Inter*net* **Electronic** Journal of
Molecular Design

November 2006, Volume 5, Number 11, Pages 542–554

Editor: Ovidiu Ivanciuc

Special issue dedicated to Professor Lemont B. Kier on the occasion of the 75th birthday

Artificial Immune Systems in Drug Design: Recognition of P–Glycoprotein Substrates 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: July 5, 2006; Revised: October 12, 2006; Accepted: October 28, 2006; Published: December 17, 2006

Citation of the article:

O. Ivanciuc, Artificial Immune Systems in Drug Design: Recognition of P–Glycoprotein Substrates with AIRS (Artificial Immune Recognition System), *Internet Electron. J. Mol. Des.* **2006**, *5*, 542–554, http://www.biochempress.com.

Inter*net* Electronic Journal of Molecular Design

BioChem Press
http://www.biochempress.com

Artificial Immune Systems in Drug Design: Recognition of P–Glycoprotein Substrates 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: March 1, 2006; Revised: June 20, 2006; Accepted: July 1, 2006; Published: December 17, 2006

Internet Electron. J. Mol. Des. **2006**, *5* (11), 542–554

Abstract

Artificial immune systems (AIS) represent a new class of machine learning procedures that simulate several mechanisms and functions of the biological immune system, such as pattern recognition, learning, memory, and optimization. In this paper we present the first application of the artificial immune recognition system (AIRS) to the recognition of the substrates of the multidrug resistance (MDR) ATP–binding cassette (ABC) transporter permeability glycoprotein (P–glycoprotein, P–gp). We evaluated the AIRS algorithm for a dataset of 201 chemicals, consisting of 116 P–gp substrates and 85 P–gp nonsubstrates. The classifiers were computed from 159 structural descriptors from five classes, namely constitutional descriptors, topological indices, electrotopological state indices, quantum descriptors, and geometrical indices. The AIRS algorithm is controlled by 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. The AIRS sensitivity to these parameters was investigated with leave–20%–out (five–fold) cross–validation predictions performed over a wide range of values for the eight AIRS parameters. The AIRS algorithm (best predictions: selectivity 0.793, specificity 0.577, accuracy 0.702, and Matthews correlation coefficient 0.380) was compared with 13 well–established machine learning algorithms. The AIRS predictions are better than those of five of these algorithms (alternating decision tree, Bayesian network, logistic regression with ridge estimator, random tree, and fast decision tree learner), showing that P–gp substrates may be successfully recognized with AIRS. In conclusion, classifiers based on artificial immune systems are valuable tools for structure–activity relationships (SAR), quantitative structure–activity relationships (QSAR), drug design, and virtual screening of chemical libraries.

Keywords. Artificial immune system; AIS; artificial immune recognition system; AIRS; pattern recognition; machine learning; P–glycoprotein; P–gp; 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	P–gp, P–glycoprotein

# Dedicated to Professor Lemont B. Kier on the occasion of the 75th birthday.

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

1 INTRODUCTION

Biological mechanisms, processes, and functions are the source of inspiration for many artificial intelligence algorithms, such as particle swarm optimization, ant colony optimization, bee colony optimization, artificial neural networks, genetic algorithms, DNA computing, and artificial immune systems. Artificial immune systems (AIS) [1–9] use the learning and memory capabilities of the immune system to develop computational algorithms for pattern recognition, function optimization, classification, process control, intrusion detection, medical diagnosis, and drug design [10–17]. 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 [18–20]. Brownlee used AIRS for a wide range of classification problems [21], confirming its utility as a supervised learning classifier.

We recently published the first application of the AIRS algorithm in modeling structure–activity relationships for drug design [16] namely to discriminate between drugs that induce torsade de pointes and drugs that do not induce torsade de pointes. In a subsequent study we showed that AIRS is successful in separating drugs that penetrate the human intestine from those that do not penetrate the intestine [17]. In this paper we present the first application of the artificial immune recognition system (AIRS) to the recognition of the substrates of the multidrug resistance (MDR) ATP–binding cassette (ABC) transporter permeability glycoprotein (P–glycoprotein, P–gp). Using a dataset of 201 drugs and 159 structural descriptors [22], AIRS is trained to discriminate between a subset of 116 P–gp substrates and a subset of 85 P–gp nonsubstrates.

2 THE ARTIFICIAL IMMUNE RECOGNITION SYSTEM

In the AIRS classification algorithm, an antigen is represented as an *n*–dimensional vector $\mathbf{X} = \{x_1, x_2, \ldots, x_n; x_i \in R \text{ for } i = 1, 2, \ldots, n\}$ and an associated class $Y = \{+1, -1\}$. For quantitative structure–activity relationships (QSAR), the **X** vector contains the structural descriptors for a molecule, whereas for the class variable *Y*, +1 encodes the presence of a property (P–gp substrate, in the present study) and –1 encodes the absence of that property (not a P–gp substrate). An identical $\{\mathbf{X}, Y\}$ encoding is used for antibodies (the solutions for the classification problem). In the AIRS procedure a B–cell is represented by an artificial recognition ball (ARB). An ARB contains 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). The steps of the AIRS algorithm are summarized in Figure 1.

(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 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

(2.1) Antigen Presentation. Each training antigen is presented 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:

$$\text{NumberClones} = \text{Stimulation}\times\text{CR}\times\text{HR} \tag{1}$$

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

(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.

(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. The amount of resources allocated to each ARB is:

$$\text{Resources} = \text{NormalizedStimulation}\times\text{CR} \tag{2}$$

(2.2.1.4) Remove ARBs with Insufficient Resources

(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 is:

$$\text{NumberClones} = \text{Stimulation}\times\text{CR} \tag{3}$$

(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 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:

$$\text{CutOff} = \text{AT}\times\text{ATS} \tag{4}$$

where the affinity threshold AT was computed during the Initialization phase, and ATS (affinity threshold scalar) 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.

Figure 1. The AIRS algorithm.

3 MATERIALS AND METHODS

P–glycoprotein 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 P–gp substrates are useful drug design tools for the early elimination of potential P–gp substrates. Gombar *et al*. used 95 compounds and 27 structural descriptors to develop a linear discriminant model that had a prediction accuracy of 86.2% was obtained on a test set of 58 compounds [23]. Xue *et al*. developed a support vector machines (SVM) classifier for P–gp substrates [24] for a dataset of 201 molecules and 159 structural descriptors, with a leave–20%–out cross–validation accuracy of 0.683 and Matthews correlation coefficient of 0.37 [22]. The P–gp substrate models developed by de Cerqueira Lima *et al*. [25] with *k*–nearest neighbors classification, decision tree, binary QSAR, and support vector machines show that the best predictions are obtained with SVM trained with atom pair or VolSurf descriptors. Crivori *et al*. used partial least squares discriminant (PLSD) analysis with VolSurf descriptors to train a P–gp substrate classifier with data for 53 diverse drugs [26]. The PLSD classifier made 72% correct predictions for an external set of 272 compounds.

We demonstrate here the AIRS application to the recognition of P–glycoprotein substrates for a dataset of 201 chemicals, consisting of 116 P–gp substrates (P–gpS) and 85 P–gp nonsubstrates (P–gpNS). The classifiers were computed from 159 structural descriptors from five classes, namely 18 constitutional descriptors, 28 topological indices, 84 electrotopological state indices, 13 quantum descriptors, and 16 geometrical indices [22]. The classification performance of the AIRS algorithm is afected by 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. In order to explore the AIRS sensitivity to these parameters, leave–20%–out (five–fold) cross–validation predictions were performed over a wide range of values for all eight parameters. All computations were performed with the AIRS2 implementation of Brownlee [21] using Weka 3.5.4 [27].

4 RESULTS AND DISCUSSION

For each AIRS model we report the following statistical indices: TP_c, true positive in calibration (number of P–gpS compounds classified as P–gpS); FN_c, false negative in calibration (number of P–gpS drugs classified as P–gpNS); TN_c, true negative in calibration (number of P–gpNS drugs classified as P–gpNS); FP_c, false positive in calibration (number of P–gpNS drugs classified as P–gpS); Se_c, calibration selectivity; Sp_c, calibration specificity; Ac_c, calibration accuracy; MCC_c, calibration Matthews correlation coefficient [28]; 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.

Table 1. AIRS Calibration and Prediction Statistics 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	98	18	55	30	0.8448	0.6471	0.7612	0.5053
2	0.02	98	18	55	30	0.8448	0.6471	0.7612	0.5053
3	0.03	98	18	55	30	0.8448	0.6471	0.7612	0.5053
4	0.04	98	18	55	30	0.8448	0.6471	0.7612	0.5053
5	0.05	101	15	56	29	0.8707	0.6588	0.7811	0.5473
6	0.06	101	15	56	29	0.8707	0.6588	0.7811	0.5473
7	0.07	102	14	53	32	0.8793	0.6235	0.7711	0.5270
8	0.08	102	14	53	32	0.8793	0.6235	0.7711	0.5270
9	0.09	102	14	53	32	0.8793	0.6235	0.7711	0.5270
10	0.10	101	15	52	33	0.8707	0.6118	0.7612	0.5056
11	0.15	107	9	54	31	0.9224	0.6353	0.8010	0.5939
12	0.20	98	18	54	31	0.8448	0.6353	0.7562	0.4947
13	0.25	99	17	43	42	0.8534	0.5059	0.7065	0.3879
14	0.30	101	15	38	47	0.8707	0.4471	0.6915	0.3562
15	0.35	101	15	38	47	0.8707	0.4471	0.6915	0.3562
16	0.40	101	15	35	50	0.8707	0.4118	0.6766	0.3228
17	0.45	102	14	33	52	0.8793	0.3882	0.6716	0.3123
18	0.50	102	14	33	52	0.8793	0.3882	0.6716	0.3123
19	0.55	102	14	33	52	0.8793	0.3882	0.6716	0.3123
20	0.60	102	14	33	52	0.8793	0.3882	0.6716	0.3123
21	0.65	102	14	33	52	0.8793	0.3882	0.6716	0.3123
22	0.70	102	14	33	52	0.8793	0.3882	0.6716	0.3123
23	0.75	102	14	33	52	0.8793	0.3882	0.6716	0.3123
24	0.80	102	14	33	52	0.8793	0.3882	0.6716	0.3123
25	0.85	102	14	33	52	0.8793	0.3882	0.6716	0.3123
26	0.90	102	14	33	52	0.8793	0.3882	0.6716	0.3123
27	0.95	102	14	33	52	0.8793	0.3882	0.6716	0.3123

Exp	ATS	TP$_p$	FN$_p$	TN$_p$	FP$_p$	Se$_p$	Sp$_p$	Ac$_p$	MCC$_p$
1	0.01	85	31	45	40	0.7328	0.5294	0.6468	0.2671
2	0.02	85	31	47	38	0.7328	0.5529	0.6567	0.2896
3	0.03	85	31	48	37	0.7328	0.5647	0.6617	0.3009
4	0.04	84	32	48	37	0.7241	0.5647	0.6567	0.2915
5	0.05	83	33	46	39	0.7155	0.5412	0.6418	0.2596
6	0.06	82	34	46	39	0.7069	0.5412	0.6368	0.2504
7	0.07	80	36	46	39	0.6897	0.5412	0.6269	0.2320
8	0.08	80	36	45	40	0.6897	0.5294	0.6219	0.2206
9	0.09	79	37	45	40	0.6810	0.5294	0.6169	0.2115
10	0.10	81	35	46	39	0.6983	0.5412	0.6318	0.2412
11	0.15	78	38	51	34	0.6724	0.6000	0.6418	0.2709
12	0.20	83	33	43	42	0.7155	0.5059	0.6269	0.2256
13	0.25	82	34	47	38	0.7069	0.5529	0.6418	0.2617
14	0.30	80	36	42	43	0.6897	0.4941	0.6070	0.1863
15	0.35	76	40	46	39	0.6552	0.5412	0.6070	0.1961
16	0.40	78	38	44	41	0.6724	0.5176	0.6070	0.1911
17	0.45	78	38	44	41	0.6724	0.5176	0.6070	0.1911
18	0.50	80	36	44	41	0.6897	0.5176	0.6169	0.2092
19	0.55	80	36	44	41	0.6897	0.5176	0.6169	0.2092
20	0.60	80	36	43	42	0.6897	0.5059	0.6119	0.1978
21	0.65	80	36	43	42	0.6897	0.5059	0.6119	0.1978
22	0.70	80	36	43	42	0.6897	0.5059	0.6119	0.1978
23	0.75	80	36	43	42	0.6897	0.5059	0.6119	0.1978
24	0.80	80	36	43	42	0.6897	0.5059	0.6119	0.1978
25	0.85	80	36	43	42	0.6897	0.5059	0.6119	0.1978
26	0.90	80	36	43	42	0.6897	0.5059	0.6119	0.1978
27	0.95	80	36	43	42	0.6897	0.5059	0.6119	0.1978

Affinity Threshold Scalar (ATS). This parameter is used in Eq. (4) to compute a cut–off value for memory cell replacement, and takes values between 0 and 1. A candidate ARB replaces a memory cell if the affinity between a candidate ARB and the best matching memory cell is lower that the threshold computed with Eq. (4). A low ATS value results in a low replacement rate, whereas a high ATS value corresponds to a high replacement rate. In order to identify the optimum replacement regimen we varied the ATS value between 0.01 and 0.95 (Table 1, experiments **1**–**27**). 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 = 0.3009 is obtained for ATS = 0.03, indicating that for the P–gp classification problem a low memory cell replacement rate is beneficial. The prediction statistics decrease significantly when ATS increases, suggesting that a high memory cell replacement rate results in poor AIRS models.

Clonal Rate (CR). The clonal rate is used in ARB resource allocation and in controlling the clonal mutation for the memory cells. 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 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]. CR is used in Eq. (3) to determine 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].

Table 2. AIRS Calibration and Prediction Statistics for Various Values of CR (Clonal Rate); (ATS = 0.03)

Exp	CR	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
28	3	106	10	50	35	0.9138	0.5882	0.7761	0.5420
29	5	92	24	56	29	0.7931	0.6588	0.7363	0.4561
30	8	95	21	54	31	0.8190	0.6353	0.7413	0.4640
31	9	92	24	57	28	0.7931	0.6706	0.7413	0.4670
32	10	98	18	55	30	0.8448	0.6471	0.7612	0.5053
33	11	97	19	54	31	0.8362	0.6353	0.7512	0.4843
34	12	99	17	55	30	0.8534	0.6471	0.7662	0.5157
35	15	94	22	56	29	0.8103	0.6588	0.7463	0.4756
36	17	98	18	56	29	0.8448	0.6588	0.7662	0.5159
37	20	98	18	56	29	0.8448	0.6588	0.7662	0.5159

Exp	CR	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
28	3	80	36	48	37	0.6897	0.5647	0.6368	0.2548
29	5	84	32	50	35	0.7241	0.5882	0.6667	0.3140
30	8	82	34	48	37	0.7069	0.5647	0.6468	0.2730
31	9	78	38	47	38	0.6724	0.5529	0.6219	0.2254
32	10	85	31	48	37	0.7328	0.5647	0.6617	0.3009
33	11	79	37	48	37	0.6810	0.5647	0.6318	0.2457
34	12	84	32	49	36	0.7241	0.5765	0.6617	0.3028
35	15	83	33	48	37	0.7155	0.5647	0.6517	0.2822
36	17	84	32	47	38	0.7241	0.5529	0.6517	0.2803
37	20	80	36	48	37	0.6897	0.5647	0.6368	0.2548

The AIRS predictions obtained when the clonal rate was varied between 3 and 20 (Table 2, experiments **28**–**37**) show that there is no apparent trend for the MCC values when CR increases. The best result, MCC = 0.3140, is obtained with CR = 5, with a modest improvement over the best value obtained in the ATS experiments.

Hypermutation Rate (HR). The hypermutation rate is an integer parameter used in Eq. (1) to determine the number of clones for each memory cell, which is in the range [0, CR×HR]. The P–gp substrate classification was investigated for HR between 1 and 10 (Table 3, experiments **38**–**47**), and the best results (HR = 2) show no improvement compared to the best results obtained in the CR experiments.

Table 3. AIRS Calibration and Prediction Statistics for Various Values of HR (Hypermutation Rate); (CR = 5)

Exp	HR	TP$_c$	FN$_c$	TN$_c$	FP$_c$	Se$_c$	Sp$_c$	Ac$_c$	MCC$_c$
38	1	96	20	54	31	0.8276	0.6353	0.7463	0.4741
39	2	92	24	56	29	0.7931	0.6588	0.7363	0.4561
40	3	96	20	55	30	0.8276	0.6471	0.7512	0.4848
41	4	98	18	54	31	0.8448	0.6353	0.7562	0.4947
42	5	100	16	54	31	0.8621	0.6353	0.7662	0.5157
43	6	94	22	60	25	0.8103	0.7059	0.7662	0.5189
44	7	99	17	54	31	0.8534	0.6353	0.7612	0.5051
45	8	102	14	53	32	0.8793	0.6235	0.7711	0.5270
46	9	98	18	55	30	0.8448	0.6471	0.7612	0.5053
47	10	94	22	57	28	0.8103	0.6706	0.7512	0.4864

Exp	HR	TP$_p$	FN$_p$	TN$_p$	FP$_p$	Se$_p$	Sp$_p$	Ac$_p$	MCC$_p$
38	1	81	35	47	38	0.6983	0.5529	0.6368	0.2525
39	2	84	32	50	35	0.7241	0.5882	0.6667	0.3140
40	3	78	38	47	38	0.6724	0.5529	0.6219	0.2254
41	4	83	33	46	39	0.7155	0.5412	0.6418	0.2596
42	5	81	35	49	36	0.6983	0.5765	0.6468	0.2752
43	6	81	35	49	36	0.6983	0.5765	0.6468	0.2752
44	7	81	35	46	39	0.6983	0.5412	0.6318	0.2412
45	8	80	36	48	37	0.6897	0.5647	0.6368	0.2548
46	9	78	38	48	37	0.6724	0.5647	0.6269	0.2368
47	10	81	35	47	38	0.6983	0.5529	0.6368	0.2525

Number of Nearest Neighbors (kNN). During the classification process (Figure 1, step 3), AIRS selects kNN memory cells that have the highest stimulation relative to an antigen, and then that antigen is classified (P–gpS or P–gpNS) based on the vote of those kNN memory cells.

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

Exp	kNN	TP$_c$	FN$_c$	TN$_c$	FP$_c$	Se$_c$	Sp$_c$	Ac$_c$	MCC$_c$
48	1	96	20	56	29	0.8276	0.6588	0.7562	0.4955
49	3	92	24	56	29	0.7931	0.6588	0.7363	0.4561
50	5	96	20	59	26	0.8276	0.6941	0.7711	0.5277
51	7	95	21	55	30	0.8190	0.6471	0.7463	0.4748
52	9	95	21	49	36	0.8190	0.5765	0.7164	0.4100
53	11	93	23	38	47	0.8017	0.4471	0.6517	0.2673
54	13	97	19	35	50	0.8362	0.4118	0.6567	0.2764
55	15	100	16	30	55	0.8621	0.3529	0.6468	0.2528
56	17	101	15	31	54	0.8707	0.3647	0.6567	0.2768
57	19	102	14	31	54	0.8793	0.3647	0.6617	0.2892

Table 4. (Continued)

Exp	kNN	TP$_p$	FN$_p$	TN$_p$	FP$_p$	Se$_p$	Sp$_p$	Ac$_p$	MCC$_p$
48	1	78	38	45	40	0.6724	0.5294	0.6119	0.2025
49	3	84	32	50	35	0.7241	0.5882	0.6667	0.3140
50	5	82	34	50	35	0.7069	0.5882	0.6567	0.2956
51	7	77	39	50	35	0.6638	0.5882	0.6318	0.2507
52	9	84	32	46	39	0.7241	0.5412	0.6468	0.2690
53	11	87	29	40	45	0.7500	0.4706	0.6318	0.2295
54	13	85	31	41	44	0.7328	0.4824	0.6269	0.2216
55	15	83	33	42	43	0.7155	0.4941	0.6219	0.2141
56	17	83	33	42	43	0.7155	0.4941	0.6219	0.2141
57	19	85	31	40	45	0.7328	0.4706	0.6219	0.2102

Although we investigated the effect of kNN for values between 1 and 19 (Table 4, experiments **48**–**57**), the best prediction is obtained for kNN = 3, with no improvement over the HR experiments.

Table 5. AIRS Calibration and Prediction Statistics for Various 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$_c$
58	1	25	91	77	8	0.2155	0.9059	0.5075	0.1619
59	10	95	21	44	41	0.8190	0.5176	0.6915	0.3555
60	20	103	13	33	52	0.8879	0.3882	0.6766	0.3248
61	30	103	13	49	36	0.8879	0.5765	0.7562	0.4967
62	40	97	19	52	33	0.8362	0.6118	0.7413	0.4630
63	50	92	24	56	29	0.7931	0.6588	0.7363	0.4561
64	60	100	16	57	28	0.8621	0.6706	0.7811	0.5472
65	70	100	16	67	18	0.8621	0.7882	0.8308	0.6525
66	80	103	13	69	16	0.8879	0.8118	0.8557	0.7033
67	90	105	11	68	17	0.9052	0.8000	0.8607	0.7132
68	100	107	9	66	19	0.9224	0.7765	0.8607	0.7139
69	120	103	13	67	18	0.8879	0.7882	0.8458	0.6824
70	140	102	14	73	12	0.8793	0.8588	0.8706	0.7360
71	160	104	12	73	12	0.8966	0.8588	0.8806	0.7554
72	180	102	14	72	13	0.8793	0.8471	0.8657	0.7253
73	200	104	12	69	16	0.8966	0.8118	0.8607	0.7134

Exp	IMCPS	TP$_p$	FN$_p$	TN$_p$	FP$_p$	Se$_p$	Sp$_p$	Ac$_p$	MCC$_p$
58	1	43	73	51	34	0.3707	0.6000	0.4677	–0.0298
59	10	63	53	48	37	0.5431	0.5647	0.5522	0.1065
60	20	68	48	48	37	0.5862	0.5647	0.5771	0.1493
61	30	70	46	43	42	0.6034	0.5059	0.5622	0.1087
62	40	75	41	49	36	0.6466	0.5765	0.6169	0.2216
63	50	84	32	50	35	0.7241	0.5882	0.6667	0.3140
64	60	83	33	43	42	0.7155	0.5059	0.6269	0.2256
65	70	81	35	45	40	0.6983	0.5294	0.6269	0.2298
66	80	81	35	43	42	0.6983	0.5059	0.6169	0.2070
67	90	83	33	51	34	0.7155	0.6000	0.6667	0.3160
68	100	85	31	48	37	0.7328	0.5647	0.6617	0.3009
69	120	88	28	49	36	0.7586	0.5765	0.6816	0.3405
70	140	87	29	46	39	0.7500	0.5412	0.6617	0.2974
71	160	80	36	45	40	0.6897	0.5294	0.6219	0.2206
72	180	85	31	47	38	0.7328	0.5529	0.6567	0.2896
73	200	85	31	47	38	0.7328	0.5529	0.6567	0.2896

Initial Memory Cell Pool Size (IMCPS). The number of initial memory cells was modified between 1 and 200 (Table 5, experiments **58**–**73**), and the classification results show that when

IMCPS < 40 the prediction statistics decrease significantly. The best prediction, MCC = 0.3405, is obtained for IMCPS = 120, with a small improvement over the kNN experiments.

Number of Instances to Compute the Affinity Threshold (NIAT). NIAT indicates the number of antigens used to compute the affinity threshold in the AIRS initialization phase. In a series of 12 experiments (NIAT between 20 and all antigens) we found no variation in the prediction MCC. For the remaining sets of experiments we used the same NIAT used in the previous sets (NIAT = all).

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. In order to determine how sensitive are the AIRS predictions to the stimulation threshold, ST was modified between 0.1 and 0.9 (Table 6, experiments **74–88**). The results obtained in this series of experiments show that the P–gp AIRS models are not sensitive to ST, and good predictions are obtained for the entire range of values. For further experiments we selected ST = 0.53, because it gives the best predictions (MCC = 0.3796).

Table 6. AIRS Calibration and Prediction Statistics for Various Values of ST (Stimulation Threshold); (NIAT = all)

Exp	ST	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
74	0.10	104	12	72	13	0.8966	0.8471	0.8756	0.7448
75	0.20	104	12	72	13	0.8966	0.8471	0.8756	0.7448
76	0.30	104	12	71	14	0.8966	0.8353	0.8706	0.7343
77	0.40	103	13	68	17	0.8879	0.8000	0.8507	0.6929
78	0.45	106	10	69	16	0.9138	0.8118	0.8706	0.7339
79	0.47	107	9	68	17	0.9224	0.8000	0.8706	0.7341
80	0.49	104	12	70	15	0.8966	0.8235	0.8657	0.7238
81	0.50	103	13	67	18	0.8879	0.7882	0.8458	0.6824
82	0.51	105	11	72	13	0.9052	0.8471	0.8806	0.7548
83	0.53	107	9	67	18	0.9224	0.7882	0.8657	0.7240
84	0.55	107	9	66	19	0.9224	0.7765	0.8607	0.7139
85	0.60	106	10	71	14	0.9138	0.8353	0.8806	0.7545
86	0.70	104	12	72	13	0.8966	0.8471	0.8756	0.7448
87	0.80	104	12	72	13	0.8966	0.8471	0.8756	0.7448
88	0.90	106	10	71	14	0.9138	0.8353	0.8806	0.7545

Exp	ST	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
74	0.10	87	29	48	37	0.7500	0.5647	0.6716	0.3198
75	0.20	87	29	48	37	0.7500	0.5647	0.6716	0.3198
76	0.30	90	26	49	36	0.7759	0.5765	0.6915	0.3599
77	0.40	90	26	47	38	0.7759	0.5529	0.6816	0.3378
78	0.45	87	29	47	38	0.7500	0.5529	0.6667	0.3086
79	0.47	91	25	47	38	0.7845	0.5529	0.6866	0.3477
80	0.49	90	26	46	39	0.7759	0.5412	0.6766	0.3267
81	0.50	88	28	49	36	0.7586	0.5765	0.6816	0.3405
82	0.51	90	26	47	38	0.7759	0.5529	0.6816	0.3378
83	0.53	92	24	49	36	0.7931	0.5765	0.7015	0.3796
84	0.55	90	26	47	38	0.7759	0.5529	0.6816	0.3378
85	0.60	91	25	49	36	0.7845	0.5765	0.6965	0.3697
86	0.70	89	27	47	38	0.7672	0.5529	0.6766	0.3280
87	0.80	86	30	51	34	0.7414	0.6000	0.6816	0.3438
88	0.90	87	29	49	36	0.7500	0.5765	0.6766	0.3310

Total Resources (TR). The number of total resources of the AIRS model limits the number of B–cells from the ARB pool. The amount of resources assigned to an ARB is calculated with Eq. (2) 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. We investigated AIRS classifiers with TR between 25 and 250, but the prediction MCC was constant in all experiments (MCC = 0.3796), with the exception of the first experiment (TR = 25, MCC = 0.3103). Our results indicate that for the P–gp substrate classification, AIRS is not sensitive to TR.

Table 7. Calibration and Prediction Statistics of Several Machine Learning Models

Exp	Model	TP_c	FN_c	TN_c	FP_c	Se_c	Sp_c	Ac_c	MCC_c
89	BayesNet	98	18	54	31	0.8448	0.6353	0.7562	0.4947
90	NaiveBayes	74	42	74	11	0.6379	0.8706	0.7363	0.5085
91	NaiveBayesUpdateable	104	12	57	28	0.8966	0.6706	0.8010	0.5901
92	Logistic	116	0	85	0	1.0000	1.0000	1.0000	1.0000
93	RBFNetwork	103	13	60	25	0.8879	0.7059	0.8109	0.6100
94	KStar	116	0	85	0	1.0000	1.0000	1.0000	1.0000
95	ADTree	113	3	67	18	0.9741	0.7882	0.8955	0.7905
96	J48	113	3	82	3	0.9741	0.9647	0.9701	0.9388
97	LMT	99	17	66	19	0.8534	0.7765	0.8209	0.6320
98	NBTree	114	2	82	3	0.9828	0.9647	0.9751	0.9490
99	RandomForest	116	0	85	0	1.0000	1.0000	1.0000	1.0000
100	RandomTree	116	0	85	0	1.0000	1.0000	1.0000	1.0000
101	REPTree	114	2	59	26	0.9828	0.6941	0.8607	0.7273

Exp	Model	TP_p	FN_p	TN_p	FP_p	Se_p	Sp_p	Ac_p	MCC_p
89	BayesNet	95	21	42	43	0.8190	0.4941	0.6816	0.3334
90	NaiveBayes	72	44	65	20	0.6207	0.7647	0.6816	0.3822
91	NaiveBayesUpdateable	93	23	54	31	0.8017	0.6353	0.7313	0.4441
92	Logistic	81	35	51	34	0.6983	0.6000	0.6567	0.2978
93	RBFNetwork	91	25	51	34	0.7845	0.6000	0.7065	0.3917
94	KStar	82	34	59	26	0.7069	0.6941	0.7015	0.3973
95	ADTree	87	29	52	33	0.7500	0.6118	0.6915	0.3644
96	J48	92	24	53	32	0.7931	0.6235	0.7214	0.4234
97	LMT	92	24	55	30	0.7931	0.6471	0.7313	0.4452
98	NBTree	94	22	56	29	0.8103	0.6588	0.7463	0.4756
99	RandomForest	101	15	57	28	0.8707	0.6706	0.7861	0.5577
100	RandomTree	84	32	48	37	0.7241	0.5647	0.6567	0.2915
101	REPTree	86	30	44	41	0.7414	0.5176	0.6468	0.2653

Comparison with other Machine Learning Algorithms. In order to compare the AIRS algorithm with other machine learning procedures, we investigated the same P–gpS/P–gpNS classification problem with 13 other machine learning algorithms (Table 7, experiments **100–112**): namely Bayesian network (BayesNet), naïve Bayes classifier (NaiveBayes), updateable naïve Bayes classifier with kernel estimator (NaiveBayesUpdateable), logistic regression with ridge estimator (Logistic), Gaussian radial basis function network (RBFNetwork), K* instance–based classifier (KStar), alternating decision tree (ADTree), C4.5 decision tree (J48), logistic model trees (LMT), decision tree with naïve Bayes classifiers at the leaves (NBTree), random forest (RandomForest), random tree (RandomTree), fast decision tree learner (REPTree). All calculations were performed with Weka 3.5.4 [27], using all descriptors.

The AIRS model gives better predictions than five machine learning algorithms: ADTree, BayesNet, Logistic, RandomTree, and REPTree. On the other hand, the predictions obtained with RandomForest (MCC = 0.5577) are much better than those provided by AIRS and the other machine learning procedures, showing that RandomForest should be the preferred approach for the classification of P–gp substrates/nonsubstrates. Other seven machine learning algorithms are better than AIRS, namely NBTree, LMT, NaiveBayesUpdateable, J48, KStar, RBFNetwork, and NaiveBayes. We want also to emphasize that the AIRS predictions (Ac = 0.7015 and MCC = 0.3796) are as good as the support vector machines reported by Xue *et al.* (Ac = 0.683 and MCC = 0.37) [22].

5 CONCLUSIONS

Artificial immune systems represent a new family of algorithms inspired by the functions, mechanisms, and structure of biological systems. The artificial immune recognition system, AIRS, [18–20] combines several elements of the biological immune system, such as learning, pattern recognition, memory, optimization, and evolution of a population of cells (agents). We recently published two AIRS applications in drug design, namely for the recognition of drugs that induce torsade de pointes [16], and for the identification of the drugs that penetrate the human intestine [17]. In this report we demonstrated the first AIRS application for the recognition of P–glycoprotein substrates.

The AIRS algorithm was applied to the classification of a dataset of 201 chemicals, consisting of 116 P–gp substrates and 85 P–gp nonsubstrates. The chemical structure of all molecules was represented by a set of 159 structural descriptors (18 constitutional descriptors, 28 topological indices, 84 electrotopological state indices, 13 quantum descriptors, and 16 geometrical indices) [22]. The calculations were performed with the AIRS2 algorithm [21] implemented in Weka [27], and the prediction ability was estimated with the leave–20%–out (five–fold) cross–validation. The classification performance of the AIRS2 algorithm was investigated for a wide range of values for the 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.

The AIRS algorithm (best predictions: selectivity 0.793, specificity 0.577, accuracy 0.702, and Matthews correlation coefficient 0.380) is as good as support vector machines [22] in predicting P–gp substrates. We also compared AIRS with other 13 well–established machine learning algorithms, and we found that AIRS surpasses five of them (alternating decision tree, Bayesian network, logistic regression with ridge estimator, random tree, and fast decision tree learner). Other eight machine learning algorithms are better than AIRS, namely RandomForest, NBTree, LMT, NaiveBayesUpdateable, J48, KStar, RBFNetwork, and NaiveBayes. The results presented in this paper add new strong evidence to the previous results [16,17] that demonstrate the utility of AIRS classifiers 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] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer–Verlag, Berlin, 2002.

[8] A. O. Tarakanov, V. A. Skormin, and S. P. Sokolova, *Immunocomputing: Principles and Applications*, Springer–Verlag, Berlin, 2003.

[9] Y. Ishida, *Immunity–Based Systems*, Springer–Verlag, Berlin, 2004.

[10] D. Dasgupta (Ed.), *Artificial Immune Systems and Their Applications*, Springer–Verlag, Berlin, 1999.

[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] 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.

[17] O. Ivanciuc, Artificial immune system prediction of the human intestinal absorption of drugs with AIRS (artificial immune recognition system), *Internet Electron. J. Mol. Des.* **2006**, *5*, 515–529, http://www.biochempress.com.

[18] 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.

[19] A. B. Watkins, AIRS: A resource limited artificial immune classifier. Department of Computer Science, MS Thesis, Mississippi State University, 2001, pp. 81.

[20] 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.

[21] 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.

[22] Y. Xue, Z. R. Li, C. W. Yap, L. Z. Sun, X. Chen, and Y. Z. Chen, Effect of molecular descriptor feature selection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1630–1638.

[23] V. K. Gombar, J. W. Polli, J. E. Humphreys, S. A. Wring, and C. S. Serabjit–Singh, Predicting P–glycoprotein

substrates by a quantitative structure–activity relationship model, *J. Pharm. Sci.* **2004**, *93*, 957–968.

[24] Y. Xue, C. W. Yap, L. Z. Sun, Z. W. Cao, J. F. Wang, and Y. Z. Chen, Prediction of P–glycoprotein substrates by a support vector machine approach, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1497–1505.

[25] P. de Cerqueira Lima, A. Golbraikh, S. Oloff, Y. Xiao, and A. Tropsha, Combinatorial QSAR modeling of P–glycoprotein substrates, *J. Chem. Inf. Model.* **2006**, *46*, 1245–1254.

[26] P. Crivori, B. Reinach, D. Pezzetta, and I. Poggesi, Computational models for identifying potential P–glycoprotein substrates and inhibitors, *Mol. Pharmaceutics* **2006**, *3*, 33–44.

[27] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2 edn., Morgan Kaufmann, San Francisco, 2005.

[28] B. W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, *Biochim. Biophys. Acta* **1975**, *405*, 442–451.