whereas clones of B-cells with lower ranks go through large mutations. All clones generated are added to the B-cell population.

(2.2.5) Insert random antigens. In order to increase the diversity of the B-cell population, a random selection of antigens from the same class is added to the B-cell pool. The number of antigens added is equal to the number of B-cells deleted during the pruning process from step (2.2.3). The diversity introduced by the antigen-based B-cells is particularly useful whenever the affinity maturation process converges to a limited number of B-cells.

(3) Final pruning. This step removes B-cells with low fitness after the system finishes the training for each antigen class and for the set number of generations G.

(3.1) Compute fitness. Each B-cell population (class) is exposed to all antigens, one antigen at a time, and only the best matching B-cells receive a score.

(3.2) Pruning. Similarly with the pruning process from step (2.2.3), all B-cells with low fitness scores lower are removed from the population.

(4) Select classifier. The populations of B-cells that survive the final pruning represent the classifier for new, unknown antigens. During the classification process, each B-cell class is exposed to the unknown antigen, and an avidity index is computed. Then the B-cell populations compete for the unknown antigen that takes the class label of the B-cell population with the highest avidity index.

The IMMUNOS-99 system was evaluated in several drug design studies, namely structure-activity relationships for acetylcholinesterase inhibitors, virtual screening of cyclooxygenase-2 inhibitors, recognition of benzodiazepine receptor ligands, and classification of thrombin inhibitors. All examples presented here investigate the influence of the seed population percentage SPP. SPP is a user defined parameter that specifies the percentage of the antigen population from each class that is used as seed for the B-cell population. If SPP = 100% then the initial B-cell population is identical with the antigen population in the same class. The influence of the SPP parameter was investigated in series of L20%O cross-validation experiments. For each drug design dataset, the IMMUNOS-99 classifier was trained for 19 values of the SPP parameter, between 0.05 and 0.95.

IMMUNOS-99 structure-activity relationships were developed for a dataset of 111 acetylcholinesterase (AChE) inhibitors characterized by 63 structural descriptors [133]. The classifiers were trained to discriminate between 55 inhibitors in class +1 (pIC_{50} between 9.52 and 6.87) and 56 inhibitors in class −1 (pIC_{50} between 6.84 and 4.27). The prediction MCC increases from 0.1349 for SPP = 0.05, has a maximum of 0.2847 for SPP = 0.35, and then decreases to 0.2110 for SPP = 0.95 (Table 15a). These results suggest that seeding the B-cell population with less than half of the antigen population improves the prediction statistics. The number of structural descriptors is reduced to 9 by feature selection, which results in a slight decrease in the IMMUNOS-99 predictions (Table 15b).

The virtual screening of cyclooxygenase-2 (COX2) inhibitors may be efficiently done with IMMUNOS-99, as shown for 322 compounds (162 compounds in class +1, pIC_{50} between 9 and 6.60; 160 compounds in class −1, pIC_{50} between 6.59 and 4) [134]. Starting from a set of 74 structural descriptors, several IMMUNOS-99 classifiers were developed to study the influence of the SPP parameter (Table 16a). The results obtained from this series of experiments indicate that the prediction statistics have similar values for a wide range of the SPP parameter, with a small improvement for SPP = 0.45. The number of structural descriptors was reduced by feature selection to 12 important descriptors, thus improving the predictions of the IMMUNOS-99 classifiers (Table 16b). The best results are obtained for SPP = 0.75 (Ac = 0.6429, MCC = 0.3855), but FP is still too large, i. e., too many inactive compounds are predicted as active.

Drug Design, Artificial Intelligence Methods in, Table 15
IMMUNOS-99 prediction statistics for AChE SAR models computed for various values of the seed population percentage SPP

SPP	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 63 structural descriptors								
0.05	30	25	33	23	0.5455	0.5893	0.5676	0.1349
0.10	45	10	22	34	0.8182	0.3929	0.6036	0.2329
0.15	44	11	18	38	0.8000	0.3214	0.5586	0.1382
0.35	45	10	25	31	0.8182	0.4464	0.6306	0.2847
0.50	45	10	22	34	0.8182	0.3929	0.6036	0.2329
0.70	43	12	23	33	0.7818	0.4107	0.5946	0.2072
0.85	44	11	23	33	0.8000	0.4107	0.6036	0.2286
0.95	44	11	23	33	0.8000	0.4107	0.6036	0.2286
(b) 9 structural descriptors								
0.05	35	20	21	35	0.6364	0.3750	0.5045	0.0118
0.10	41	14	18	38	0.7455	0.3214	0.5315	0.0738
0.15	47	8	15	41	0.8545	0.2679	0.5586	0.1510
0.35	48	7	6	50	0.8727	0.1071	0.4865	-0.0313
0.50	53	2	3	53	0.9636	0.0536	0.5045	0.0415
0.70	55	0	3	53	1.0000	0.0536	0.5225	0.1652
0.85	54	1	3	53	0.9818	0.0536	0.5135	0.0949
0.95	54	1	2	54	0.9818	0.0357	0.5045	0.0541

Drug Design, Artificial Intelligence Methods in, Table 16
IMMUNOS-99 prediction statistics for COX2 SAR models computed for various values of the seed population percentage SPP

SPP	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 74 structural descriptors								
0.05	113	49	72	88	0.6975	0.4500	0.5745	0.1523
0.15	104	58	75	85	0.6420	0.4688	0.5559	0.1124
0.30	93	69	91	69	0.5741	0.5687	0.5714	0.1428
0.45	90	72	96	64	0.5556	0.6000	0.5776	0.1557
0.60	93	69	93	67	0.5741	0.5813	0.5776	0.1553
0.75	90	72	94	66	0.5556	0.5875	0.5714	0.1431
0.85	93	69	89	71	0.5741	0.5563	0.5652	0.1303
0.95	93	69	90	70	0.5741	0.5625	0.5683	0.1366
(b) 12 structural descriptors								
0.05	160	2	25	135	0.9877	0.1562	0.5745	0.2596
0.15	159	3	34	126	0.9815	0.2125	0.5994	0.3041
0.30	158	4	43	117	0.9753	0.2687	0.6242	0.3456
0.45	159	3	41	119	0.9815	0.2562	0.6211	0.3461
0.60	159	3	46	114	0.9815	0.2875	0.6366	0.3744
0.75	159	3	48	112	0.9815	0.3000	0.6429	0.3855
0.85	157	5	50	110	0.9691	0.3125	0.6429	0.3742
0.95	158	4	50	110	0.9753	0.3125	0.6460	0.3852

Drug Design, Artificial Intelligence Methods in, Table 17
IMMUNOS-99 prediction statistics for BZR SAR models computed for various values of the seed population percentage SPP

SPP	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 75 structural descriptors								
0.05	48	34	47	34	0.5854	0.5802	0.5828	0.1656
0.15	35	47	68	13	0.4268	0.8395	0.6319	0.2922
0.25	39	43	67	14	0.4756	0.8272	0.6503	0.3232
0.35	35	47	70	11	0.4268	0.8642	0.6442	0.3233
0.50	36	46	69	12	0.4390	0.8519	0.6442	0.3191
0.65	34	48	69	12	0.4146	0.8519	0.6319	0.2960
0.75	36	46	71	10	0.4390	0.8765	0.6564	0.3506
0.95	36	46	70	11	0.4390	0.8642	0.6503	0.3347
(b) 16 structural descriptors								
0.05	57	25	39	42	0.6951	0.4815	0.5890	0.1808
0.15	55	27	51	30	0.6707	0.6296	0.6503	0.3006
0.25	54	28	51	30	0.6585	0.6296	0.6442	0.2883
0.35	55	27	49	32	0.6707	0.6049	0.6380	0.2763
0.50	53	29	49	32	0.6463	0.6049	0.6258	0.2515
0.65	51	31	51	30	0.6220	0.6296	0.6258	0.2516
0.75	50	32	51	30	0.6098	0.6296	0.6196	0.2394
0.95	50	32	51	30	0.6098	0.6296	0.6196	0.2394

The IMMUNOS-99 immune system was also tested for the dataset of benzodiazepine receptor (BZR) ligands (82 high affinity ligands and 81 low affinity ligands) [123]. The best predictions for the entire pool of 75 structural descriptors were obtained for SPP = 0.75 (Ac = 0.6564, MCC 0.3506; Table 17a). To evaluate the importance of feature selection, the number of structural descriptors was reduced to 16 and the entire analysis was repeated for the full range of SPP values. Although FN decreases (active compounds predicted inactive), FP increases which results in slightly worse predictions (Table 17b). The best predictions are obtained also for SPP = 0.15 (Ac = 0.6503, MCC = 0.3006), but the results suggest that IMMUNOS-99 predictions do not improve with feature selection.

The classification of thrombin (THR) inhibitors with IMMUNOS-99 was investigated for 88 chemicals (44 compounds in class +1, pK_i between 8.48 and 6.70; 44 compounds in class −1, pK_i between 6.68 and 4.36) and 66 structural descriptors [135]. The prediction statistics indicate that the IMMUNOS-99 is not very successful in discriminating thrombin inhibitors from non-inhibitors (Table 18a). In all 19 experiments that explore the influence of the SPP parameter, almost all chemical compounds are predicted in the class +1 (inhibitors). As a result, FN is small (which is good) but FP is very large (which is bad), and the overall statistics are low. A maximum is identified for SPP = 0.15 (Ac = 0.5568, MCC = 0.2100). Fea-

Drug Design, Artificial Intelligence Methods in, Table 18
IMMUNOS-99 prediction statistics for THR SAR models computed for various values of the seed population percentage SPP

SPP	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
(a) 66 structural descriptors								
0.05	38	6	10	34	0.8636	0.2273	0.5455	0.1179
0.15	43	1	6	38	0.9773	0.1364	0.5568	0.2100
0.25	44	0	2	42	1.0000	0.0455	0.5227	0.1525
0.40	44	0	2	42	1.0000	0.0455	0.5227	0.1525
0.50	43	1	1	43	0.9773	0.0227	0.5000	0.0000
0.65	43	1	2	42	0.9773	0.0455	0.5114	0.0626
0.80	44	0	1	43	1.0000	0.0227	0.5114	0.1072
0.95	43	1	2	42	0.9773	0.0455	0.5114	0.0626
(b) 7 structural descriptors								
0.05	41	3	7	37	0.9318	0.1591	0.5455	0.1432
0.15	44	0	6	38	1.0000	0.1364	0.5682	0.2705
0.25	44	0	3	41	1.0000	0.0682	0.5341	0.1879
0.40	44	0	5	39	1.0000	0.1136	0.5568	0.2454
0.50	44	0	5	39	1.0000	0.1136	0.5568	0.2454
0.65	44	0	5	39	1.0000	0.1136	0.5568	0.2454
0.80	44	0	4	40	1.0000	0.0909	0.5455	0.2182
0.95	44	0	3	41	1.0000	0.0682	0.5341	0.1879

ture selection reduces the pool of descriptors to 7, and results in slightly better models (Table 18b). FP is still too large for the whole range of SPP values, which explains the low values for the statistical indices. Compared with the other three artificial immune systems, IMMUNOS-99 seems to be the most difficult to tune in order to obtain good predictions. Feature selection has no or small effect in improving IMMUNOS-99 models, which suggests that other algorithms should be investigated to reduce the pool of structural descriptors.

Future Directions

Pharmaceutical drug discovery use computer-assisted molecular design to increase the chances of bringing a drug on the market, and to lower the research and development costs. Computational models are used to simulate the physical, chemical, biological, and toxicological properties of drug candidates, thus replacing expensive and time-consuming large scale experiments. The entire process consists of iterative steps, in which experimental results are used to train computational models, which in turn suggest novel molecules that are synthesized and tested in the laboratory. We reviewed here the most important artificial intelligence algorithms used in drug design, namely genetic algorithms, ant colony optimization, particle swarm optimization, and artificial immune systems. The main advantage of artificial intelligence algorithms is their ability to explore search spaces of high dimensionality, and to identify the global optimum for complex and difficult problems. Genetic algorithms have a long history of applications in QSAR and drug design, and their operation is thoroughly explored. The other artificial intelligence algorithms were adopted only recently, but they already demonstrated strong results that make them competitors for GA. More important, ACO, PSO and AIS bring new simulation capabilities, thus complementing GA. A promising direction of development is a combined use of these artificial intelligence algorithms that could provide better predictions of molecular properties. Another source of improvement might come from the integration of the molecular graph into the artificial intelligence algorithms, which would complement (or even substitute) the use of structural descriptors.

Bibliography

1. Holland J (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor
2. Goldberg DE (1989) Genetic Algorithms in Search, Optimization & Machine Learning. Addison Wesley, Reading
3. Jones G (1998) Genetic and evolutionary algorithms. In: Schleyer PvR, Allinger NL, Clark T, Gasteiger J, Kollman PA, Schaefer III HF, Schreiner PR (eds) The Encyclopedia of Computational Chemistry. Wiley, Chichester, pp 1127–1136
4. Terfloth L, Gasteiger J (2001) Neural networks and genetic algorithms in drug design. Drug Discov Today 6:S102–S108
5. von Homeyer A (2003) Evolutionary algorithms and their applications in chemistry. In: Gasteiger J (ed) Handbook of Chemoinformatics, vol 3. Wiley-VCH, Weinheim, pp 1239–1280
6. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: Optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B Cybern 26:29–41
7. Dorigo M, Gambardella LM (1997) Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1:53–66
8. Dorigo M, Di Caro G, Gambardella LM (1999) Ant algorithms for discrete optimization. Artif Life 5:137–172
9. Dorigo M, Stützle T (2004) Ant Colony Optimization. MIT Press, Cambridge
10. Dorigo M, Blum C (2005) Ant colony optimization theory: A survey. Theor Comput Sci 344:243–278
11. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, vol 4. pp 1942–1948
12. Banks A, Vincent J, Anyakoha C (2007) A review of particle swarm optimization Part I: background and development. Nat Comput 6:467–484
13. Banks A, Vincent J, Anyakoha C (2008) A review of particle swarm optimization Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Nat Comput 7:109–124

14. Chuang LY, Chang HW, Tu CJ, Yang CH (2008) Improved binary PSO for feature selection using gene expression data. Comput Biol Chem 32:29–38
15. Namasivayam V, Günther R (2007) PSO@AUTODOCK: A fast flexible molecular docking program based on swarm intelligence. Chem Biol Drug Des 70:475–484
16. Agrafiotis DK, Cedeño W (2002) Feature selection for structure-activity correlation using binary particle swarms. J Med Chem 45:1098–1107
17. Huang J, Ma G, Muhammad I, Cheng Y (2007) Identifying P-glycoprotein substrates using a support vector machine optimized by a particle swarm. J Chem Inf Model 47:1638–1647
18. Shen Q, Shi WM, Yang XP, Ye BX (2006) Particle swarm algorithm trained neural network for QSAR studies of inhibitors of platelet-derived growth factor receptor phosphorylation. Eur J Pharm Sci 28:369–376
19. Hunt JE, Cooke DE (1996) Learning using an artificial immune system. J Netw Comput Appl 19:189–212
20. de Castro LN, Von Zuben FJ (1999) Artificial immune systems: Part I Basic theory and applications. FEEC/UNICAMP, Brazil
21. de Castro LN, Von Zuben FJ (2000) Artificial immune systems: Part II A survey of applications. FEEC/UNICAMP, Brazil
22. Timmis J, Neal M, Hunt J (2000) An artificial immune system for data analysis. Biosystems 55:143–150
23. Chao DL, Forrest S (2003) Information immune systems. Genet Programm Evolv Mach 4:311–331
24. de Castro LN, Timmis JI (2003) Artificial immune systems as a novel soft computing paradigm. Soft Comput 7:526–544
25. Musilek P, Lau A, Reformat M, Wyard-Scott L (2006) Immune programming. Inf Sci 176:972–1002
26. Timmis J (2007) Artificial immune systems – Today and tomorrow. Nat Comput 6:1–18
27. Forrest S, Beauchemin C (2007) Computer immunology. Immunol Rev 216:176–197
28. Dasgupta D (1999) Artificial Immune Systems and Their Applications. Springer, Berlin
29. de Castro LN, Timmis J (2002) Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Berlin
30. Tarakanov AO, Skormin VA, Sokolova SP (2003) Immunocomputing: Principles and Applications. Springer, Berlin
31. Ishida Y (2004) Immunity-Based Systems. Springer, Berlin
32. Timmis J, Bentley P, Hart E (2003) Artificial Immune Systems: Second International Conference, ICARIS 2003, Edinburgh, September 1–3. Lecture Notes in Computer Science, vol 2787. Springer, Berlin
33. Nicosia G, Cutello V, Bentley PJ, Timmis JI (2004) Artificial Immune Systems: Third International Conference, ICARIS 2004, Catania, September 13–16. Lecture Notes in Computer Science, vol 3239. Springer, Berlin
34. Jacob C, Pilat ML, Bentley PJ, Timmis J (2005) Artificial Immune Systems: 4th International Conference, ICARIS 2005, Banff, August 14–17. Lecture Notes in Computer Science, vol 3627. Springer, Berlin
35. Bersini H, Carneiro J (2006) Artificial Immune Systems: 5th International Conference, ICARIS 2006, Oeiras, September 4–6. Lecture Notes in Computer Science, vol 4163. Springer, Berlin
36. Ando S, Iba H (2004) Classification of gene expression profile using combinatory method of evolutionary computation and machine learning. Genet Programm Evolv Mach 5:145–156
37. Bezerra GB, CanÇado GMA, Menossi M, de Castro LN, Von Zuben FJ (2005) Recent advances in gene expression data clustering: A case study with comparative results. Genet Mol Res 4:514–524
38. Tsankova D, Georgieva V, Kasabov N (2005) Artificial immune networks as a paradigm for classification and profiling of gene expression data. J Comput Theor Nanosci 2:543–550
39. Şahan S, Polat K, Kodaz H, Güneş S (2007) A new hybrid method based on fuzzy-artificial immune system and k-nn algorithm for breast cancer diagnosis. Comput Biol Med 37:415–423
40. Polat K, Güneş S (2008) Computer aided medical diagnosis system based on principal component analysis and artificial immune recognition system classifier algorithm. Expert Syst Appl 34:773–779
41. Polat K, Şahan S, Güneş S (2006) A new method to medical diagnosis: Artificial immune recognition system (AIRS) with fuzzy weighted pre-processing and application to ECG arrhythmia. Expert Syst Appl 31:264–269
42. Latifoglu F, Şahan S, Kara S, Güneş S (2007) Diagnosis of atherosclerosis from carotid artery Doppler signals as a real-world medical application of artificial immune systems. Expert Syst Appl 33:786–793
43. Cutello V, Nicosia G, Pavone M, Timmis J (2007) An immune algorithm for protein structure prediction on lattice models. IEEE Trans Evol Comput 11:101–117
44. Anile AM, Cutello V, Narzisi G, Nicosia G, Spinella S (2007) Determination of protein structure and dynamics combining immune algorithms and pattern search methods. Nat Comput 6:55–72
45. Morris GM, Goodsell DS, Halliday RS, Huey R, Hart WE, Belew RK, Olson AJ (1998) Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. J Comput Chem 19:1639–1662
46. Wang R, Gao Y, Lai LH (2000) LigBuilder: A multi-purpose program for structure-based drug design. J Mol Model 6:498–516
47. So SS, Karplus M (1996) Evolutionary optimization in quantitative structure-activity relationship: An application of genetic neural networks. J Med Chem 39:1521–1530
48. Venkatasubramanian V, Chan K, Caruthers JM (1995) Evolutionary design of molecules with desired properties using the genetic algorithm. J Chem Inf Comput Sci 35:188–195
49. Sundaram A, Venkatasubramanian V (1998) Parametric sensitivity and search-space characterization studies of genetic algorithms for computer-aided polymer design. J Chem Inf Comput Sci 38:1177–1191
50. Gillet VJ, Willett P, Bradshaw J, Green DVS (1999) Selecting combinatorial libraries to optimize diversity and physical properties. J Chem Inf Comput Sci 39:169–177
51. Ivanciuc O, Ivanciuc T, Cabrol-Bass D (2002) QSAR for dihydrofolate reductase inhibitors with molecular graph structural descriptors. J Mol Struct (Theochem) 582:39–51
52. Wegner JK, Fröhlich H, Zell A (2004) Feature selection for descriptor based classification models, 2. Human intestinal absorption (HIA). J Chem Inf Comput Sci 44:931–939
53. Weber L (1998) Evolutionary combinatorial chemistry: application of genetic algorithms. Drug Discov Today 3:379–385
54. Weber L (2005) Current status of virtual combinatorial library design. QSAR Comb Sci 24:809–823
55. Gallop MA, Barrett RW, Dower WJ, Fodor SPA, Gordon EM (1994) Applications of combinatorial technologies to drug discovery, 1. Background and peptide combinatorial libraries. J Med Chem 37:1233–1251

56. Gordon EM, Barrett RW, Dower WJ, Fodor SPA, Gallop MA (1994) Applications of combinatorial technologies to drug discovery, 2. Combinatorial organic-synthesis, library screening strategies, and future directions. J Med Chem 37:1385–1401
57. Weber L (1998) Applications of genetic algorithms in molecular diversity. Curr Opin Chem Biol 2:381–385
58. Illgen K, Enderle T, Broger C, Weber L (2000) Simulated molecular evolution in a full combinatorial library. Chem Biol 7:433–441
59. Ugi I, Almstetter M, Bock H, Dömling A, Ebert B, Gruber B, Hanusch-Kompa C, Heck S, Kehagia-Drikos K, Lorenz K, Papathoma S, Raditschnig R, Schmid T, Werner B, von Zychlinski A (1998) MCR XVII. Three types of MCRs and the libraries – Their chemistry of natural events and preparative chemistry. Croat Chem Acta 71:527–547
60. Weber L (2002) Multi-component reactions and evolutionary chemistry. Drug Discov Today 7:143–147
61. Globus A, Lawtonb J, Wipke T (1999) Automatic molecular design using evolutionary techniques. Nanotechnology 10:290–299
62. Brown N, McKay B, Gilardoni F, Gasteiger J (2004) A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. J Chem Inf Comput Sci 44:1079–1087
63. Brown N, McKay B, Gasteiger J (2006) A novel workflow for the inverse QSPR problem using multiobjective optimization. J Comput Aided Mol Des 20:333–341
64. Lewis RA (2005) A general method for exploiting QSAR models in lead optimization. J Med Chem 48:1638–1648
65. Izrailev S, Agrafiotis D (2001) A novel method for building regression tree models for QSAR based on artificial ant colony systems. J Chem Inf Comput Sci 41:176–180
66. Izrailev S, Agrafiotis DK (2002) Variable selection for QSAR by artificial ant colony systems. SAR QSAR Environ Res 13:417–423
67. Shelokar PS, Jayaraman VK, Kulkarni BD (2004) An ant colony approach for clustering. Anal Chim Acta 509:187–195
68. He Y, Chen D, Zhao W (2006) Ensemble classifier system based on ant colony algorithm and its application in chemical pattern classification. Chemom Intell Lab Syst 80:39–49
69. Korb O, Stützle T, Exner TE (2006) PLANTS: Application of ant colony optimization to structure-based drug design. Ant Colony Optimization and Swarm Intelligence. Proceedings, TS5 vol 4150. pp 247–258
70. Shen Q, Jiang JH, Tao JC, Shen GL, Yu RQ (2005) Modified ant colony optimization algorithm for variable selection in QSAR modeling: QSAR studies of cyclooxygenase inhibitors. J Chem Inf Model 45:1024–1029
71. Gunturi SB, Narayanan R, Khandelwal A (2006) In silico ADME modelling 2: Computational models to predict human serum albumin binding affinity using ant colony systems. Bioorg Med Chem 14:4118–4129
72. Shi WM, Shen Q, Kong W, Ye BX (2007) QSAR analysis of tyrosine kinase inhibitor using modified ant colony optimization and multiple linear regression. Eur J Med Chem 42:81–86
73. Karpenko O, Shi J, Dai Y (2005) Prediction of MHC class II binders using the ant colony search strategy. Artif Intell Med 35:147–156
74. Hernandez P, Gras R, Frey J, Appel RD (2003) Popitam: Towards new heuristic strategies to improve protein identification from tandem mass spectrometry data. Proteomics 3:870–878
75. Shen Q, Shi WM, Kong W, Ye BX (2007) A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification. Talanta 71:1679–1683
76. Call ST, Zubarev DY, Boldyrev AI (2007) Global minimum structure searches via particle swarm optimization. J Comput Chem 28:1177–1186
77. Chang BCH, Ratnaweera A, Halgamuge SK, Watson HC (2004) Particle swarm optimisation for protein motif discovery. Genet Programm Evolv Mach 5:203–214
78. Chen K, Li T, Cao T (2006) Tribe-PSO: A novel global optimization algorithm and its application in molecular docking. Chemom Intell Lab Syst 82:248–259
79. Chen HM, Liu BF, Huang HL, Hwang SF, Ho SY (2007) SODOCK: Swarm optimization for highly flexible protein-ligand docking. J Comput Chem 28:612–623
80. Cedeño W, Agrafiotis DK (2003) Using particle swarms for the development of QSAR models based on K-nearest neighbor and kernel regression. J Comput Aided Mol Des 17:255–263
81. Lü JX, Shen Q, Jiang JH, Shen GL, Yu RQ (2004) QSAR analysis of cyclooxygenase inhibitor using particle swarm optimization and multiple linear regression. J Pharm Biomed Anal 35:679–687
82. Shen Q, Jiang JH, Jiao CX, Shen GL, Yu RQ (2004) Modified particle swarm optimization algorithm for variable selection in MLR and PLS modeling: QSAR studies of antagonism of angiotensin II antagonists. Eur J Pharm Sci 22:145–152
83. Lin WQ, Jiang JH, Shen Q, Shen GL, Yu RQ (2005) Optimized block-wise variable combination by particle swarm optimization for partial least squares modeling in quantitative structure-activity relationship studies. J Chem Inf Model 45:486–493
84. Lin L, Lin WQ, Jiang JH, Shen GL, Yu RQ (2005) QSAR analysis of substituted bis[(acridine-4-carboxamide)propyl] methylamines using optimized block-wise variable combination by particle swarm optimization for partial least squares modeling. Eur J Pharm Sci 25:245–254
85. Hu L, Wu H, Lin W, Jiang J, Yu R (2007) Quantitative structure-activity relationship studies for the binding affinities of imidazobenzodiazepines for the $\alpha 6$ benzodiazepine receptor isoform utilizing optimized blockwise variable combination by particle swarm optimization for partial least squares modeling. QSAR Comb Sci 26:92–101
86. Shen Q, Jiang JH, Jiao CX, Huan SY, Shen GL, Yu RQ (2004) Optimized partition of minimum spanning tree for piecewise modeling by particle swarm algorithm. QSAR studies of antagonism of angiotensin II antagonists. J Chem Inf Comput Sci 44:2027–2031
87. Lin WQ, Jiang JH, Shen Q, Wu HL, Shen GL, Yu RQ (2005) Piecewise hypersphere modeling by particle swarm optimization in QSAR studies of bioactivities of chemical compounds. J Chem Inf Model 45:535–541
88. Lin L, Lin WQ, Jiang JH, Zhou YP, Shen GL, Yu RQ (2005) QSAR analysis of a series of 2-aryl(heteroaryl)-2,5-dihydropyrazolo[4,3-c]quinolin-3-(3H)-ones using piecewise hyper-sphere modeling by particle swarm optimization. Anal Chim Acta 552:42–49
89. Xu L, Jiang JH, Lin WQ, Zhou YP, Wu HL, Shen GL, Yu RQ

TS5 Please provide series name.

Editor's or typesetter's annotations (will be removed before the final TEX run)

(2007) Optimized sample-weighted partial least squares. Talanta 71:561–566
90. Shen Q, Jiang JH, Jiao CX, Lin WQ, Shen GL, Yu RQ (2004) Hybridized particle swarm algorithm for adaptive structure training of multilayer feed-forward neural network: QSAR studies of bioactivity of organic compounds. J Comput Chem 25:1726–1735
91. Zhou YP, Jiang JH, Lin WQ, Zou HY, Wu HL, Shen GL, Yu RQ (2006) Adaptive configuring of radial basis function network by hybrid particle swarm algorithm for QSAR studies of organic compounds. J Chem Inf Model 46:2494–2501
92. Zhou YP, Jiang JH, Lin WQ, Xu L, Wu HL, Shen GL, Yu RQ (2007) Artificial neural network-based transformation for nonlinear partial least-square regression with application to QSAR studies. Talanta 71:848–853
93. Meissner M, Schmuker M, Schneider G (2006) Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics 7:125
94. Ivanciuc O (2007) Applications of support vector machines in chemistry. In: Lipkowitz KB, Cundari TR (eds) Reviews in Computational Chemistry, vol 23. Wiley-VCH, Weinheim, pp 291–400
95. Lin WQ, Jiang JH, Zhou YP, Wu HL, Shen GL, Yu RQ (2007) Support vector machine based training of multilayer feedforward neural networks as optimized by particle swarm algorithm: Application in QSAR studies of bioactivity of organic compounds. J Comput Chem 28:519–527
96. Tang LJ, Zhou YP, Jiang JH, Zou HY, Wu HL, Shen GL, Yu RQ (2007) Radial basis function network-based transform for a nonlinear support vector machine as optimized by a particle swarm optimization algorithm with application to QSAR studies. J Chem Inf Model 47:1438–1445
97. de Castro LN (2004) Dynamics of an artificial immune network. J Exp Theor Artif Intell 16:19–39
98. Bezerra GB, de Castro LN, Von Zuben FJ (2004) A hierarchical immune network applied to gene expression data. In: Nicosia G, Cutello V, Bentley PJ, Timmis JI (eds) Artificial Immune Systems: Third International Conference, ICARIS 2004. Catania, September 13–16. TS5 vol 3239. Springer, Berlin, pp 14–27
99. Watkins A, Timmis J, Boggess L (2004) Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm. Genet Programm Evolv Mach 5:291–317
100. Meng L, van der Putten P, Wang H (2005) A comprehensive benchmark of the artificial immune recognition system (AIRS). Advanced Data Mining and Applications, ProceedingsTS5, vol 3584. pp 575–582
101. Watkins AB (2001) AIRS: A resource limited artificial immune classifier. Department of Computer Science, vol MS. Mississippi State University, pp 81
102. Watkins AB (2005) Exploiting immunological metaphors in the development of serial, parallel and distributed learning algorithms, vol PhD. University of Kent, pp 314
103. de Castro LN, Von Zuben FJ (2000) The clonal selection algorithm with engineering applications. In: Whitley D, Goldberg D, Cantu-Paz E, Spector L, Parmee I, Beyer HG (eds) GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference, July 10–12. Las Vegas, Morgan Kaufmann, pp 36–37
104. de Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 6:239–251
105. Brownlee J (2005) Clonal selection theory & CLONAG. The clonal selection classification algorithm (CSCA). Centre for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology (SUT). Victoria
106. Carter JH (2000) The immune system as a model for pattern recognition and classification. J Am Med Inf Assoc 7:28–41
107. Brownlee J (2005) Immunos-81. The misunderstood artificial immune system. Centre for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology (SUT). Victoria
108. Witten IH, Frank E (2005) Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco, pp 525
109. Brownlee J (2005) Artificial immune recognition system (AIRS). A review and analysis. Centre for Intelligent Systems and Complex Processes (CISCP), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology (SUT). Victoria
110. Watkins A, Timmis J (2002) Artificial immune recognition system (AIRS): Revisions and refinements. Artificial Immune Systems: First International Conference, ICARIS 2002. University of Kent at Canterbury, pp 173–181
111. Fenichel RR, Malik M, Antzelevitch C, Sanguinetti M, Roden DM, Priori SG, Ruskin JN, Lipicky RJ, Cantilena LR (2004) Drug-induced torsades de pointes and implications for drug development. J Cardiovasc Electrophysiol 15:475–495
112. Ivanciuc O (2006) Artificial immune system classification of drug-induced torsade de pointes with AIRS (artificial immune recognition system). Internet Electron J Mol Des 5:488–502
113. Ivanciuc O (2007) Artificial immune systems in drug design: Structure-activity relationships for torsade de pointes with AIRS (artificial immune recognition system). Internet Electron J Mol Des 6:47–62
114. Stenberg P, Luthman K, Artursson P (2000) Virtual screening of intestinal drug permeability. J Control Release 65:231–243
115. Ponce YM, Pérez MAC, Zaldivar VR, Sanz MB, Mota DS, Torrens F (2005) Prediction of intestinal epithelial transport of drug in (Caco-2) cell culture from molecular structure using in silico approaches during early drug discovery. Internet Electron J Mol Des 4:124–150
116. Linnankoski J, Makela JM, Ranta VP, Urtti A, Yliperttula M (2006) Computational prediction of oral drug absorption based on absorption rate constants in humans. J Med Chem 49:3674–3681
117. Iyer M, Tseng YJ, Senese CL, Liu J, Hopfinger AJ (2007) Prediction and mechanistic interpretation of human oral drug absorption using MI-QSAR analysis. Mol Pharmaceutics 4:218–231
118. Ivanciuc O (2006) Artificial immune system prediction of the human intestinal absorption of drugs with AIRS (artificial immune recognition system). Internet Electron J Mol Des 5:515–529
119. Ivanciuc O (2007) Feature Selection in AIRS (Artificial Immune Recognition System) Structure-Activity Relationships. Internet Electron J Mol Des 6:331–344
120. Crivori P, Reinach B, Pezzetta D, Poggesi I (2006) Computational models for identifying potential P-glycoprotein substrates and inhibitors. Mol Pharmaceutics 3:33–44

121. Kaiser D, Terfloth L, Kopp S, Schulz J, de Laet R, Chiba P, Ecker GF, Gasteiger J (2007) Self-organizing maps for identification of new inhibitors of P-glycoprotein. J Med Chem 50:1698–1702
122. Ivanciuc O (2006) Artificial immune systems in drug design: Recognition of P-glycoprotein substrates with AIRS (artificial immune recognition system). Internet Electron J Mol Des 5:542–554
123. Ivanciuc O (2006) Structure-activity relationships with artificial immune systems: Classification of benzodiazepine receptor ligands with AIRS, CLONALG, CSCA, and IMMUNOS. Internet Electron J Mol Des 5:000–000 TS6
124. Verhaar HJM, Solbé J, Speksnijder J, van Leeuwen CJ, Hermens JLM (2000) Classifying environmental pollutants: Part 3. External validation of the classification system. Chemosphere 40:875–883
125. Ivanciuc O (2003) Aquatic toxicity prediction for polar and nonpolar narcotic pollutants with support vector machines. Internet Electron J Mol Des 2:195–208
126. Ivanciuc O (2007) Artificial immune systems in aquatic toxicology: Structure-activity relationships for the mechanism of toxic action with AIRS (artificial immune recognition system). Internet Electron J Mol Des 6:13–28
127. Ivanciuc O (2007) Drug design with artificial immune systems: Structure-activity relationships for glycogen phosphorylase B inhibitors with CLONALG (clonal selection algorithm). Internet Electron J Mol Des 6:311–319
128. Ivanciuc O (2007) Structure-activity relationships in aquatic toxicology with artificial immune systems: Mechanism of toxic action classification of polar and nonpolar narcotic pollutants with CLONALG (clonal selection algorithm). Internet Electron J Mol Des 6:106–114
129. Ivanciuc O (2007) Artificial immune systems structure-activity relationships for estrogen receptor ligands with CSCA (clonal selection classification system). Internet Electron J Mol Des 6:81–89
130. Ivanciuc O (2007) Artificial immune systems in the virtual screening of dihydrofolate reductase inhibitors with CSCA (clonal selection classification system). Internet Electron J Mol Des 6:253–261
131. Ivanciuc O (2007) Drug design with artificial immune systems: Classification of angiotensin converting enzyme inhibitors with CSCA (clonal selection classification system). Internet Electron J Mol Des 6:135–143
132. Ivanciuc O (2007) Artificial immune systems in structure-activity relationships: Classification of thermolysin inhibitors with CSCA (clonal selection classification system). Internet Electron J Mol Des 6:209–217
133. Ivanciuc O (2007) Structure-activity relationships for acetylcholinesterase inhibitors with the IMMUNOS artificial immune system. Internet Electron J Mol Des 6:167–175
134. Ivanciuc O (2007) Virtual screening of cyclooxygenase-2 inhibitors with the IMMUNOS artificial immune system. Internet Electron J Mol Des 6:200–208
135. Ivanciuc O (2007) Structure-activity relationships with the IMMUNOS artificial immune system for thrombin inhibitors. Internet Electron J Mol Des 6:262–270

TS6 Please give page numbers, if possible.

Editor's or typesetter's annotations (will be removed before the final TEX run)