CODEN IEJMAT *Internet Electronic Journal of Molecular Design* **2006**, *5*, 488–502 ISSN 1538–6414
BioChem Press http://www.biochempress.com

Inter*net* Electronic Journal of Molecular Design

September 2006, Volume 5, Number 9, Pages 488–502

Editor: Ovidiu Ivanciuc

Artificial Immune System Classification of Drug–induced Torsade de Pointes with AIRS (Artificial Immune Recognition System)

Ovidiu Ivanciuc [1]

[1] Department of Biochemistry and Molecular Biology, University of Texas Medical Branch, 301 University Boulevard, Galveston, Texas 77555–0857

Received: June 25, 2006; Revised: August 17, 2006; Accepted: September 3, 2006; Published: December 8, 2006

Citation of the article:
O. Ivanciuc, Artificial Immune System Classification of Drug–induced Torsade de Pointes with AIRS (Artificial Immune Recognition System), *Internet Electron. J. Mol. Des.* **2006**, *5*, 488–502, http://www.biochempress.com.

Inter*net* Electronic Journal of Molecular Design

BioChem Press
http://www.biochempress.com

Artificial Immune System Classification of Drug–induced Torsade de Pointes with AIRS (Artificial Immune Recognition System)

Ovidiu Ivanciuc [1,*]

[1] Department of Biochemistry and Molecular Biology, University of Texas Medical Branch, 301 University Boulevard, Galveston, Texas 77555–0857

Received: June 25, 2006; Revised: August 17, 2006; Accepted: September 3, 2006; Published: December 8, 2006

Internet Electron. J. Mol. Des. **2006**, *5* (9), 488–502

Abstract

Artificial immune systems (AIS) represent a family of machine learning algorithms that use immune system components and mechanisms as templates in modeling information processes, such as pattern recognition and classification. This paper demonstrates the first application of the artificial immune recognition system (AIRS) algorithm in modeling structure–activity relationships (SAR). A dataset of 349 drugs was used in the evaluation of the AIRS algorithm. The learning task was to classify these chemicals into a subset of 106 drugs that induce torsade de pointes (TdP) and a subset of 243 drugs that do not induce TdP. The chemical structure was described with five linear solvation energy relationships descriptors, namely the excess molar refraction, the combined dipolarity/polarizability, the overall solute hydrogen bond acidity, the overall solute hydrogen bond basicity, and the McGowan's characteristic volume. The classification performance of the AIRS algorithm depends on a large number of 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. The cross–validation predictions were investigated over of a wide range of values for these eight AIRS parameters. The best leave–10%–out cross–validation predictions of the AIRS algorithm (selectivity 0.783, specificity 0.893, accuracy 0.860, and Matthews correlation coefficient 0.671) surpass those obtained with 11 other machine learning algorithms, namely logistic regression, Bayesian network, naïve Bayesian classifier, alternating decision tree, C4.5 decision tree, logistic model trees, decision tree with naïve Bayesian classifiers at the leaves, fast decision tree learner, random trees, random forests, and K* instance–based classifier. The results obtained suggest that classifiers based on artificial immune systems may be successful in structure–activity relationships, drug design, and virtual screening of chemical libraries.

Keywords. Artificial immune system; AIS; artificial immune recognition system; AIRS; torsade de pointes; TdP; quantitative structure–activity relationships; QSAR.

Abbreviations and notations

AIRS, artificial immune recognition system	IMPS, initial memory cell pool size
ATS, affinity threshold scalar	NIAT, number of instances to compute the affinity threshold
CR, clonal rate	ST, stimulation threshold
HR, hypermutation rate	TR, total resources
kNN, number of nearest neighbors	TdP, torsade de pointes

* Correspondence author; E–mail: ivanciuc@gmail.com.

1 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, and particle swarm optimization. Artificial immune systems (AIS) are computational tools inspired by the processes from the biological immune system [1–6]. AIS use the learning and memory capabilities of the immune system to develop computational algorithms for pattern recognition, function optimization, classification, process control, and intrusion detection. The major AIS algorithms and the most important applications are presented in numerous books and conference proceedings: *Artificial Immune Systems and Their Applications* edited by Dasgupta [7]; *Artificial Immune Systems: A New Computational Intelligence Approach* by de Castro and Timmis [8]; *Immunocomputing: Principles and Applications*, by Tarakanov, Skormin, and Sokolova [9]; *Immunity–Based Systems* by Ishida [10]; *Artificial Immune Systems: ICARIS 2004* edited by Nicosia, Cutello, Bentley, and Timmis [11]; *Artificial Immune Systems: ICARIS 2005* edited by Jacob, Pilat, Bentley, and Timmis [12]. AIS models were successfully applied to biological and medical problems, such as classification of gene expression data [13–15], breast cancer identification [16,17], classification of liver disorders [16], detection of heart diseases [18], and diagnosis of thyroid diseases [19].

Watkins, Timmis, and Boggess developed an efficient machine learning algorithm, the artificial immune recognition system (AIRS), which encodes several principles and mechanisms of the immune system [20–22]. Brownlee used AIRS for a wide range of classification problems [23], confirming its utility as a supervised learning classifier.

In this study we demonstrate the first application of the AIRS algorithm in modeling structure–activity relationships for drug design. The learning task investigated here is the classification of chemical compounds into drugs that induce torsade de pointes (TdP+) and drugs that do not induce torsade de pointes (TdP–). Torsade de pointes (TdP) is a polymorphic ventricular arrhythmia that may be caused by drugs that induce the prolongation of the QT interval [24–26]. QT prolongation and TdP may be caused by a large number of drugs, such as antiarrhytmics, antihistamines, antimicrobials, antidepressants, and antipsychotics [27–30]. The human ether–à–go–go related gene (hERG) encodes the primary component of the K^+ channel that is responsible for the repolarization of the ventricules [30,31]. Mutations in the hERG K^+ channel gene may increase the binding affinity for certain chemical compounds that block the channel and induce lethal arrhythmias [30–33]. The drug design and development costs may be significantly reduced if, along with other ADME/Tox filters, chemical compounds that have the potential to bind and inhibit the hERG K^+ channel are eliminated as early as possible. The experimental determination of the hERG K^+ channel inhibition by a certain chemical compound, which is performed with the voltage clamp

technique, is time–consuming and expensive. To accelerate the drug development process, the inhibition of the hERG K^+ channel is predicted with various quantitative structure–activity relationships (QSAR) [34–39]. In this study we investigated a dataset of 349 drugs [40] and the learning task was to classify these chemicals into a subset of 106 drugs that induce torsade de pointes and a subset of 243 drugs that do not induce torsade de pointes.

2 THE ARTIFICIAL IMMUNE RECOGNITION SYSTEM

The artificial immune system characteristics that are relevant to AIRS are briefly reviewed below [2,20,21]. 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 recognition mechanism encoded into an antibody may be improved upon the presentation of several antigens with similar characteristics. In the AIRS classification algorithm, 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 stimulation value measures the similarity between an ARB and an antigen. Each AIRS model has a limited number of resources, and ARBs compete for their allocation. Resources are removed 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).

Detailed descriptions of the artificial immune recognition system may be found in the literature [20–23]. We present here only the most important characteristics of the AIRS procedure, in order to highlight the parameters that control its classification ability. The AIRS algorithm consists of the following steps:

(1) Initialization
(2) Train for all Antigens
  (2.1) Antigen Training
  (2.2) Competition for Limited Resources
  (2.3) Memory Cell Selection
(3) Classification

The most important step is represented by the ARB competition for limited resources, which is an iterative process:

(2.2) Competition for Limited Resources
(2.2.1) Perform Competition for Resources
(2.2.1.1) Stimulate the ARB Pool with Antigen
(2.2.1.2) Normalize the ARB Stimulation Values
(2.2.1.3) Allocate Limited Resources Based on Stimulation
(2.2.1.4) Remove ARBs with Insufficient Resources
(2.2.2) Continue with (2.3) if the Stop Condition is Satisfied
(2.2.3) Generate Mutated Clones of Surviving ARBs
(2.2.4) Go to (2.2.1)

The steps of the AIRS algorithm are briefly described below:

(1) Initialization. The training data are normalized between 0 and 1. The Euclidean distance is computed for all pairs of antigens, and then the affinity is determined as the ratio between the distance and the maximum distance. The affinity threshold 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

(2.1) Antigen Training. Each training antigen is exposed to the memory cell pool, and each memory cell receives a stimulation value, stimulation = 1 – affinity. The memory cells with the highest stimulation are selected, and a number of mutated clones are created and added to the ARB pool. The number of clones generated is computed with the formula:

$$\text{NumberClones} = \text{Stimulation} \times \text{ClonalRate} \times \text{HypermutationRate} \quad (1)$$

(2.2) Competition for Limited Resources. The scope of this process is to select those ARBs that have the best recognition capabilities, while optimally allocating the resources to the best ARBs. The number of clones generated in the step (2.2.3) is:

$$\text{NumberClones} = \text{Stimulation} \times \text{ClonalRate} \quad (2)$$

The amount of resources allocated to each ARB in the step (2.2.1.3) is:

$$\text{Resources} = \text{NormalizedStimulation} \times \text{ClonalRate} \quad (3)$$

The total amount of resources is a user defined parameter. ARBs without resources are removed from the memory cell pool. The stop condition for the ARB refinement is met when the average normalized stimulation is higher than a user defined stimulation threshold.

(2.3) Memory Cell Selection. In this step, new ARB classifiers are evaluated for inclusion in the

memory cell pool. An ARB is inserted in the memory cell pool if its stimulation value is better 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 CutOff value:

$$\text{CutOff} = \text{AffinityThreshold} \times \text{AffinityThresholdScalar} \quad (4)$$

where the AffinityThreshold was computed during the Initialization phase, and the AffinityThresholdScalar is a user defined parameter.

(3) Classification. 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.

3 MATERIALS AND METHODS

The learning task investigated here is the classification of chemical compounds into drugs that induce torsade de pointes and drugs that do not induce torsade de pointes. The dataset was collected from the literature [40], and consists of 106 TdP+ drugs and 243 TdP– drugs. The chemical structure was described with five linear solvation energy relationships (LSER) descriptors [41–43], namely the overall solute hydrogen bond acidity *A*, the overall solute hydrogen bond basicity *B*, the combined dipolarity/polarizability *S*, the excess molar refraction *E*, and the McGowan's characteristic volume *V*. All computations were performed with the AIRS2 implementation of Brownlee [23] using Weka 3.5.4 (http://sourceforge.net/projects/weka).

4 RESULTS AND DISCUSSION

We investigated the classification performance of AIRS2 over a large range of the eight user defined parameters, namely 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. The classification prediction was evaluated with leave–10%–out cross–validation. The statistical indices reported for each AIRS model are: TP_c, true positive in calibration (number of Td+ drugs classified as Td+); FN_c, false negative in calibration (number of Td+ drugs classified as Td–); TN_c, true negative in calibration (number of Td– drugs classified as Td–); FP_c, false positive in calibration (number of Td– drugs classified as Td+); Se_c, calibration selectivity; Sp_c, calibration specificity; Ac_c, calibration accuracy; MCC_c, calibration Matthews correlation coefficient; TP_p, true positive in prediction; FN_p, false negative in prediction; TN_p, true negative in prediction; FP_p, false positive in prediction; Se_p, prediction selectivity; Sp_p, prediction specificity; Ac_p, prediction accuracy; MCC_p, prediction Matthews correlation coefficient.

Affinity Threshold Scalar (ATS). ATS takes values between 0 and 1, and it is used in Eq. (4) to compute a cut–off value for memory cell replacement. If the affinity between a candidate ARB and the best matching memory cell is lower that the threshold computed with Eq. (4), then the ARB replaces the memory cell. A low value for ATS results in a low replacement rate, whereas a high ATS value increases the replacement rate. In experiments **1**–**14** (Table 1) we varied the ATS value between 0.01 and 0.9 in order to identify the optimum replacement regimen. The initial values for the remaining parameters are: clonal rate = 10, hypermutation rate = 2, number of nearest neighbors = 3, initial memory cell pool size = 50, number of instances to compute the affinity threshold = all, stimulation threshold = 0.5, and total resources = 150. These parameters are optimized in the above order, and the optimum value is used in all subsequent experiments. The highest prediction MCC = 0.6323 is obtained for ATS = 0.05, indicating that for this classification problem a low memory cell replacement rate is beneficial.

Table 1. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of ATS (Affinity Threshold Scalar)

Exp	ATS	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
1	0.01	91	15	220	23	0.8585	0.9053	0.8911	0.7490
2	0.04	84	22	226	17	0.7925	0.9300	0.8883	0.7327
3	0.05	87	19	223	20	0.8208	0.9177	0.8883	0.7365
4	0.06	87	19	217	26	0.8208	0.8930	0.8711	0.7015
5	0.07	85	21	216	27	0.8019	0.8889	0.8625	0.6805
6	0.10	83	23	219	24	0.7830	0.9012	0.8653	0.6825
7	0.20	72	34	217	26	0.6792	0.8930	0.8281	0.5856
8	0.30	68	38	213	30	0.6415	0.8765	0.8052	0.5301
9	0.40	53	53	214	29	0.5000	0.8807	0.7650	0.4129
10	0.50	59	47	203	40	0.5566	0.8354	0.7507	0.3999
11	0.60	59	47	204	39	0.5566	0.8395	0.7536	0.4053
12	0.70	60	46	213	30	0.5660	0.8765	0.7822	0.4652
13	0.80	53	53	204	39	0.5000	0.8395	0.7364	0.3544
14	0.90	65	41	200	43	0.6132	0.8230	0.7593	0.4340

Exp	ATS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
1	0.01	76	30	213	30	0.7170	0.8765	0.8281	0.5935
2	0.04	78	28	210	33	0.7358	0.8642	0.8252	0.5925
3	0.05	78	28	217	26	0.7358	0.8930	0.8453	0.6323
4	0.06	76	30	210	33	0.7170	0.8642	0.8195	0.5767
5	0.07	76	30	207	36	0.7170	0.8519	0.8109	0.5603
6	0.10	78	28	206	37	0.7358	0.8477	0.8138	0.5710
7	0.20	65	41	213	30	0.6132	0.8765	0.7966	0.5060
8	0.30	71	35	210	33	0.6698	0.8642	0.8052	0.5369
9	0.40	61	45	207	36	0.5755	0.8519	0.7679	0.4387
10	0.50	63	43	203	40	0.5943	0.8354	0.7622	0.4333
11	0.60	59	47	194	49	0.5566	0.7984	0.7249	0.3531
12	0.70	55	51	198	45	0.5189	0.8148	0.7249	0.3394
13	0.80	52	54	200	43	0.4906	0.8230	0.7221	0.3240
14	0.90	49	57	198	45	0.4623	0.8148	0.7077	0.2872

[a] Notations: Exp, experiment number; TP_c, true positive in calibration; FN_c, false negative in calibration; TN_c, true negative in calibration; FP_c, false positive in calibration; Se_c, calibration selectivity; Sp_c, calibration specificity; Ac_c, calibration accuracy; MCC_c, calibration Matthews correlation coefficient; TP_p, true positive in prediction; FN_p, false negative in prediction; TN_p, true negative in prediction; FP_p, false positive in prediction; Se_p, prediction selectivity; Sp_p, prediction specificity; Ac_p, prediction accuracy; MCC_p, prediction Matthews correlation coefficient.

Clonal Rate (CR). CR takes integer values, and is used in ARB resource allocation and in controlling the clonal mutation for the memory cell population. In Eq (1), CR is used to determine the number of mutated clones generated from each memory cell and then added to the ARB pool. In Eq. (2), CR is involved in the computation of the number of clones generated from each ARB during the ARB refinement process. Therefore, the number of ARB clones generated is in the range [0, CR]. In Eq. (3), CR is multiplied with the normalized stimulation of an ARB to determine the number of resources allocated to that ARB. The number of resources allocated to each ARB is in the range [0, CR].

The clonal rate was varied between 3 and 17, as shown in experiments **15**–**23** (Table 2). A general trend for the prediction MCC is to increase from CR = 3 up to CR = 10, and then to decrease when CR increases up to 17. These results suggest that for CR = 10 the AIRS generates the optimum number of clones and allocates the optimum number of resources.

Table 2. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of CR (Clonal Rate); (ATS = 0.05)

Exp	CR	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
15	3	75	31	216	27	0.7075	0.8889	0.8338	0.6031
16	5	81	25	215	28	0.7642	0.8848	0.8481	0.6439
17	8	87	19	223	20	0.8208	0.9177	0.8883	0.7365
18	9	86	20	221	22	0.8113	0.9095	0.8797	0.7170
19	10	87	19	223	20	0.8208	0.9177	0.8883	0.7365
20	11	81	25	222	21	0.7642	0.9136	0.8682	0.6853
21	12	91	15	225	18	0.8585	0.9259	0.9054	0.7784
22	15	87	19	220	23	0.8208	0.9053	0.8797	0.7187
23	17	83	23	217	26	0.7830	0.8930	0.8596	0.6708

Exp	CR	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
15	3	68	38	211	32	0.6415	0.8683	0.7994	0.5185
16	5	71	35	210	33	0.6698	0.8642	0.8052	0.5369
17	8	77	29	218	25	0.7264	0.8971	0.8453	0.6305
18	9	76	30	212	31	0.7170	0.8724	0.8252	0.5879
19	10	78	28	217	26	0.7358	0.8930	0.8453	0.6323
20	11	76	30	216	27	0.7170	0.8889	0.8367	0.6109
21	12	77	29	210	33	0.7264	0.8642	0.8223	0.5846
22	15	76	30	213	30	0.7170	0.8765	0.8281	0.5935
23	17	76	30	210	33	0.7170	0.8642	0.8195	0.5767

Hypermutation Rate (HR). The hypermutation rate takes integer values and is used in Eq. (1) to determine the number of clones for each memory cell, which is in the range [0, CR×HR]. We investigated the TdP classification for values of the hypermutation rate between 1 and 10, as shown in experiments **24**–**33** (Table 3). The best predictions are obtained with HR = 2, with the prediction MCC = 0.6323, whereas for other HR values the prediction statistics are slightly lower. The same HR value was used in the previous experiments, which explains the fact that the predictions are not improved in this set of experiments.

Table 3. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of HR (Hypermutation Rate); (CR = 10)

Exp	HR	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
24	1	83	23	222	21	0.7830	0.9136	0.8739	0.7004
25	2	87	19	223	20	0.8208	0.9177	0.8883	0.7365
26	3	93	13	222	21	0.8774	0.9136	0.9026	0.7756
27	4	87	19	223	20	0.8208	0.9177	0.8883	0.7365
28	5	84	22	228	15	0.7925	0.9383	0.8940	0.7455
29	6	91	15	218	25	0.8585	0.8971	0.8854	0.7376
30	7	88	18	221	22	0.8302	0.9095	0.8854	0.7321
31	8	88	18	219	24	0.8302	0.9012	0.8797	0.7205
32	9	90	16	222	21	0.8491	0.9136	0.8940	0.7531
33	10	88	18	215	28	0.8302	0.8848	0.8682	0.6980

Exp	HR	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
24	1	74	32	215	28	0.6981	0.8848	0.8281	0.5894
25	2	78	28	217	26	0.7358	0.8930	0.8453	0.6323
26	3	77	29	204	39	0.7264	0.8395	0.8052	0.5525
27	4	76	30	215	28	0.7170	0.8848	0.8338	0.6050
28	5	76	30	218	25	0.7170	0.8971	0.8424	0.6227
29	6	79	27	214	29	0.7453	0.8807	0.8395	0.6227
30	7	72	34	218	25	0.6792	0.8971	0.8309	0.5917
31	8	79	27	212	31	0.7453	0.8724	0.8338	0.6114
32	9	75	31	213	30	0.7075	0.8765	0.8252	0.5857
33	10	77	29	212	31	0.7264	0.8724	0.8281	0.5957

Number of Nearest Neighbors (kNN). The number *k* of nearest neighbors is used in the classification process, in which the *k* most stimulated memory cells to a given antigen vote for the class (TdP+ or TdP–) of that antigen. The results obtained in the experiments **34**–**43** (Table 4) show identical prediction accuracy for $k = 3$ and $k = 5$. We selected $k = 3$ for further experiments because it is faster to compute.

Table 4. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of kNN (Number of Nearest Neighbors); (HR = 2)

Exp	kNN	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
34	1	90	16	228	15	0.8491	0.9383	0.9112	0.7894
35	3	87	19	223	20	0.8208	0.9177	0.8883	0.7365
36	5	87	19	221	22	0.8208	0.9095	0.8825	0.7246
37	7	78	28	220	23	0.7358	0.9053	0.8539	0.6502
38	9	68	38	223	20	0.6415	0.9177	0.8338	0.5922
39	11	60	46	224	19	0.5660	0.9218	0.8138	0.5361
40	13	58	48	230	13	0.5472	0.9465	0.8252	0.5640
41	15	56	50	231	12	0.5283	0.9506	0.8223	0.5560
42	17	55	51	233	10	0.5189	0.9588	0.8252	0.5643
43	19	49	57	230	13	0.4623	0.9465	0.7994	0.4918

Exp	kNN	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
34	1	78	28	215	28	0.7358	0.8848	0.8395	0.6206
35	3	78	28	217	26	0.7358	0.8930	0.8453	0.6323
36	5	79	27	216	27	0.7453	0.8889	0.8453	0.6342
37	7	75	31	215	28	0.7075	0.8848	0.8309	0.5972
38	9	72	34	217	26	0.6792	0.8930	0.8281	0.5856
39	11	67	39	217	26	0.6321	0.8930	0.8138	0.5462
40	13	65	41	214	29	0.6132	0.8807	0.7994	0.5120
41	15	62	44	219	24	0.5849	0.9012	0.8052	0.5188
42	17	58	48	216	27	0.5472	0.8889	0.7851	0.4672
43	19	55	51	215	28	0.5189	0.8848	0.7736	0.4360

Initial Memory Cell Pool Size (IMCPS). The number of initial memory cells was modified from 1 to 220 (experiments **44**–**59**, Table 5), and the classification results show only small variations, with better results for AIRS models that have IMCPS > 30. Compared with previous experiments, a minor prediction improvement is obtained for IMCPS = 80, with a prediction MCC = 0.6362. This IMCPS value was adopted for further experiments.

Table 5. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of IMCPS (Initial Memory Cell Pool Size); (kNN = 3)

Exp	IMCPS	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
44	1	89	17	217	26	0.8396	0.8930	0.8768	0.7168
45	10	84	22	225	18	0.7925	0.9259	0.8854	0.7264
46	20	91	15	221	22	0.8585	0.9095	0.8940	0.7547
47	30	83	23	227	16	0.7830	0.9342	0.8883	0.7316
48	40	82	24	218	25	0.7736	0.8971	0.8596	0.6689
49	50	87	19	223	20	0.8208	0.9177	0.8883	0.7365
50	60	91	15	217	26	0.8585	0.8930	0.8825	0.7321
51	70	93	13	215	28	0.8774	0.8848	0.8825	0.7364
52	80	91	15	216	27	0.8585	0.8889	0.8797	0.7265
53	100	89	17	225	18	0.8396	0.9259	0.8997	0.7635
54	120	89	17	226	17	0.8396	0.9300	0.9026	0.7697
55	140	91	15	225	18	0.8585	0.9259	0.9054	0.7784
56	160	90	16	224	19	0.8491	0.9218	0.8997	0.7649
57	180	83	23	231	12	0.7830	0.9506	0.8997	0.7580
58	200	81	25	232	11	0.7642	0.9547	0.8968	0.7503
59	220	82	24	233	10	0.7736	0.9588	0.9026	0.7645

Exp	IMCPS	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
44	1	66	40	205	38	0.6226	0.8436	0.7765	0.4688
45	10	71	35	210	33	0.6698	0.8642	0.8052	0.5369
46	20	67	39	211	32	0.6321	0.8683	0.7966	0.5105
47	30	68	38	204	39	0.6415	0.8395	0.7794	0.4798
48	40	73	33	212	31	0.6887	0.8724	0.8166	0.5642
49	50	78	28	217	26	0.7358	0.8930	0.8453	0.6323
50	60	71	35	217	26	0.6698	0.8930	0.8252	0.5777
51	70	75	31	206	37	0.7075	0.8477	0.8052	0.5470
52	80	80	26	215	28	0.7547	0.8848	0.8453	0.6362
53	100	72	34	206	37	0.6792	0.8477	0.7966	0.5229
54	120	76	30	205	38	0.7170	0.8436	0.8052	0.5497
55	140	76	30	213	30	0.7170	0.8765	0.8281	0.5935
56	160	74	32	215	28	0.6981	0.8848	0.8281	0.5894
57	180	76	30	201	42	0.7170	0.8272	0.7937	0.5290
58	200	75	31	208	35	0.7075	0.8560	0.8109	0.5578
59	220	76	30	211	32	0.7170	0.8683	0.8223	0.5823

Number of Instances to Compute the Affinity Threshold (NIAT). During the AIRS initialization process, the affinity threshold is computed as the average affinity for NIAT antigens from the training set. In experiments **60**–**71** (Table 6) we tried to identify an optimum value for NIAT (in previous experiments the entire training set was used to compute the affinity threshold). An improvement is obtained for NIAT = 100, with a prediction MCC = 0.6708, whereas NIAT values between 25 and the entire dataset all give good predictions.

Table 6. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of NIAT (Number of Instances to Compute the Affinity Threshold); (IMCPS = 80)

Exp	NIAT	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
60	25	94	12	219	24	0.8868	0.9012	0.8968	0.7660
61	50	90	16	217	26	0.8491	0.8930	0.8797	0.7244
62	75	90	16	217	26	0.8491	0.8930	0.8797	0.7244
63	100	95	11	225	18	0.8962	0.9259	0.9169	0.8080
64	125	96	10	223	20	0.9057	0.9177	0.9140	0.8038
65	150	92	14	224	19	0.8679	0.9218	0.9054	0.7798
66	175	91	15	228	15	0.8585	0.9383	0.9140	0.7968
67	200	93	13	222	21	0.8774	0.9136	0.9026	0.7756
68	225	92	14	225	18	0.8679	0.9259	0.9083	0.7858
69	250	92	14	225	18	0.8679	0.9259	0.9083	0.7858
70	275	92	14	225	18	0.8679	0.9259	0.9083	0.7858
71	all	91	15	216	27	0.8585	0.8889	0.8797	0.7265

Exp	NIAT	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
60	25	83	23	213	30	0.7830	0.8765	0.8481	0.6482
61	50	81	25	213	30	0.7642	0.8765	0.8424	0.6326
62	75	78	28	213	30	0.7358	0.8765	0.8338	0.6092
63	100	83	23	217	26	0.7830	0.8930	0.8596	0.6708
64	125	85	21	209	34	0.8019	0.8601	0.8424	0.6422
65	150	84	22	210	33	0.7925	0.8642	0.8424	0.6397
66	175	80	26	207	36	0.7547	0.8519	0.8223	0.5921
67	200	82	24	209	34	0.7736	0.8601	0.8338	0.6186
68	225	83	23	206	37	0.7830	0.8477	0.8281	0.6107
69	250	78	28	209	34	0.7358	0.8601	0.8223	0.5870
70	275	83	23	214	29	0.7830	0.8807	0.8510	0.6538
71	all	80	26	215	28	0.7547	0.8848	0.8453	0.6362

Stimulation Threshold (ST). The stimulation threshold is a parameter in the range [0, 1] and is used to determine the stop condition for the process of refining the ARB pool for a specific antigen. The ARB refinement stops when the average normalized ARB stimulation is higher than ST. The stimulation threshold was modified from 0.1 to 0.9 (experiments **72**–**86**, Table 7), and the best predictions were obtained for ST = 0.5. The same ST value was used in all previous experiments, and thus no improvement is obtained for the prediction statistics.

Table 7. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of ST (Stimulation Threshold); (NIAT = 100)

Exp	ST	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
72	0.10	92	14	218	25	0.8679	0.8971	0.8883	0.7453
73	0.20	92	14	218	25	0.8679	0.8971	0.8883	0.7453
74	0.30	91	15	212	31	0.8585	0.8724	0.8682	0.7049
75	0.40	92	14	215	28	0.8679	0.8848	0.8797	0.7287
76	0.45	91	15	216	27	0.8585	0.8889	0.8797	0.7265
77	0.47	98	8	216	27	0.9245	0.8889	0.8997	0.7802
78	0.49	92	14	219	24	0.8679	0.9012	0.8911	0.7509
79	0.50	95	11	225	18	0.8962	0.9259	0.9169	0.8080
80	0.51	88	18	217	26	0.8302	0.8930	0.8739	0.7091
81	0.53	93	13	223	20	0.8774	0.9177	0.9054	0.7814
82	0.55	87	19	226	17	0.8208	0.9300	0.8968	0.7549
83	0.60	90	16	223	20	0.8491	0.9177	0.8968	0.7590
84	0.70	90	16	227	16	0.8491	0.9342	0.9083	0.7832
85	0.80	90	16	229	14	0.8491	0.9424	0.9140	0.7957
86	0.90	93	13	226	17	0.8774	0.9300	0.9140	0.7992

Table 7. (Continued)

Exp	ST	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
72	0.10	73	33	205	38	0.6887	0.8436	0.7966	0.5256
73	0.20	73	33	205	38	0.6887	0.8436	0.7966	0.5256
74	0.30	75	31	203	40	0.7075	0.8354	0.7966	0.5312
75	0.40	76	30	202	41	0.7170	0.8313	0.7966	0.5341
76	0.45	72	34	210	33	0.6792	0.8642	0.8080	0.5449
77	0.47	77	29	208	35	0.7264	0.8560	0.8166	0.5737
78	0.49	83	23	206	37	0.7830	0.8477	0.8281	0.6107
79	0.50	83	23	217	26	0.7830	0.8930	0.8596	0.6708
80	0.51	83	23	216	27	0.7830	0.8889	0.8567	0.6651
81	0.53	76	30	214	29	0.7170	0.8807	0.8309	0.5992
82	0.55	80	26	205	38	0.7547	0.8436	0.8166	0.5816
83	0.60	82	24	209	34	0.7736	0.8601	0.8338	0.6186
84	0.70	73	33	213	30	0.6887	0.8765	0.8195	0.5699
85	0.80	77	29	210	33	0.7264	0.8642	0.8223	0.5846
86	0.90	73	33	214	29	0.6887	0.8807	0.8223	0.5757

Total Resources (TR). The number of total resources limits the number of ARBs from the ARB pool. The amount of resources assigned to an ARB is calculated with Eq. (3) as a number in the range [0, CR]. Resources are allocated to the ARBs with high stimulation values, and taken from those with small stimulation values. ARBs without resources are removed from the cell population.

In experiments **87**–**92** (Table 8) the total amount of resources was increased from 25 to 150, showing a steady increase of the prediction MCC. The best prediction MCC is obtained for TR = 125 and TR = 150, but with no improvement over the previous group of experiments.

Table 8. Calibration and Prediction Statistics of AIRS Models Computed for Various Values of TR (Total Resources); (ST = 0.5)

Exp	TR	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
87	25	92	14	218	25	0.8679	0.8971	0.8883	0.7453
88	50	88	18	210	33	0.8302	0.8642	0.8539	0.6710
89	75	89	17	221	22	0.8396	0.9095	0.8883	0.7397
90	100	92	14	222	21	0.8679	0.9136	0.8997	0.7681
91	125	95	11	225	18	0.8962	0.9259	0.9169	0.8080
92	150	95	11	225	18	0.8962	0.9259	0.9169	0.8080

Exp	TR	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
87	25	73	33	205	38	0.6887	0.8436	0.7966	0.5256
88	50	71	35	209	34	0.6698	0.8601	0.8023	0.5313
89	75	73	33	208	35	0.6887	0.8560	0.8052	0.5418
90	100	72	34	215	28	0.6792	0.8848	0.8223	0.5737
91	125	83	23	217	26	0.7830	0.8930	0.8596	0.6708
92	150	83	23	217	26	0.7830	0.8930	0.8596	0.6708

After performing a large number of experiments, we can conclude that the AIRS procedure is stable, and offers good predictions for a wide range of the user defined parameters. The largest variation of the prediction MCC was observed in the optimization of the affinity threshold scalar ATS. For the remaining groups of experiments, MCC_p improved slightly or at all, because the default values for these parameters were (near) optimal.

Comparison with other Machine Learning Algorithms. The same TdP+/TdP– classification problem was solved with 11 other machine learning algorithms namely logistic regression LogisticReg, Bayesian network BayesNet, naïve Bayesian classifier NaiveBayes, alternating decision tree ADTree, C4.5 decision tree J48, logistic model trees LMT, decision tree with naïve Bayesian classifiers at the leaves NBTree, fast decision tree learner REPTree, random trees RandomTree, random forests RandomForest, and K* instance–based classifier KStar. All calculations were performed with Weka.

Table 9. Calibration and Prediction Statistics of Several Machine Learning Models

Model	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
LogisticReg	42	64	222	21	0.3962	0.9136	0.7564	0.3704
BayesNet	31	75	232	11	0.2925	0.9547	0.7536	0.3494
NaiveBayes	4	102	240	3	0.0377	0.9877	0.6991	0.0833
ADTree	87	19	203	40	0.8208	0.8354	0.8309	0.6272
J48	92	14	240	3	0.8679	0.9877	0.9513	0.8840
LMT	83	23	233	10	0.7830	0.9588	0.9054	0.7717
NBTree	77	29	232	11	0.7264	0.9547	0.8854	0.7213
REPTree	85	21	226	17	0.8019	0.9300	0.8911	0.7401
RandomTree	106	0	243	0	1.0000	1.0000	1.0000	1.0000
RandomForest	106	0	242	1	1.0000	0.9959	0.9971	0.9933
KStar	106	0	243	0	1.0000	1.0000	1.0000	1.0000

Model	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
LogisticReg	42	64	221	22	0.3962	0.9095	0.7536	0.3633
BayesNet	35	71	218	25	0.3302	0.8971	0.7249	0.2770
NaiveBayes	11	95	222	21	0.1038	0.9136	0.6676	0.0277
ADTree	52	54	215	28	0.4906	0.8848	0.7650	0.4106
J48	52	54	216	27	0.4906	0.8889	0.7679	0.4170
LMT	59	47	225	18	0.5566	0.9259	0.8138	0.5351
NBTree	52	54	213	30	0.4906	0.8765	0.7593	0.3982
REPTree	58	48	214	29	0.5472	0.8807	0.7794	0.4548
RandomTree	69	37	198	45	0.6509	0.8148	0.7650	0.4567
RandomForest	78	28	212	31	0.7358	0.8724	0.8309	0.6036
KStar	81	25	211	32	0.7642	0.8683	0.8367	0.6216

The results reported in Table 9 show that the best predictions are obtained with logistic model trees (MCC_p = 0.5351), random forests (MCC_p = 0.6036), and K* instance–based classifier (MCC_p = 0.6216). The predictions obtained with the AIRS2 algorithm (MCC_p = 0.6708) are higher than those obtained with these 11 machine learning procedures, indicating that the artificial immune recognition system is a powerful classification method, that may be applied with success in structure–activity studies.

5 CONCLUSIONS

Artificial immune systems use procedures inspired from biological immune systems for pattern recognition and classification. In this report we demonstrated the first application of the artificial immune recognition system algorithm [20–22] in modeling structure–activity relationships. The

learning task was to classify a dataset of 349 chemicals [40] into a subset of 106 drugs that induce torsade de pointes and a subset of 243 drugs that do not induce torsade de pointes. The AIRS2 procedure [23] as implemented in Weka was used for all machine learning experiments. As structural descriptors we used linear solvation energy relationships descriptors [41–43], namely the excess molar refraction, the combined dipolarity/polarizability, the overall solute hydrogen bond acidity, the overall solute hydrogen bond basicity, and the McGowan's characteristic volume [40].

The classification performance of the AIRS2 algorithm was investigated 92 experiments and for a wide range of values for the 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. The largest variation of the prediction statistics was observed for ATS, whereas small or no improvement was observed during the optimization of the remaining parameters. The best value for the affinity threshold scalar was ATS = 0.05, indicating that for the TdP drug classification problem a low memory cell replacement rate is beneficial.

The best leave–10%–out cross–validation predictions of the AIRS algorithm (selectivity 0.783, specificity 0.893, accuracy 0.860, and Matthews correlation coefficient 0.671) surpass those obtained with 11 other machine learning algorithms, namely logistic regression, Bayesian network, naïve Bayesian classifier, alternating decision tree, C4.5 decision tree, logistic model trees, decision tree with naïve Bayesian classifiers at the leaves, fast decision tree learner, random trees, random forests, and K* instance–based classifier. The results obtained suggest that classifiers based on artificial immune systems may be successful in structure–activity relationships, drug design, and virtual screening of chemical libraries.

6 REFERENCES

[1] J. E. Hunt and D. E. Cooke, Learning using an artificial immune system, *J. Netw. Comput. Appl.* **1996**, *19*, 189–212.
[2] L. N. de Castro and F. J. Von Zuben, Artificial immune systems: Part I – Basic theory and applications. FEEC/UNICAMP, Brazil, 1999.
[3] L. N. de Castro and F. J. Von Zuben, Artificial immune systems: Part II – A survey of applications. FEEC/UNICAMP, Brazil, 2000.
[4] J. Timmis, M. Neal, and J. Hunt, An artificial immune system for data analysis, *BioSystems* **2000**, *55*, 143–150.
[5] L. N. de Castro and J. I. Timmis, Artificial immune systems as a novel soft computing paradigm, *Soft Comput.* **2003**, *7*, 526–544.
[6] L. N. De Castro, Dynamics of an artificial immune network, *J. Exp. Theor. Artif. Intell.* **2004**, *16*, 19–39.
[7] D. Dasgupta (Ed.), *Artificial Immune Systems and Their Applications*, Springer–Verlag, Berlin, 1999.
[8] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer–Verlag, Berlin, 2002.
[9] A. O. Tarakanov, V. A. Skormin, and S. P. Sokolova, *Immunocomputing: Principles and Applications*, Springer–Verlag, Berlin, 2003.
[10] Y. Ishida, *Immunity–Based Systems*, Springer–Verlag, Berlin, 2004.
[11] G. Nicosia, V. Cutello, P. J. Bentley, and J. I. Timmis (Eds.), *Artificial Immune Systems: Third International Conference, ICARIS 2004, Catania, Sicily, Italy, September 13–16, 2004*, *Lecture Notes in Computer Science*, *Vol. 3239*, Springer–Verlag, Berlin, 2004.

[12] C. Jacob, M. L. Pilat, P. J. Bentley, and J. Timmis (Eds.), *Artificial Immune Systems: 4th International Conference, ICARIS 2005, Banff, Alberta, Canada, August 14–17, 2005*, *Lecture Notes in Computer Science*, *Vol. 3627*, Springer–Verlag, Berlin, 2005.
[13] S. Ando and H. Iba, Classification of gene expression profile using combinatory method of evolutionary computation and machine learning, *Genet. Programm. Evolv. Mach.* **2004**, *5*, 145–156.
[14] G. B. Bezerra, G. M. A. Cançado, M. Menossi, L. N. de Castro, and F. J. Von Zuben, Recent advances in gene expression data clustering: A case study with comparative results, *Genet. Mol. Res.* **2005**, *4*, 514–524.
[15] D. Tsankova, V. Georgieva, and N. Kasabov, Artificial immune networks as a paradigm for classification and profiling of gene expression data, *J. Comput. Theor. Nanosci.* **2005**, *2*, 543–550.
[16] K. Polat, S. Şahan, H. Kodaz, and S. Güneş, Breast cancer and liver disorders classification using artificial immune recognition system (AIRS) with performance evaluation by fuzzy resource allocation mechanism, *Expert Syst. Appl.* **2007**, *32*, 172–183.
[17] S. Şahan, K. Polat, H. Kodaz, and S. Güneş, A new hybrid method based on fuzzy–artificial immune system and *k*–nn algorithm for breast cancer diagnosis, *Comput. Biol. Med.* **2007**, *37*, 415–423.
[18] K. Polat, S. Şahan, and S. Güneş, Automatic detection of heart disease using an artificial immune recognition system (AIRS) with fuzzy resource allocation mechanism and *k*–nn (nearest neighbour) based weighting preprocessing, *Expert Syst. Appl.* **2007**, *32*, 625–631.
[19] K. Polat, S. Şahan, and S. Güneş, A novel hybrid method based on artificial immune recognition system (AIRS) with fuzzy weighted pre–processing for thyroid disease diagnosis, *Expert Syst. Appl.* **2007**, *32*, 1141–1147.
[20] A. Watkins, J. Timmis, and L. Boggess, Artificial immune recognition system (AIRS): An immune–inspired supervised learning algorithm, *Genet. Programm. Evolv. Mach.* **2004**, *5*, 291–317.
[21] A. B. Watkins, AIRS: A resource limited artificial immune classifier. Department of Computer Science, MS Thesis, Mississippi State University, 2001, pp. 81.
[22] A. B. Watkins, Exploiting immunological metaphors in the development of serial, parallel and distributed learning algorithms. PhD Thesis, University of Kent, Canterbury, UK, 2005, pp. 314.
[23] J. Brownlee, 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, Australia, 2005.
[24] Y. G. Yap and A. J. Camm, Drug induced QT prolongation and torsades de pointes, *Heart* **2003**, *89*, 1363–1372.
[25] L. Belardinelli, C. Antzelevitch, and M. A. Vos, Assessing predictors of drug–induced torsade de pointes, *Trends Pharmacol. Sci.* **2003**, *24*, 619–625.
[26] R. R. Fenichel, M. Malik, C. Antzelevitch, M. Sanguinetti, D. M. Roden, S. G. Priori, J. N. Ruskin, R. J. Lipicky, and L. R. Cantilena, Drug–induced torsades de pointes and implications for drug development, *J. Cardiovasc. Electrophysiol.* **2004**, *15*, 475–495.
[27] B. Darpö, Spectrum of drugs prolonging QT interval and the incidence of torsades de pointes, *Eur. Heart J. Suppl.* **2001**, *3*, K70–K80.
[28] W. S. Redfern, L. Carlsson, A. S. Davis, W. G. Lynch, I. MacKenzie, S. Palethorpe, P. K. S. Siegl, I. Strang, A. T. Sullivan, R. Wallis, A. J. Camm, and T. G. Hammond, Relationships between preclinical cardiac electrophysiology, clinical QT interval prolongation and torsade de pointes for a broad range of drugs: Evidence for a provisional safety margin in drug development, *Cardiovasc. Res.* **2003**, *58*, 32–45.
[29] D. M. Roden, Drug–induced prolongation of the QT interval, *N. Engl. J. Med.* **2004**, *350*, 1013–1022.
[30] M. Jalaie and D. D. Holsworth, QT interval prolongation: And the beat goes on, *Mini–Rev. Med. Chem.* **2005**, *5*, 1083–1091.
[31] J. S. Mitcheson, J. Chen, M. Lin, C. Culberson, and M. C. Sanguinetti, A structural basis for drug–induced long QT syndrome, *Proc. Natl. Acad. Sci. USA* **2000**, *97*, 12329–12333.
[32] P. Hoffmann and B. Warner, Are hERG channel inhibition and QT interval prolongation all there is in drug–induced torsadogenesis? A review of emerging trends, *J. Pharmacol. Toxicol. Methods* **2006**, *53*, 87–105.
[33] K. Kamiya, R. Niwa, J. S. Mitcheson, and M. C. Sanguinetti, Molecular determinants of hERG channel block, *Mol. Pharmacol.* **2006**, *69*, 1709–1716.
[34] S. Ekins, W. J. Crumb, R. D. Sarazan, J. H. Wikel, and S. A. Wrighton, Three–dimensional quantitative structure–activity relationship for inhibition of human ether–a–go–go–related gene potassium channel, *J. Pharmacol. Exp. Ther.* **2002**, *301*, 427–434.
[35] D. Fernandez, A. Ghanta, G. W. Kauffman, and M. C. Sanguinetti, Physicochemical features of the hERG channel drug binding site, *J. Biol. Chem.* **2004**, *279*, 10120–10127.
[36] M. C. Sanguinetti and J. S. Mitcheson, Predicting drug–hERG channel interactions that cause acquired long QT syndrome, *Trends Pharmacol. Sci.* **2005**, *26*, 119–124.
[37] O. Roche, G. Trube, J. Zuegge, P. Pflimlin, A. Alanine, and G. Schneider, A virtual screening method for prediction of the hERG potassium channel liability of compound libraries, *ChemBioChem* **2002**, *3*, 455–459.
[38] H. Sun, An accurate and interpretable Bayesian classification model for prediction of hERG liability,

ChemMedChem **2006**, *1*, 315–322.
[39] M. Seierstad and D. K. Agrafiotis, A QSAR model of hERG binding using a large, diverse, and internally consistent training set, *Chem. Biol. Drug Des.* **2006**, *67*, 284–296.
[40] C. W. Yap, C. Z. Cai, Y. Xue, and Y. Z. Chen, Prediction of torsade–causing potential of drugs by support vector machine approach, *Toxicol. Sci.* **2004**, *79*, 170–177.
[41] M. H. Abraham, Scales of solute hydrogen–bonding: Their construction and application to physicochemical and biochemical processes, *Chem. Soc. Rev.* **1993**, *22*, 73–83.
[42] M. H. Abraham, A. Ibrahim, A. M. Zissimos, Y. H. Zhao, J. Comer, and D. P. Reynolds, Application of hydrogen bonding calculations in property based drug design, *Drug Discov. Today* **2002**, *7*, 1056–1063.
[43] J. A. Platts, D. Butina, M. H. Abraham, and A. Hersey, Estimation of molecular linear free energy relation descriptors using a group contribution approach, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 835–845.